V2403C Series Linux Software User Manual

Version 1.0, January 2022

www.moxa.com/products

V2403C Series Linux Software User Manual

Copyright Notice

Trademarks

Disclaimer

Technical Support Contact Information

www.moxa.com/support

Table of Contents

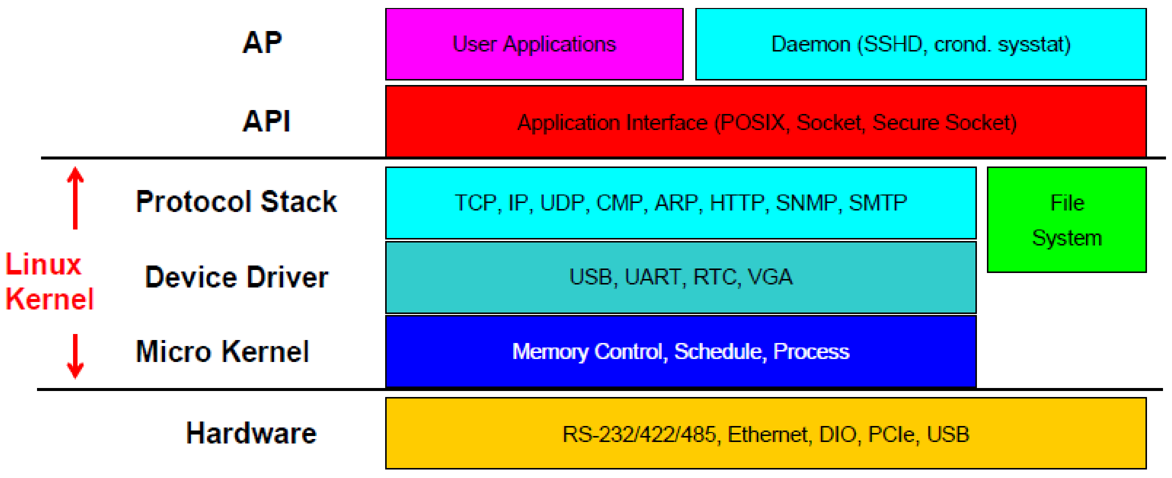
1. Introduction

Thank you for purchasing the Moxa V2403C Series x86 ready-to-run embedded computer. This manual introduces the software configuration and management of the V2403C Series models, which run the Linux operating system.

Linux is an open, scalable operating system that allows you to build a wide range of innovative, small-footprint devices. Software written for desktop PCs can be easily ported to the embedded computer with a GNU cross compiler and minimum source code modifications. A typical Linux-based device is designed for a specific use, and is often not connected to other computers, or a number of such devices connect to a centralized, front-end host.

Software Specifications

The Linux operating system preinstalled on the V2403C embedded computer is the **Debian 9 "Stretch"** distribution. The Debian project involves a worldwide group of volunteers who endeavor to produce an operating system distribution composed entirely of free software. The Debian GNU/Linux follows the standard Linux architecture, making it easy to use programs that meet the POSIX standard. Program porting can be done with the GNU Tool Chain provided by Moxa. In addition to Standard POSIX APIs, device drivers for Moxa UART and other special peripherals are also included. An example software architecture is shown below:


AP
User Applications
Daemon (SSHD, crond. sysstat)
API
Application Interface (POSIX, Socket, Secure Socket)
Linux Kernel
Protocol Stack
TCP, IP, UDP, CMP, ARP, HTTP, SNMP, SMTP
File System
Device Driver
USB, UART, RTC, VGA
Micro Kernel
Memory Control, Schedule, Process
Hardware
RS-232/422/485, Ethernet, DIO, PCIe, USB

ATTENTION

ATTENTION

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

Software Components

The V2403C Linux models are preinstalled with the Debian 9 Stretch Linux distribution. For a list of the software components, refer to the *Software Components* section.

2. Software Configuration

In this chapter, we explain how to operate a V2403C Linux model computer directly from your desktop. You can connect to the computer: using a monitor or via SSH over the network console from a Windows or Linux machine. This chapter describes basic Linux operating system configurations. Advanced network management and configuration instructions will be described in the next "Chapter 3, **Managing Communications**."

Account Management

Connect the V2403C to a display and turn on the computer. Enter the following information to log in into the computer.

Login: moxa
Password: moxa

For security reasons, we have disabled the root account. We strongly recommend changing the password at the first login. After successfully logging in, specify a new password.

```
Using username "moxa".
Linux Moxa 4.9.0-6-amd64 #1 SMP Debian 4.9.88-1 (2018-04-29) x86_64

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

For further information check:
http://www.moxa.com/

You have mail.
Last login: Wed Mar  6 00:10:56 2019 from 10.144.54.91
You are using Moxa embedded computer.
Please change the default password in consideration of higher security level or
disable the default user, moxa.
moxa@Moxa:~$
```

When you finish changing the password, remember to type **sudo** each time you want to run commands with the privilege of a root. For example, typing **`sudo ifconfig enp0s31f6 192.168.100.100`** will allow you to configure the IP address of the LAN 1 port.

```
moxa@Moxa:~$ sudo ifconfig enp0s31f6 192.168.100.100
moxa@Moxa:~$ sudo  ifconfig enp0s31f6
enp0s31f6: flags=4099<UP,BROADCAST,MULTICAST>  mtu 1500
        inet 192.168.100.100  netmask 255.255.255.0  broadcast 192.168.100.255
        ether 00:90:e8:00:d7:38  txqueuelen 1000  (Ethernet)
        RX packets 0  bytes 0 (0.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 0  bytes 0 (0.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
        device memory 0xb1300000-b137ffff
```

In addition, use **`sudo –i`** to login as root to have more privileges.

```
moxa@Moxa:~# sudo -i
[sudo] password for moxa:
root@Moxa:~$
```

For security reasons, the system has enforced a limitation on the number of consecutive invalid access attempts by a user during a set time period based on applicable organizational policy. The configuration is available at **/etc/pam.d/common-auth**.

The system default configuration locks out the user in 120 seconds after 3 unsuccessful login attempts. You can modify this setting in the following entry of the configuration file.

```
...
auth required pam_tally2.so file=/var/log/tallylog deny=3 even_deny_root
unlock_time=120
```

The password strength system default has configured a stronger password strength checking based on minimum length (4-16) and variety of character types by libpam-cracklib module. The configuration is in /etc/pam.d/common-password.

```
...
password        requisite                      pam_cracklib.so retry=3 difok=3
dcredit=-1 lcredit=-1 minlen=12 ocredit=-1 ucredit=-1
password        [success=1 default=ignore]      pam_unix.so obscure use_authtok
try_first_pass sha512
...
```

Setting Up a Desktop Environment

This section introduces the desktop environment setup. The Linux operating system by default doesn't install a desktop environment. Debian supports all kinds of fully featured graphical environment, such as, Gnome, KDE, lighter environment like Xfce and LXDE. Users can choose to install one of these desktop systems. You can use these commands to install a desktop environment:

To install Gnome:

```
moxa@Moxa:~# sudo apt-get install task-gnome-desktop
```

To install KDE:

```
moxa@Moxa:~# sudo apt-get install aptitude tasksel
moxa@Moxa:~# sudo aptitude install ~t^desktop$ ~t^kde-desktop$
```

To install Xfce:

```
moxa@Moxa:~# sudo apt-get install xfce4 xfce4-goodies task-xfce-desktop
```

To install the minimum LXDE:

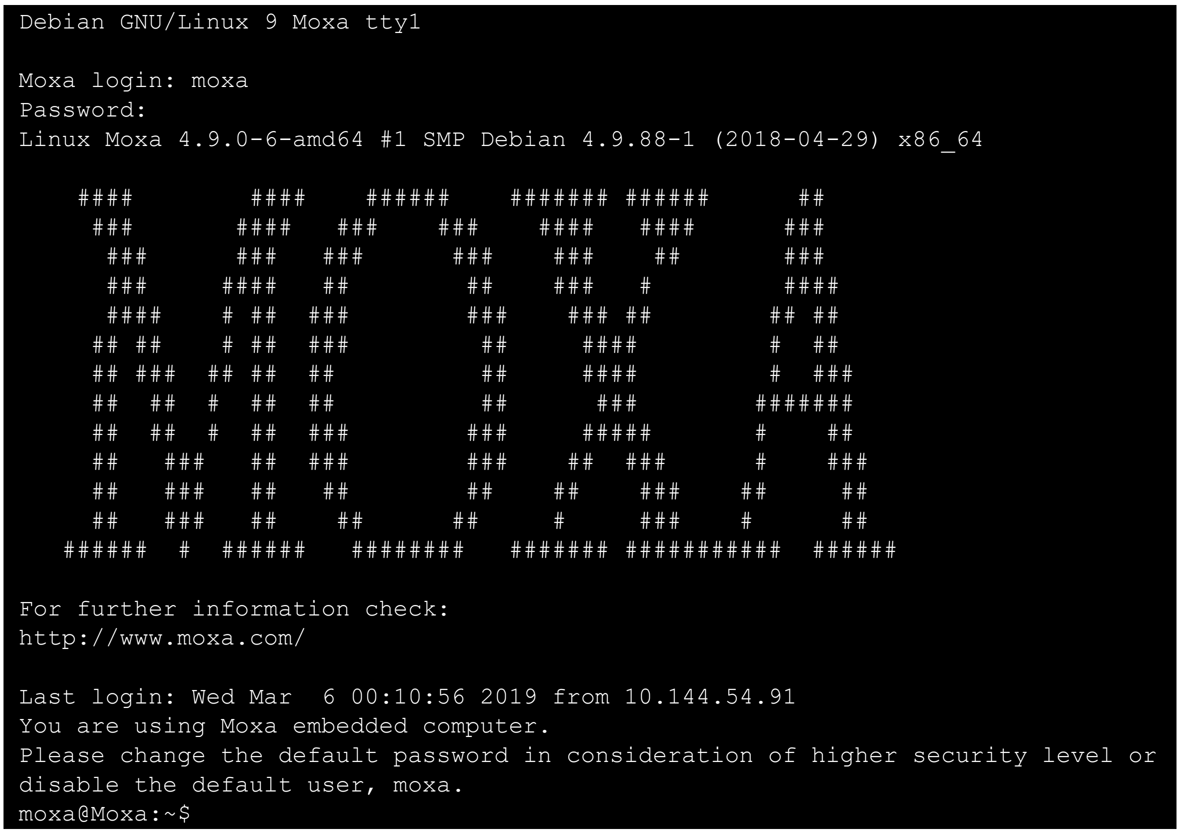
```
moxa@Moxa:~# sudo apt-get install lxde-core lxde
```

Starting From a VGA Console

Connect the display monitor to the connector on your computer, and then power it up by connecting it to the power adapter. It takes approximately 30 to 60 seconds for the system to boot up. Once the system is ready, a login screen will appear on your monitor.

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

Login: moxa
Password: moxa



```
Debian GNU/Linux 9 Moxa tty1

Moxa login: moxa
Password:
Linux Moxa 4.9.0-6-amd64 #1 SMP Debian 4.9.88-1 (2018-04-29) x86_64

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

For further information check:
http://www.moxa.com/

Last login: Wed Mar  6 00:10:56 2019 from 10.144.54.91
You are using Moxa embedded computer.
Please change the default password in consideration of higher security level or
disable the default user, moxa.
moxa@Moxa:~$
```

Connecting From an SSH Console

The V2403C computer supports the SSH console to offer users better network security compared to Telnet. The default IP addresses and netmasks of the network interfaces are as follows:

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

Before using the SSH client, you should change the IP address of your development workstation so that the network ports are on the same subnet as the IP address for the LAN port that you will connect to. For example, if you will connect to LAN1, you could set your PC's IP address to 192.168.3.126, and the netmask to 255.255.255.0. If you will connect to LAN2, you could set your PC's IP address to 192.168.4.126, and the netmask to 255.255.255.0.

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

Login: moxa
Password: moxa

ATTENTION

For security reasons, the system will automatically logout if the ssh remote console or serial console is inactive for 5 minutes. If you do not want the system to automatically logout during the development phase, you can comment out the following configuration settings.

The sshd automatic logout is configured in **/etc/ssh/sshd_config**

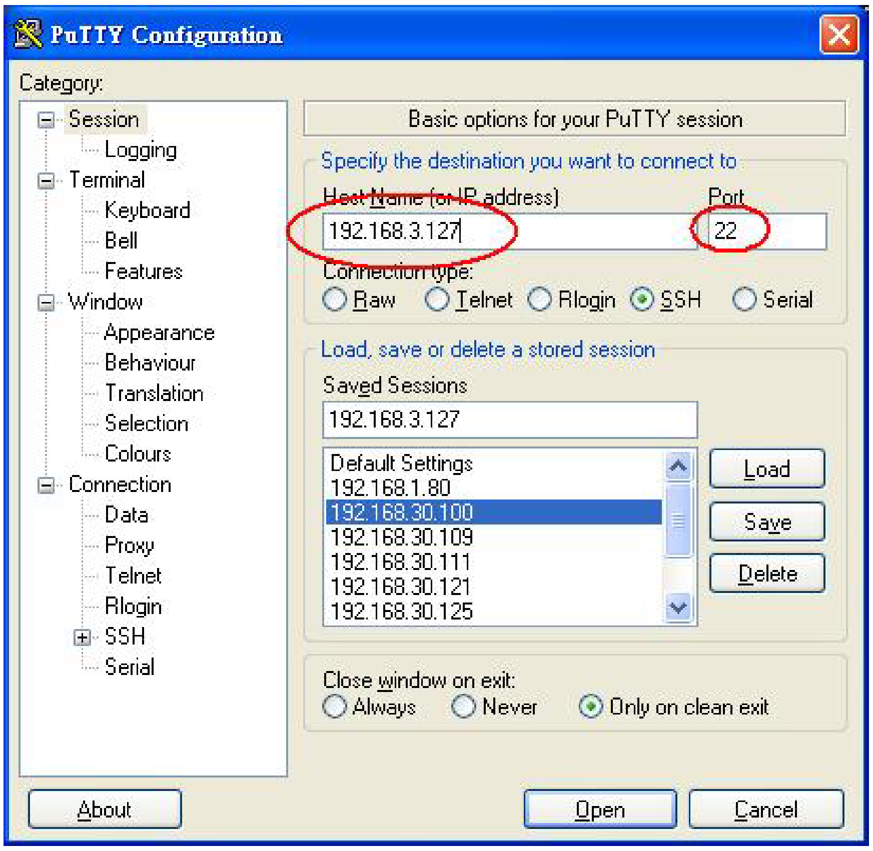
```
...
LoginGraceTime 120
PermitRootLogin without-password
StrictModes yes
```

The console automatic logout is configured in **/etc/profile.d/moxa.sh**

```
TMOUT=300
Export TMOUT
```

Windows Users

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



PuTTY Configuration

Basic options for your PuTTY session

Specify the destination you want to connect to

Host Name (or IP address): 192.168.3.127
Port: 22

Connection type: Raw Telnet Rlogin [✓ SSH] Serial

Load, save or delete a stored session

Saved Sessions: 192.168.3.127

Default Settings
192.168.1.80
192.168.30.100
192.168.30.109
192.168.30.111
192.168.30.121
192.168.30.125

Close window on exit: Always Never [✓ Only on clean exit]

Linux Users

From a Linux machine, use the **ssh** command to access the console utility via SSH.

```
# ssh moxa@192.168.3.127
```

Select **yes** to open the connection.

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

Adjusting the System Time

The V2403C has two time settings. One is the system time, and the other is provided by an RTC (Real-time Clock) built into the hardware.

Setting the Time Manually

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

Use the following command to set the system time.

```
# date MMDDhhmmYYYY
```

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

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

```
# hwclock –w
```

```
root@Moxa:/home/moxa# date
Wed Mar  6 19:33:51 CST 2019
root@Moxa:/home/moxa# hwclock
2019-03-06 19:33:57.482903+0800
root@Moxa:/home/moxa# date 030619352019.30
Wed Mar  6 19:35:30 CST 2019
root@Moxa:/home/moxa# hwclock -w
root@Moxa:/home/moxa# date; hwclock
Wed Mar  6 19:35:34 CST 2019
2019-03-06 19:35:34.061120+0800
```

NTP Client or system-timesyncd Service

The V2403C can use a NTP (Network Time Protocol) client to initialize a time request to a remote NTP server. Use the **ntpdate** command to update the system time. Make sure that the device is connected to an Ethernet network before you run the **ntpdate** command.

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

```
# hwclock –w
```

For more information about NTP and NTP server addresses, visit http://www.ntp.org.

```
root@Moxa:/home/moxa# ntpdate time.stdtime.gov.tw
 6 Mar 19:36:21 ntpdate[1172]: adjust time server 118.163.81.61 offset -
0.000877 sec
root@Moxa:/home/moxa# hwclock -w
root@Moxa:/home/moxa# date; hwclock
Wed Mar  6 19:36:50 CST 2019
2019-03-06 19:36:50.154796+0800
```

The V2403C has a built-in system-timesyncd that is used for Network Time Synchronization. This service default is enabled.

```
root@Moxa:/home/moxa# systemctl status systemd-timesyncd
systemd-timesyncd.service - Network Time Synchronization
   Loaded: loaded (/lib/systemd/system/systemd-timesyncd.service; enabled;
vendor preset: enabled)
  Drop-In: /lib/systemd/system/systemd-timesyncd.service.d
           —disable-with-time-daemon.conf
   Active: active (running) since Wed 2019-03-06 19:30:32 CST; 7min ago
     Docs: man:systemd-timesyncd.service(8)
 Main PID: 274 (systemd-timesyn)
   Status: "Synchronized to time server 103.18.128.60:123
(2.debian.pool.ntp.org)."
    Tasks: 2 (limit: 4915)
   CGroup: /system.slice/systemd-timesyncd.service
           —274 /lib/systemd/systemd-timesyncd

Mar 06 19:30:31 Moxa systemd[1]: Starting Network Time Synchronization...
Mar 06 19:30:32 Moxa systemd[1]: Started Network Time Synchronization.
Mar 06 19:31:02 Moxa systemd-timesyncd[274]: Synchronized to time server
103.18.128.60:123 (2.debian.pool.ntp.org).
```

ATTENTION

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

Managing the Service Using the systemd Script

Linux services can be started or stopped using system script. If you want to start up some service, you can use the **systemctl** command to enable or disable the service.

You can follow this example to add or remove your service in the system. First, you should write a system service unit. This example creates a systemd service unit at **/etc/systemd/system/networking-check.service**.

```
[Unit]
After=snmpd.service

[Service]
ExecStart=/usr/local/bin/networking-check.sh

[Install]
WantedBy=default.target
```

- After: Instructs systemd on when the script should be run. In our case the script will run after snmpd.service has started.
- ExecStart: This field provides a full path the actual script to be execute
- WantedBy: Into what boot target the systemd unit should be installed

This is a basic example of a system script.

You can also check out another example **systemd.serviceNext**. Create the **/usr/local/bin/networking-check.sh** script to check the network status. This example will ping a global DNS server to check if a network is available and write the results into the **/var/log/networking-check.log**.

```
moxa@Moxa:~# sudo vi /usr/local/bin/networking-check.sh
#!/bin/sh

while [ 1 ]; do
     date >> /var/log/networking-check.log
     ping -q -w 1 8.8.8.8
     if [ $? -eq  0 ]; then
            echo "Network is available" >> /var/log/networking-check.log
     else
            echo "Network is not available" >> /var/log/networking-check.log
     fi
     sleep 1
done
```

Before we launch this service, we need to make this script executable:

```
root@Moxa:~# chmod a+x /usr/local/bin/networking-check.sh
```

Then we can start the networking-check service by this command

```
root@Moxa:~# systemctl start networking-check
```

The **networking-check.sh** should launch in the background.

```
root@Moxa:~# ps aux|grep networking-check
root      2260  0.0  0.0   4288  1500 ?        Ss   14:49   0:00 /bin/sh
/usr/local/bin/networking-check.sh
root      2276  0.0  0.0  12784   980 pts/0    S+   14:49   0:00 grep
networking-check
```

/var/log/networking-check.log should be created.

```
root@Moxa:~# cat /var/log/networking-check.log
Wed Mar 14 14:49:09 EDT 2018
Network is available
...
```

Remember use this command to stop this service to prevent the log of this example occupied too much disk space.

```
root@Moxa:~# systemctl stop networking-check
```

Finally, you can enable this service at boot time by this command and reboot the system.

```
root@Moxa:~# systemctl enable networking-check
root@Moxa:~# reboot
```

To disable this service by the systemctl disable command.

```
root@Moxa:~# systemctl disable networking-check
```

Cron—Daemon for Executing Scheduled Commands

The Cron daemon will search the **/etc/crontab** directory for crontab files.

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

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

Mm	h	Dom	mon	Dow	user	command
minute	hour	Date	month	Week	user	command
0-59	0-23	1-31	1-12	0-6 (0 is Sunday)		

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

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

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

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

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

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

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

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

   Add the following line to the end of crontab:

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

Installing a USB Storage Device

This system doesn't support auto mounting of USB storage devices. In a Linux system, the devices should be mounted manually. Before mounting the USB storage, you should check the USB storage name using the **dmesg** command.

```
root@Moxa:~# dmesg
...
[  564.751226] sd 6:0:0:0: Attached scsi generic sg1 type 0
[  564.752400] sd 6:0:0:0: [sdb] 3973118 512-byte logical blocks: (2.03 GB/1.89
GiB)
[  564.753008] sd 6:0:0:0: [sdb] Write Protect is off
[  564.753013] sd 6:0:0:0: [sdb] Mode Sense: 03 00 00 00
[  564.753674] sd 6:0:0:0: [sdb] No Caching mode page found
[  564.753797] sd 6:0:0:0: [sdb] Assuming drive cache: write through
[  564.759333]  sdb: sdb1
[  564.762273] sd 6:0:0:0: [sdb] Attached SCSI removable disk
```

Or check **/proc/partitions**

```
root@Moxa:~# cat /proc/partitions
major minor  #blocks  name
   8        0    7824600 sda
   8        1    7823576 sda1
   8       16    1986559 sdb
   8       17    1985535 sdb1
```

Mount the USB storage partition 1, **/dev/sdb1** on **/mnt**.

```
root:~# mount -t vfat /dev/sdb1 /mnt
```

```
root:~# mount
...
/dev/sdb1 on /mnt type vfat
(rw,relatime,fmask=0022,dmask=0022,codepage=437,iocharset=ascii,shortname=mixed
,utf8,errors=remount-ro)
```

If you want to automatically mount the USB storage at boot time, you can add the following in **/etc/fstab**.

```
...
LABEL=root    /               ext4    noatime,errors=remount-ro 0       1
#usbfs           /proc/bus/usb   usbfs   defaults 0       0
/dev/sdb1        /mnt     vfat    defaults        0       0
```

ATTENTION

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

APT—Installing and Removing Packages

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

Use vi editor to open **/etc/apt/sources.list**.

```
deb mirror://debian.moxa.com/debian/mirrors stretch main contrib non-free

deb  http://deb.debian.org/debian stretch main contrib non-free
#deb-src  http://deb.debian.org/debian stretch main contrib non-free

deb  http://deb.debian.org/debian stretch-updates main contrib non-free
#deb-src  http://deb.debian.org/debian stretch-updates main contrib non-free

deb  http://deb.debian.org/debian stretch-backports main contrib non-free
#deb-src  http://deb.debian.org/debian stretch-backports main contrib non-free

deb  http://security.debian.org/ stretch/updates main contrib non-free
#deb-src  http://security.debian.org/ stretch/updates main contrib non-free
```

1. Update the source list after you configure it.

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

2. Once you indicate which package you want to install (IPsec-tools, for example), type:

```
moxa@Moxa:~# sudo apt-get install ipsec-tools
```

3. Use one of the following commands to remove a package:
   a. For a simple package removal:

```
moxa@Moxa:~# sudo apt-get remove ipsec-tools
```

   b. For a complete package removal:

```
moxa@Moxa:~# sudo apt-get remove ipsec-tools --purge
```

ATTENTION

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

```
moxa@Moxa:~# apt-get clean
```

3. Managing Communications

The V2403C ready-to-run embedded computer is a network-centric platform designed to serve as a front-end for data acquisition and industrial control applications. This chapter describes how to configure the various communication functions supported by the Linux operating system.

Changing the interfaces Configuration File

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

```
root@Moxa:~# cd /etc/network
```

2. Type **vi interfaces** to edit the network configuration file with **vi** editor. You can configure the V2403C computer's Ethernet ports for static or dynamic (DHCP) IP addresses.

```
root@Moxa:~#/etc/network# vi interfaces
```

Static IP Address

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

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

# The primary network interface
auto enp0s31f6
allow-hotplug enp0s31f6
iface enp0s31f6 inet static
        address 192.168.3.127
        netmask 255.255.255.0
        broadcast 192.168.3.255

auto enp9s0
allow-hotplug enp9s0
iface enp9s0 inet static
        address 192.168.4.127
        netmask 255.255.255.0
        broadcast 192.168.4.255

auto enp10s0
allow-hotplug enp10s0
iface enp10s0 inet static
        address 192.168.5.127
        netmask 255.255.255.0
        broadcast 192.168.5.255

auto enp11s0
allow-hotplug enp11s0
iface enp11s0 inet static
        address 192.168.6.127
        netmask 255.255.255.0
        broadcast 192.168.6.255
```

Dynamic IP Address Using DHCP

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

```
# The primary network interface
auto enp0s31f6
iface enp0s31f6 inet dhcp
```

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

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

```
moxa@Moxa:~# sudo ip addr flush enp0s31f6
moxa@Moxa:~# sudo service networking restart
```

Adjusting IP Addresses During Runtime

IP settings can be adjusted at runtime, but the new settings will not be saved to the flash ROM without the file **/etc/network/interfaces** being modified. For example, type the command **`# ifconfig enp0s31f6 192.168.1.1`** to change the IP address of LAN1 to 192.168.1.1.

```
moxa@Moxa:~# sudo ifconfig enp0s31f6 192.168.1.1
```

DNS Client

The V2403C computer supports a DNS client (but not a DNS server). To set up a DNS client, you need to edit three configuration files: **/etc/hostname, /etc/resolv.conf,** and **/etc/nsswitch.conf**.

/etc/hostname

1. Edit /etc/hostname:

```
moxa@Moxa:~# sudo vi /etc/hostname
MOXA
```

2. Reboot the hostname.

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

3. Check the new hostname.

```
moxa@Moxa:~# sudo hostname
```

/etc/resolv.conf

This is the most important file that you need to edit when using DNS. For example, before using **`# ntpdate time.stdtime.gov.tw`** to update the system time, you will need to add the DNS server address to the file. Ask your network administrator which DNS server address you should use. The DNS server's IP address is specified with the **nameserver** command. For example, add the following line to **/etc/resolv.conf** (assuming the DNS server's IP address is 8.8.8.8): **nameserver 8.8.8.8.**

```
moxa@Moxa:/etc# cat resolv.conf
#
# resolv.conf  This file is the resolver configuration file
# See resolver(5).
#
#nameserver 192.168.1.16

nameserver 8.8.8.8
nameserver 8.8.4.4
nameserver 168.95.1.1
```

/etc/nsswitch.conf

This file defines the sequence of files, **/etc/hosts** or **/etc/resolv.conf**, to be read to resolve the IP address. The **hosts** line in **/etc/nsswitch.conf** means use **/etc/host** first and DNS service to resolve the address.

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

passwd:          compat
group:           compat
shadow:          compat

hosts:           files dns
networks:        files

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

netgroup:        nis
```

Configuring Ethernet Bonding

The Linux bonding driver provides a method for aggregating multiple network interfaces into a single logical "bonded" interface. To use the bonding feature, you have to load the bonding driver with the mode setting. Then, use **ifenslave** to add the Ethernet interface into the bond0 interface. Here is the script to bond enp9s0 and enp10s0 together, you can include this in **/etc/init.d/bonding.sh**.

```
#! /bin/bash

NAME=bonding
PATH=/bin:/usr/bin:/sbin:/usr/sbin
BONDING_IP=192.168.3.127

case "$1" in
  start)
    # to set ethX interfaces as slave the bond0 must have an ip
    echo "Starting bonding service: $NAME."
    modprobe bonding mode=1 miimon=100             # load bonding module

    ifdown enp10s0                          # putting down LAN3
    ifdown enp9s0                    # putting down LAN2

    ifconfig bond0 $BONDING_IP netmask 255.255.255.0 up   # set ip address

    ifenslave bond0 enp10s0                 # set LAN3 in slave for bond0
    ifenslave bond0 enp9s0                  # set LAN2 in slave for bond0
    ;;

  stop)
    echo "Stopping bonding service: $NAME"
    ifenslave -d bond0 enp10s0              # release LAN3 from bond0
    ifenslave -d bond0 enp9s0               # release LAN2 from bond0

    ifconfig bond0 down                     # putting down bond0
    modprobe -r bonding                     # unload bonding module
    ifup enp10s0
    ifup enp9s0
    ;;

  restart)
    $0 stop
    $0 start $BONDING_IP
    ;;

  *)
    echo "Usage: systemctl {start|stop|restart} $NAME "
    exit 1
    ;;
esac

exit 0
```

Creates a systemd service unit at **/etc/systemd/system/bonding.service** for bonding the Ethernet services.

```
[Unit]
Description=Bonding service

[Service]
Type=oneshot
ExecStart=/sbin/bonding.sh start
ExecStop=/sbin/bonding.sh stop
RemainAfterExit=yes

[Install]
WantedBy=default.target
```

To install it, use the following command.

```
moxa@Moxa:~# sudo systemctl enable bonding
```

To uninstall it, use this command.

```
moxa@Moxa:~# sudo systemctl disable bonding
```

IPTABLES

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

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

Linux 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 IP address of the packet.
- **Destination NAT (DNAT)**—changes the first destination IP address of the packet.
- **MASQUERADE**—a special form for SNAT. If one host can connect to the Internet, then the other computers that connect to this host can connect to the Internet when the computer does not have an actual IP address.
- **REDIRECT**—a special form of DNAT that re-sends packets to a local host independent of the destination IP address.

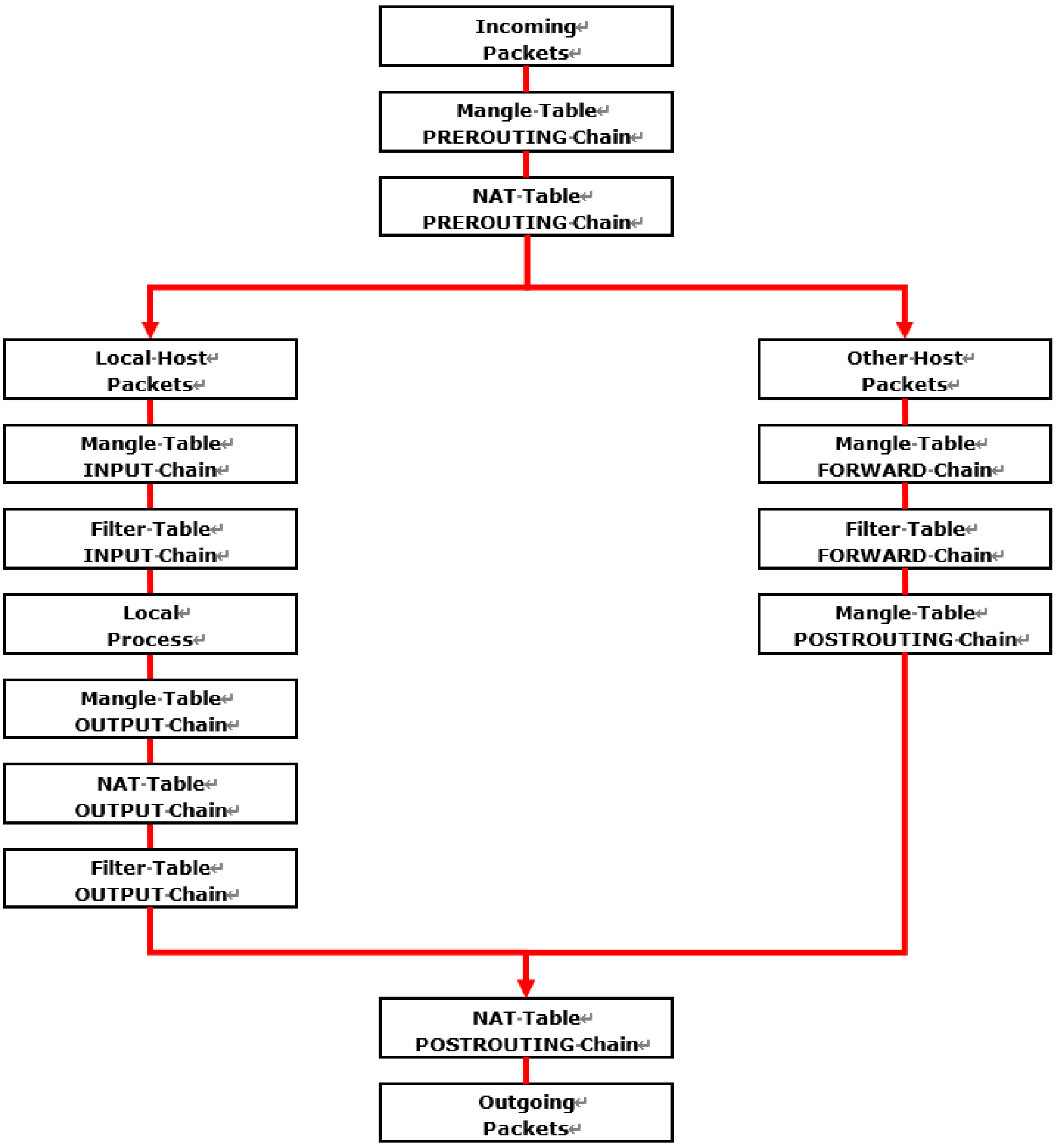
Mangle Table—includes two chains

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

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

IPTABLES Hierarchy

The following figure shows the IPTABLES hierarchy.


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

IPTABLES Modules

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

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

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

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

Use **lsmod** to check if the **ip_tables** module has already been loaded in the V2403C computer. Use **modprobe** to insert and enable the module.

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

ATTENTION

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

http://www.linuxguruz.com/iptables/

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

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

Observing and Erasing Chain Rules

Usage

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

-t tables: Tables 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.

Example

# iptables -L -n

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

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

Defining the Policy for Chain Rules

Usage

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

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

Example

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

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

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 the lo interface.
# iptables -A INPUT -i lo -j ACCEPT

Example 2: Accept TCP packets from 192.168.0.1.
# iptables -A INPUT -i enp0s31f6 -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 enp0s31f6 -p tcp -s 192.168.1.0/24 -j ACCEPT

Example 4: Drop TCP packets from 192.168.1.25.
# iptables -A INPUT -i enp0s31f6 -p tcp -s 192.168.1.25 -j DROP

Example 5: Drop TCP packets addressed for port 21.
# iptables -A INPUT -i enp0s31f6 -p tcp --dport 21 -j DROP

Example 6: Accept TCP packets from 192.168.0.24 to the V2403C computer's port 137, 138, 139
# iptables -A INPUT -i enp0s31f6 -p tcp -s 192.168.0.24 --dport 137:139 -j ACCEPT

Example 7: Log TCP packets that visit the V2403C computer's port 25.
# iptables -A INPUT -i enp0s31f6 -p tcp --dport 25 -j LOG

Example 8: Drop all packets from MAC address 01:02:03:04:05:06.
# iptables -A INPUT -i enp0s31f6 -p all -m mac --mac-source 01:02:03:04:05:06 -j DROP

ATTENTION

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

Network Address Translation

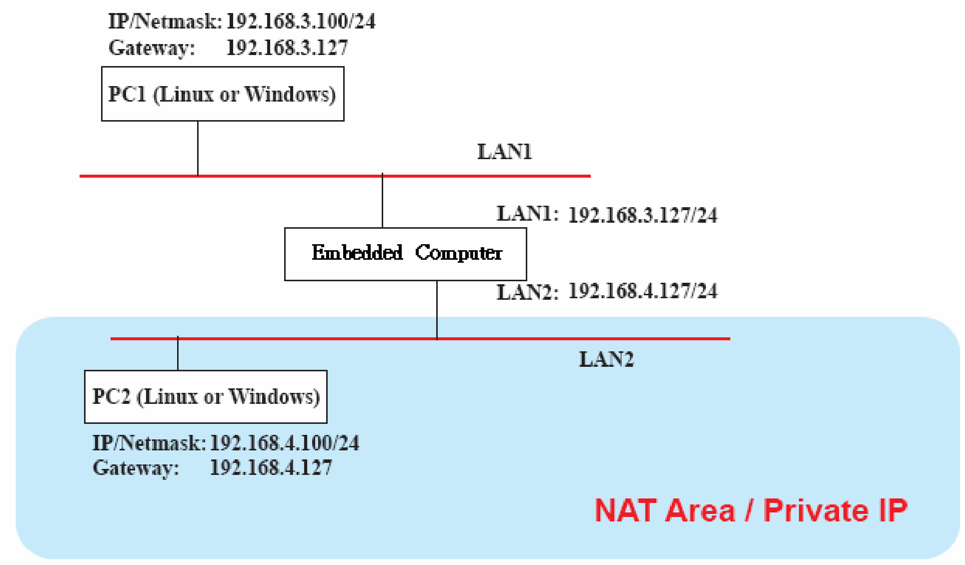
The Network Address Translation (NAT) protocol translates IP addresses used on one network into IP addresses used on a connecting network. One network is designated the inside network and the other is the outside network. Typically, the V2403C computer connects several devices on a network and maps local inside network addresses to one or more global outside IP addresses, and un-maps the global IP addresses on incoming packets back into local IP addresses.

ATTENTION

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

NAT Example

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



Enabling NAT at Bootup

In most real-world situations, you will want to use a simple shell script to enable NAT when the V2403C computer 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= "enp0s31f6"  #This is an external interface for setting up a valid IP
address.
EXNET= "192.168.4.0/24"  #This is an internal network address.
# Step 1. Insert modules.
# Here 2> /dev/null means the standard error messages will be dump to null
device.
modprobe ip_tables 2 > /dev/null
modprobe ip_nat_ftp 2 > /dev/null
modprobe ip_nat_irc 2 > /dev/null
modprobe ip_conntrack 2 > /dev/null
modprobe ip_conntrack_ftp 2 > /dev/null
modprobe ip_conntrack_irc 2 > /dev/null
# Step 2. Define variables, enable routing and erase default rules.
PATH=/bin:/sbin:/usr/bin:/usr/sbin:/usr/local/bin:/usr/local/sbin
export PATH
echo "1" > /proc/sys/net/ipv4/ip_forward
/sbin/iptables -F
/sbin/iptables -X
/sbin/iptables -Z
/sbin/iptables -F -t nat
/sbin/iptables -X -t nat
/sbin/iptables -Z -t nat
/sbin/iptables -P INPUT ACCEPT
/sbin/iptables -P OUTPUT ACCEPT
/sbin/iptables -P FORWARD ACCEPT
```

```
/sbin/iptables -t nat -P PREROUTING ACCEPT
/sbin/iptables -t nat -P POSTROUTING ACCEPT
/sbin/iptables -t nat -P OUTPUT ACCEPT
# Step 3. Enable IP masquerade.
#ehco 1 > /proc/sys/net/ipv4/ip_forward
#modprobe ipt_MASQUERADE
#iptables –t nat –A POSTROUTING -o enp0s31f6 –j MASQUERADE
```

PPP (Point-to-point Protocol)

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

Modem/PPP access is almost identical to connecting directly to a network through the embedded computer Ethernet port. Since PPP is a peer-to-peer system, the Linux operating system can use PPP to link two networks (or a local network to the Internet) to create a Wide Area Network (WAN).

ATTENTION

Click on the following links for more information about PPP:

http://tldp.org/HOWTO/PPP-HOWTO/index.html

http://axion.physics.ubc.ca/ppp-linux.html

Connecting to a PPP Server Over a Dial-up Connection

The following command is used to connect to a PPP server by modem. Use this command for old ppp servers that prompt for a login and password.

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

NOTE

The **debug crtscts** and **defaultroute 192.1.1.17** are optional.

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

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

The pppd options are described below:

connect ‘chat etc…’	This option includes the command to contact the PPP server. The **chat** program is used to dial a remote computer. The entire command is enclosed in single quotes because pppd expects a one-word argument for the **connect** option. The options for **chat** are given below:
-v	verbose mode; log what we do to syslog
“ “	Double quotes—don’t wait for a prompt, but instead just run the command (note that you must include a space after the second quotation mark)
ATDT5551212	Dial the modem, and then ...
CONNECT	Wait for an answer.
“ “	Send a return (null text followed by the usual return)
login: username, password: password	Log in with username and password.

NOTE

For more information on the **chat** utility, refer to the chat man page **chat.8**.

/dev/	Specify the callout serial port.
115200	The baud rate.
debug	Log status in syslog.
crtscts	Use hardware flow control between the computer and modem (at baudrate of 115200 this is a must).
modem	Indicates that this is a modem device; pppd will hang up the phone before and after making the call.
defaultroute	Once the PPP link is established, make it the default route; if you have a PPP link to the Internet, this is probably what you want.
192.1.1.17	This is a degenerate case of a general option of the form x.x.x.x:y.y.y.y. Here x.x.x.x is the local IP address and y.y.y.y is the IP address of the remote end of the PPP connection. If this option is not specified, or if just one side is specified, then x.x.x.x defaults to the IP address associated with the local machine's hostname (located in **/etc/hosts**), and y.y.y.y is determined by the remote machine.

Connecting to a PPP Server Over a Hard-wired Link

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

```
#pppd connect 'chat -v" " " " ' noipdefault /dev/ttyM0 19200 crtscts
```

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

```
#pppd connect 'chat -v" " " " ' user root password root noipdefault /dev/ttyM0
19200 crtscts
```

Checking the Connection

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

```
# /sbin/ifconfig
```

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

ppp0 should be one of the network interfaces. You should recognize the first IP address as the IP address of the computer, and **P-t-P address** is the IP address of the server. The output should be similar to the following:

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

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

Now, type:

# ping z.z.z.z

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

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

Try typing:

# netstat -nr

```
You should see three routes similar to the following:
Kernel routing table
Destination   Gateway       Genmask           Flags  Metric Ref    Use    iface
129.67.1.165  0.0.0.0       255.255.255.255   UH     0      0      6      ppp0
127.0.0.0     0.0.0.0       255.0.0.0         U      0      0      0      lo
0.0.0.0       129.67.1.165  0.0.0.0           UG     0      0      6298   ppp0
```

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

Setting Up a Machine for Incoming PPP Connections

Method 1: pppd dial-in with pppd commands

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

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

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

```
*    *    ""    *
```

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

The following example does not check the username and password:

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

Method 2: pppd dial-in with pppd script

Configure the dial-in script at **/etc/ppp/peer/dialin**.

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

# The chat script (be sure to edit that file, too!)
init "/usr/sbin/chat -v -f /etc/ppp/ppp-ttyMUE0.chat"

# Set up routing to go through this PPP link
defaultroute

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

```
/dev/ttyM0

# Speed
115200

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

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

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

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

```
# pppd call dialin
```

PPPoE

Use the following procedure to configure PPPoE:

1. Connect the V2403C computer's LAN port to an ADSL modem with a cross-over cable, HUB, or switch.
2. Log in to the V2403C computer as the root user.
3. Edit the file **/etc/ppp/chap-secrets** and add the following:

   "username@hinet.net"   *   "password"   *

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

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

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

4. Edit the file /etc/ppp/pap-secrets and add the following:

   "username@hinet.net"   *   "password"   *

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

# INBOUND connections

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


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

# OUTBOUND connections
```

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

5. Edit the file **/etc/ppp/options** and add the following line:

   plugin rp-pppoe

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

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

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

# Load the pppoe plugin
plugin rp-pppoe.so

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

6. If you use LAN1 to connect to the ADSL modem, add the file **/etc/ppp/options.enp0s31f6**, if you use LAN2 to connect to the ADSL modem, add **/etc/ppp/options.enp9s0**, etc.

```
name username@hinet.net
mtu 1492
mru 1492
defaultroute
noipdefault
~
~
 "/etc/ppp/options.enp0s31f6" 5 lines, 67 characters
```

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

7. Set up DNS.

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

   nameserver ip_addr_of_first_dns_server

   nameserver ip_addr_of_second_dns_server

   For example:

   nameserver 168.95.1.1

   nameserver 139.175.10.20

```
moxa@Moxa:/etc# cat resolv.conf
#
# resolv.conf  This file is the resolver configuration file
# See resolver(5).
#
nameserver 168.95.1.1
nameserver 139.175.10.20
```

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

   #pppd enp0s31f6

8. The ADSL modem is connected to the **LAN1** port, which is named **enp0s31f6**. If the ADSL modem is connected to **LAN2**, use **enp9s0**, etc.

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

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

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

Network File System Client

The Network File System (NFS) is used to mount a disk partition on a remote machine (as if it were on a local hard drive), allowing fast and seamless sharing of files across a network. NFS allows users to develop applications for the V2403C computer without worrying about the amount of disk space that will be available. The V2403C computer only supports NFS client protocol.

ATTENTION

Click on the following links for more information about NFS.

http://www.ietf.org/rfc/rfc1213.txt

http://www.faqs.org/rfcs/rfc1317.html

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

1. Scan the NFS Server's shared directory:

   ```
   #showmount  -e  HOST
   ```

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

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

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

3. Mount the remote directory to a local directory:

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

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

Simple Network Management Protocol

The V2403C computer comes with the SNMP v2c (Simple Network Management Protocol) software package. The **snmpd** service is disabled by default. You can enable it using the following command.

```
moxa@Moxa:~# sudo apt-get update && apt-get install snmpd
```

The snmpd configuration is available in **/etc/snmp/snmpd.conf**. If you want to support the SNMP service for all listening interfaces, you should remove the '#' and restart the **snmpd** service.

```
# /etc/snmp/snmpd.conf
# ...
#  Listen for connections from the local system only
# agentAddress  udp:127.0.0.1:161
#  Listen for connections on all interfaces (both IPv4 *and* IPv6)
# For security concern, we comment out this line. If you want to support SNMP
on all Ethernet Interfaces, please remove the '#' and restart the snmpd
service.
# agentAddress udp:161,udp6:[::1]:161
```

Then restart the snmpd service.

```
moxa@Moxa:~# sudo systemctl restart snmpd
```

The following example shows an SNMP agent responding to a query from the SNMP browser on the host site:

```
root@Moxa:/home/moxa# snmpwalk -c public -v2c 127.0.0.1
iso.3.6.1.2.1.1.1.0 = STRING: "Linux Moxa 4.9.0-6-amd64 #1 SMP Debian 4.9.88-1
(2018-04-29) x86_64"
iso.3.6.1.2.1.1.2.0 = OID: iso.3.6.1.4.1.8072.3.2.10
iso.3.6.1.2.1.1.3.0 = Timeticks: (21934) 0:03:39.34
iso.3.6.1.2.1.1.4.0 = STRING: "Me <me@example.org>"
iso.3.6.1.2.1.1.5.0 = STRING: "Moxa"
iso.3.6.1.2.1.1.6.0 = STRING: "Sitting on the Dock of the Bay"
iso.3.6.1.2.1.1.7.0 = INTEGER: 72
iso.3.6.1.2.1.1.8.0 = Timeticks: (1) 0:00:00.01
iso.3.6.1.2.1.1.9.1.2.1 = OID: iso.3.6.1.6.3.11.3.1.1
iso.3.6.1.2.1.1.9.1.2.2 = OID: iso.3.6.1.6.3.15.2.1.1
iso.3.6.1.2.1.1.9.1.2.3 = OID: iso.3.6.1.6.3.10.3.1.1
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.3.6.1.6.3.16.2.2.1
iso.3.6.1.2.1.1.9.1.2.6 = OID: iso.3.6.1.2.1.49
iso.3.6.1.2.1.1.9.1.2.7 = OID: iso.3.6.1.2.1.4
iso.3.6.1.2.1.1.9.1.2.8 = OID: iso.3.6.1.2.1.50
iso.3.6.1.2.1.1.9.1.2.9 = OID: iso.3.6.1.6.3.13.3.1.3
iso.3.6.1.2.1.1.9.1.2.10 = OID: iso.3.6.1.2.1.92
iso.3.6.1.2.1.1.9.1.3.1 = STRING: "The MIB for Message Processing and
Dispatching."
iso.3.6.1.2.1.1.9.1.3.2 = STRING: "The management information definitions for
the SNMP User-based Security Model."
iso.3.6.1.2.1.1.9.1.3.3 = STRING: "The SNMP Management Architecture MIB."
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: "View-based Access Control Model for SNMP."
iso.3.6.1.2.1.1.9.1.3.6 = STRING: "The MIB module for managing TCP
implementations"
iso.3.6.1.2.1.1.9.1.3.7 = STRING: "The MIB module for managing IP and ICMP
implementations"
iso.3.6.1.2.1.1.9.1.3.8 = STRING: "The MIB module for managing UDP
implementations"
iso.3.6.1.2.1.1.9.1.3.9 = STRING: "The MIB modules for managing SNMP
Notification, plus filtering."
iso.3.6.1.2.1.1.9.1.3.10 = STRING: "The MIB module for logging SNMP
Notifications."
iso.3.6.1.2.1.1.9.1.4.1 = Timeticks: (0) 0:00:00.00
iso.3.6.1.2.1.1.9.1.4.2 = Timeticks: (0) 0:00:00.00
iso.3.6.1.2.1.1.9.1.4.3 = Timeticks: (0) 0:00:00.00
iso.3.6.1.2.1.1.9.1.4.4 = Timeticks: (0) 0:00:00.00
iso.3.6.1.2.1.1.9.1.4.5 = Timeticks: (0) 0:00:00.00
iso.3.6.1.2.1.1.9.1.4.6 = Timeticks: (0) 0:00:00.00
iso.3.6.1.2.1.1.9.1.4.7 = Timeticks: (0) 0:00:00.00
iso.3.6.1.2.1.1.9.1.4.8 = Timeticks: (0) 0:00:00.00
iso.3.6.1.2.1.1.9.1.4.9 = Timeticks: (0) 0:00:00.00
iso.3.6.1.2.1.1.9.1.4.10 = Timeticks: (1) 0:00:00.01
iso.3.6.1.2.1.25.1.1.0 = Timeticks: (1105369) 3:04:13.69
iso.3.6.1.2.1.25.1.2.0 = Hex-STRING: 07 E3 03 07 0E 02 10 00 2B 08 00
iso.3.6.1.2.1.25.1.3.0 = INTEGER: 393216
iso.3.6.1.2.1.25.1.4.0 = STRING: "BOOT_IMAGE=/boot/vmlinuz-4.9.0-6-amd64
root=UUID=babacb3b-a96c-449b-89e0-5505c43a1a40 ro quiet
"
iso.3.6.1.2.1.25.1.5.0 = Gauge32: 1
iso.3.6.1.2.1.25.1.6.0 = Gauge32: 102
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)
```

OpenVPN

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

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

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

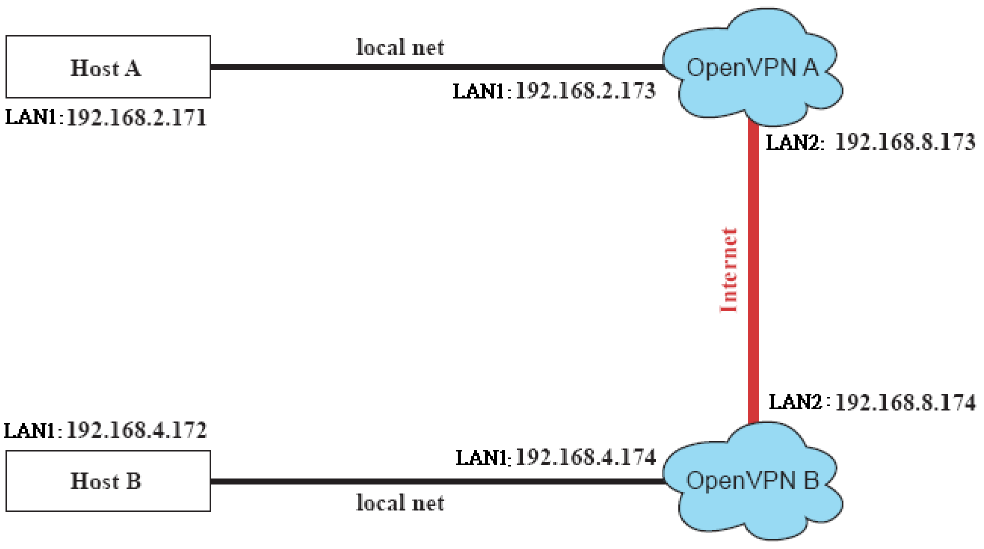
Installing OpenVPN

OpenVPN is the community VPN software. You can install it by this command.

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

Ethernet Bridging for Private Networks on Different Subnets

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



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

2. Generate a preset shared key by typing the following command:

   ```
   # openvpn --genkey --secret secrouter.key
   ```

3. Copy the file that is generated to the OpenVPN machine:

   ```
   # scp /etc/openvpn/secrouter.key 192.168.8.174:/etc/openvpn
   ```

ATTENTION

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

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

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

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

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

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

```
#!/bin/bash
# Create global variables
# Define Bridge Interface
br=“br0”
# Define list of TAP interfaces to be bridged,
# for example tap=“tap0 tap1 tap2”.
tap=“tap0”
# Define physical ethernet interface to be bridged
# with TAP interface(s) above.
eth=“enp9s0”
eth_ip=“192.168.8.173”
eth_netmask=“255.255.255.0”
eth_broadcast=“192.168.8.255”
#gw=“192.168.8.174”
...
```

Start the bridge script file to configure the bridge interface.

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

On machine OpenVPN B, modify the remote address in the configuration file.

#/etc/openvpn/tap0-br.conf.

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

5. Modify the routing table in the **/etc/openvpn/tap0-br.sh script** file as follows:

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

6. Configure the bridge interface in the **/etc/openvpn/bridge** file as follows:

```
#!/bin/bash
# Create global variables
# Define Bridge Interface
br="br0"
# Define list of TAP interfaces to be bridged,
# for example tap="tap0 tap1 tap2".
tap="tap0"
# Define physical ethernet interface to be bridged
# with TAP interface(s) above.
eth="enp9s0"
eth_ip="192.168.8.174"
eth_netmask="255.255.255.0"
eth_broadcast="192.168.8.255"
#gw="192.168.8.173"
...
```

7. Start the bridge script file to configure the bridge interface.

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

ATTENTION

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

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

8. Start both OpenVPN peers on machine OpenVPN A and OpenVPN B.

```
# openvpn --config /etc/openvpn/tap0-br.conf&
```

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

ATTENTION

You can create link symbols to start the OpenVPN service at boot time:

```
# ln -sf /etc/init.d/openvpn /etc/rc2.d/S16openvpn
```

To stop the service, you should create these links:

```
# ln -sf /etc/init.d/openvpn /etc/rc0.d/K80openvpn
# ln -sf /etc/init.d/openvpn /etc/rc6.d/K80openvpn
```

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

```
Destination   Gateway   Genmsk          Flags  Metric  Ref  Use  Iface
192.168.5.0   0.0.0.0   255.255.255.0    U       0      0    0   enp10s0
192.168.4.0   0.0.0.0   255.255.255.0    U       0      0    0   br0
192.168.3.0   0.0.0.0   255.255.255.0    U       0      0    0   enp0s31f6
192.168.30.0  0.0.0.0   255.255.255.0    U       0      0    0   enp11s0
192.168.8.0   0.0.0.0   255.255.255.0    U       0      0    0   br0
```

Interface **enp9s0** and device **tap0** both connect to the bridging interface, and the virtual device **tun** sits on top of **tap0**. This ensures that all traffic coming to this bridge from internal networks connected to interface enp9s0 write to the TAP/TUN device that the OpenVPN program monitors. Once the OpenVPN program detects traffic on the virtual device, it sends the traffic to its peer.

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

    `# route add -net 192.168.4.0 netmask 255.255.255.0 dev enp0s31f6`

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

    `# route add -net 192.168.2.0 netmask 255.255.255.0 dev enp0s31f6`

    Now ping Host B from Host A by typing:

    `# ping 192.168.4.174`

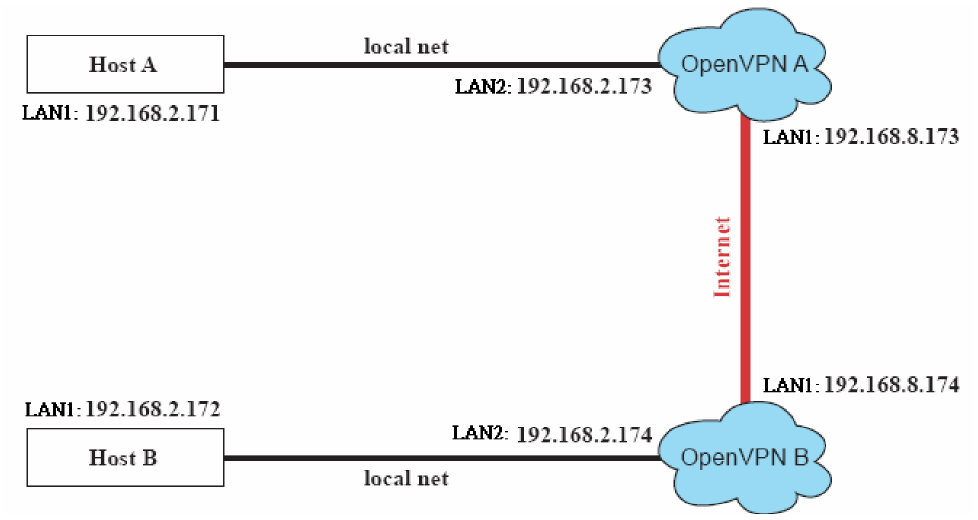
    A successful ping indicates that you have created a VPN system that only allows authorized users from one internal network to access users at the remote site. For this system, all data is transmitted by UDP packets on port 5000 between OpenVPN peers.

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

    `# killall -TERM openvpn`

Ethernet Bridging for Private Networks on the Same Subnet

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

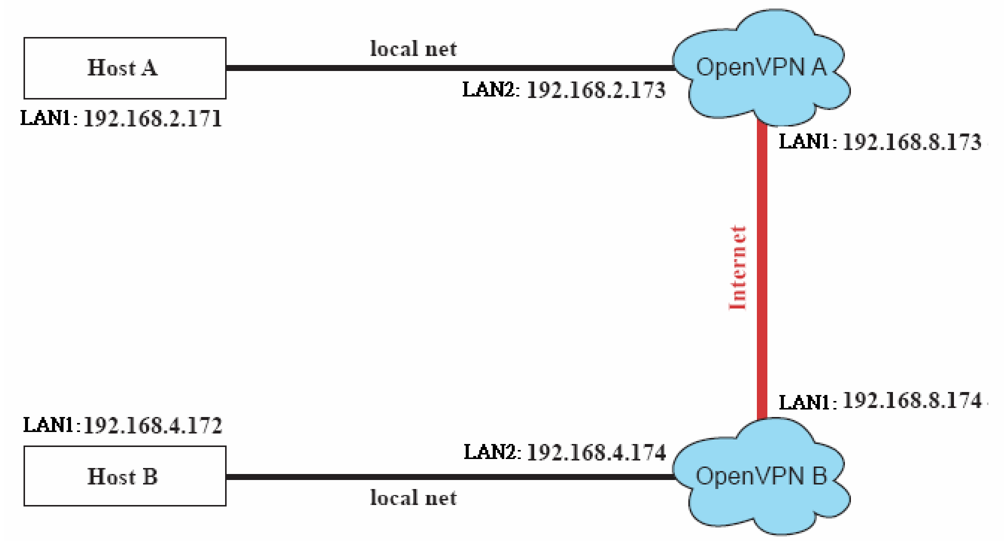


2. The configuration procedure is almost the same as for the previous example. The only difference is that you will need to comment out the parameter **up** in **/etc/openvpn/tap0-br.conf** of OpenVPN A and **/etc/openvpn/tap0-br.conf** of OpenVPN B.

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

Routed IP

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



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

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

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

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

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

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

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

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

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

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

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

```
Destination    Gateway       Genmsk            Flags  Metric  Ref  Use   Iface
192.168.4.174   *            255.255.255.255  UH      0       0    0    tun0
192.168.4.0    192.168.4.174 255.255.255.0    UG      0       0    0    tun0
192.168.2.0     *            255.255.255.0    U       0       0    0    enp9s0
192.168.8.0     *            255.255.255.0    U       0       0    0   enp0s31f6
```

4. System Restore and Backup

The V2403C computer is installed with the Embedded Linux operating system, which is located in the mSATA shipped with the V2403C computer. Although it rarely happens, you may find on occasion that operating system files and/or the disk file system have been damaged. In this chapter we describe how to restore the Linux operating system.

Restore Environment

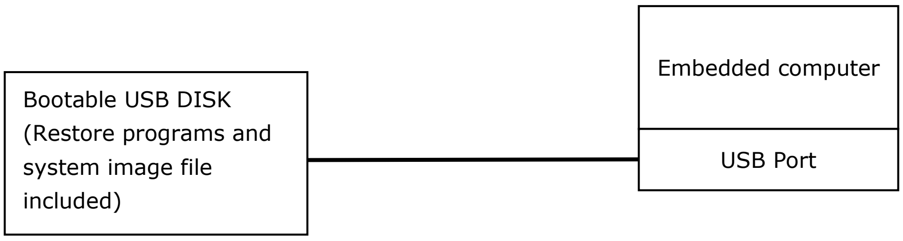
The restore environment includes the V2403C embedded computer and a bootable USB disk with the restore programs and system image file.

Hardware

The hardware used includes a PC, a V2403C computer, and a USB disk with the restore programs.

NOTE

The USB disk should be at least 2GB.


Bootable USB DISK (Restore programs and system image file included)
Embedded computer
USB Port

Restoring the System From a USB Drive

Step 1: Prepare your USB drive

<u>For Windows user:</u>

Execute **Win32DiskImager installer** from the **utility_tools/** folder on the Software CD. Or you can download it from https://sourceforge.net/projects/win32diskimager/

After install processes, execute **Win32DiskImager**, and select the Moxa Live USB image file in the directory of **Restore\moxa_live_image\FWR_<product>_<version>_ReBuild_<date>_live_image.img**

The Moxa Live USB image file **contains corresponding firmware image**.

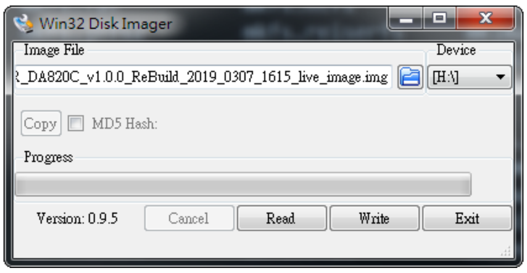

Win32 Disk Imager

Image File: ₹_DA820C_v1.0.0_ReBuild_2019_0307_1615_live_image.img

Device: [H:\]

<u>For Debian Linux user:</u>

Copy the ISO file in the directory of

Restore\moxa_live_image\FWR_<product>_<version>_ReBuild_<date>_live_image.img

For example:

(**/dev/sde** is USB storage device node)

```
root@Moxa:/home/moxa# dd if=FWR_DA820C_<version>_ReBuild_<date>_live_image.img
of=/dev/sde conv=noerror,sync status=progress bs=4096
262144+0 records in
262144+0 records out
1073741824 bytes (1.1 GB, 1.0 GiB) copied, 403.449 s, 2.7 MB/s
root@Moxa:/home/moxa# parted /dev/sde print Fix
Warning: Not all of the space available to /dev/sde appears to be used, you can
fix the GPT to use all of the space (an extra 13126656 blocks) or continue with
the
current setting?
Model: SanDisk Cruzer Blade (scsi)
Disk /dev/sde: 7795MB
Sector size (logical/physical): 512B/512B
Partition Table: gpt
Disk Flags:

Number  Start   End     Size    File system  Name   Flags
 1      1049kB  10.5MB  9437kB  fat16        EFI    boot, esp
 2      10.5MB  268MB   258MB   ext4         ROOT
 3      268MB   1073MB  804MB   ext4         IMAGE
root@Moxa:/home/moxa# parted -s /dev/sde resizepart 3 100%
root@Moxa:/home/moxa# sync
root@Moxa:/home/moxa# e2fsck -fy /dev/sde3
e2fsck 1.43.4 (31-Jan-2017)
Pass 1: Checking inodes, blocks, and sizes
Pass 2: Checking directory structure
Pass 3: Checking directory connectivity
Pass 4: Checking reference counts
Pass 5: Checking group summary information
/dev/sde3: 23/49152 files (0.0% non-contiguous), 136129/196352 blocks
root@Moxa:/home/moxa# resize2fs /dev/sde3
resize2fs 1.43.4 (31-Jan-2017)
Resizing the filesystem on /dev/sde3 to 1837435 (4k) blocks.
The filesystem on /dev/sde3 is now 1837435 (4k) blocks long.
```

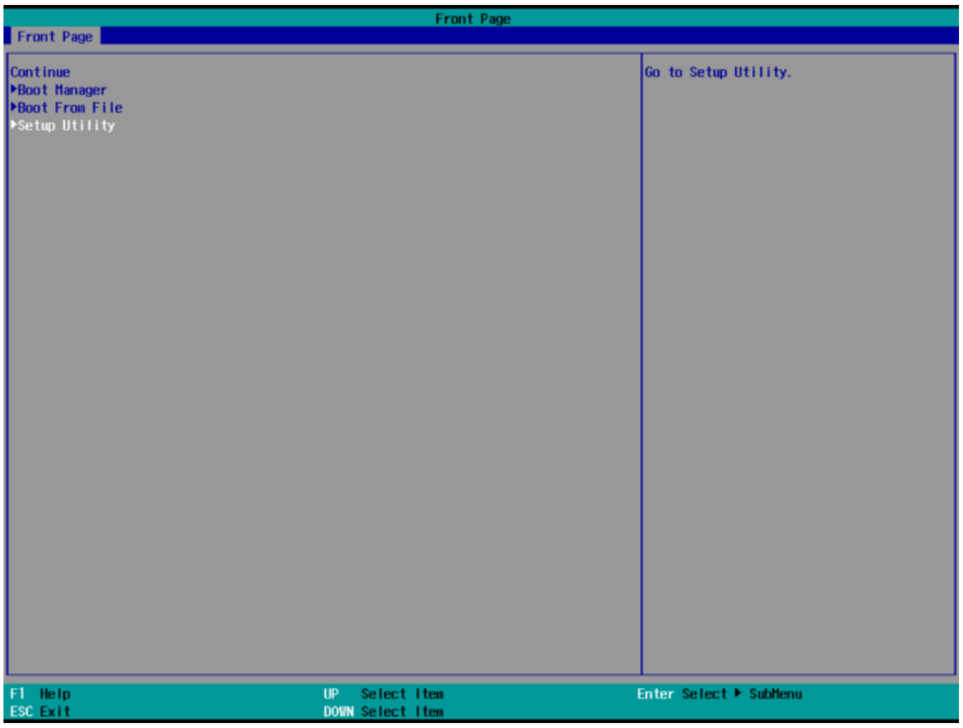
The standalone Debian firmware image file in the directory of
Restore\firmware\FWR_<product>_<version>_ReBuild_<date>.img

Compressed firmware image:
Restore\firmware\FWR_<product>_<version>_ReBuild_<date>.img.gz

Step 2: Change the BIOS settings

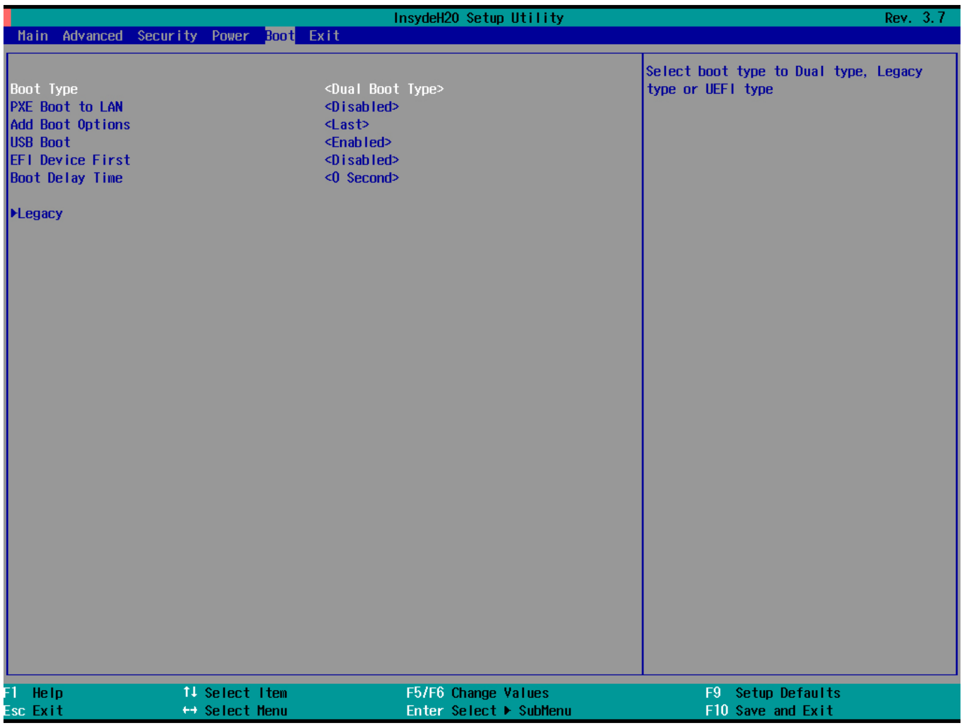
You will need to change the BIOS settings to boot from the USB disk.

1. Turn on the computer and press **F2**. Select **Setup Utility** in the following screen.



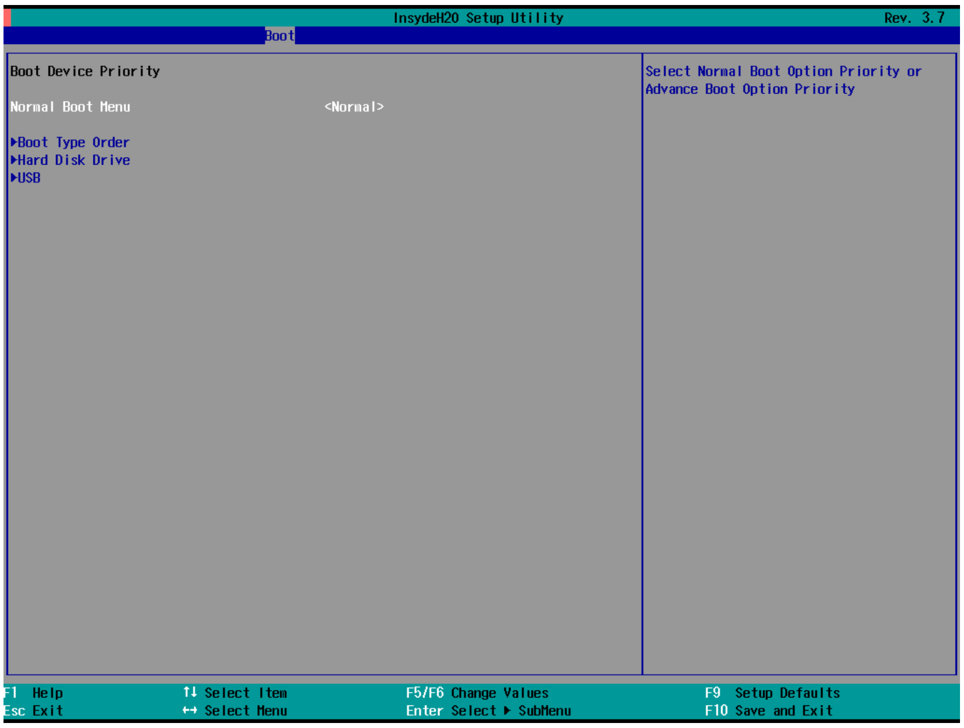
```
                                   Front Page
 Front Page

Continue                                              Go to Setup Utility.
▶Boot Manager
▶Boot From File
▶Setup Utility

F1  Help                     UP   Select Item         Enter Select ▶ SubMenu
ESC Exit                     DOWN Select Item
```

2. Select **Boot** and then select UEFI **Boot Type**. Press **Enter** to continue.



```
                          InsydeH20 Setup Utility                         Rev. 3.7
 Main  Advanced  Security  Power  Boot  Exit

                                                       Select boot type to Dual type, Legacy
Boot Type                    <Dual Boot Type>          type or UEFI type
PXE Boot to LAN              <Disabled>
Add Boot Options             <Last>
USB Boot                     <Enabled>
EFI Device First             <Disabled>
Boot Delay Time              <0 Second>

▶Legacy

F1  Help        ↑↓ Select Item     F5/F6 Change Values         F9  Setup Defaults
Esc Exit        ↔ Select Menu      Enter Select ▶ SubMenu      F10 Save and Exit
```

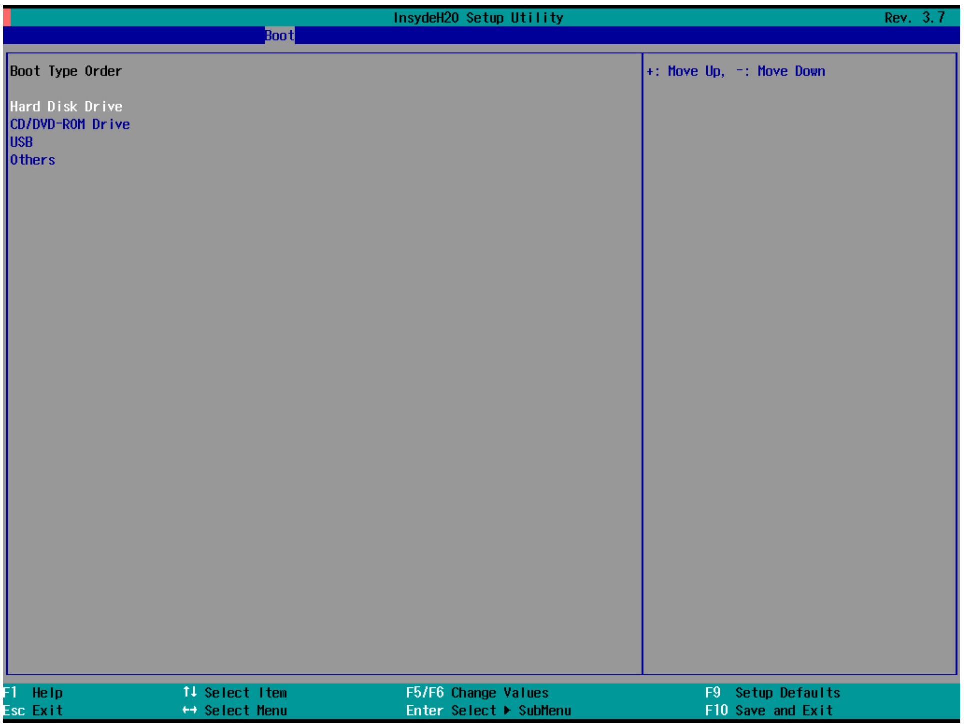
3. Select **Boot order**.


```
InsydeH20 Setup Utility                                        Rev. 3.7
                         Boot
Boot Device Priority                              Select Normal Boot Option Priority or
                                                  Advance Boot Option Priority
Normal Boot Menu                <Normal>

▶Boot Type Order
▶Hard Disk Drive
▶USB

F1  Help        ↑↓ Select Item    F5/F6 Change Values       F9  Setup Defaults
Esc Exit        ↔ Select Menu     Enter Select ▶ SubMenu    F10 Save and Exit
```

4. Select USB disk and then press "+" to move it to the first boot device position.

WARNING

An incorrect boot priority will lead to restore or boot failure.


```
InsydeH20 Setup Utility                                        Rev. 3.7
                         Boot
Boot Type Order                                   +: Move Up, -: Move Down

Hard Disk Drive
CD/DVD-ROM Drive
USB
Others

F1  Help        ↑↓ Select Item    F5/F6 Change Values       F9  Setup Defaults
Esc Exit        ↔ Select Menu     Enter Select ▶ SubMenu    F10 Save and Exit
```

5. Press **F10** and then press **Enter** to save and exit BIOS setup.
6. Insert the USB disk and then reboot the computer.
7. Press F2 to enter the BIOS setting.

8. Select the **Boot Manager**.
9. Select EFI **USB device**.

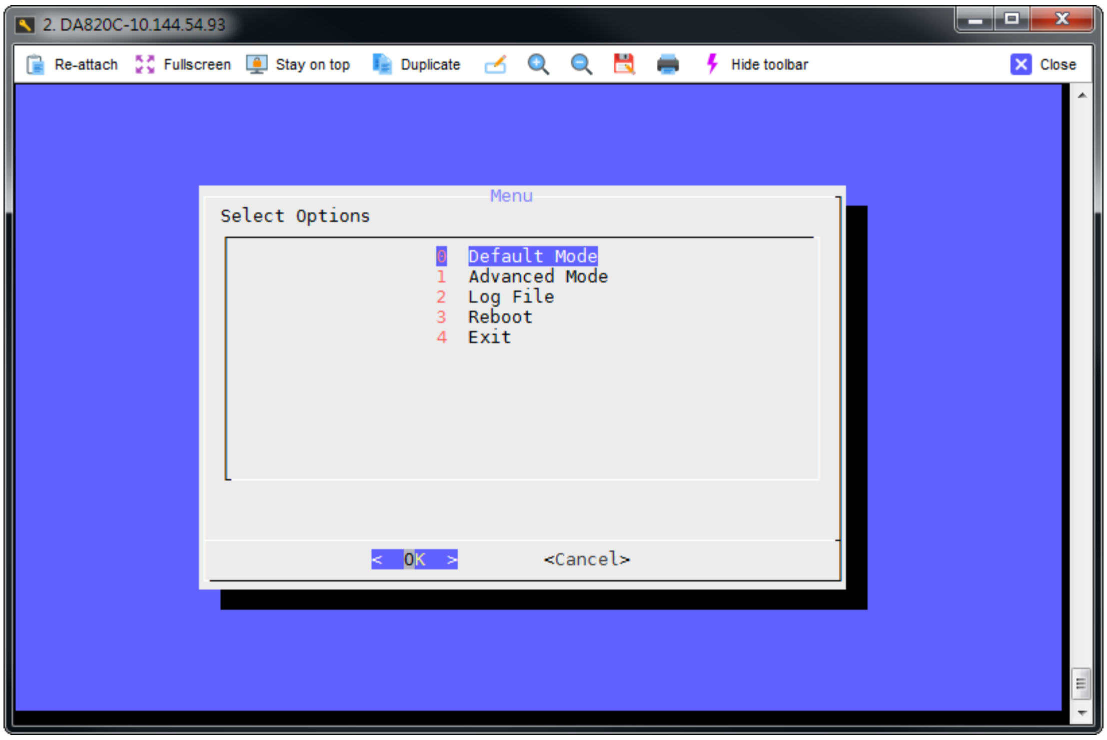
   The system will boot from the restore utility.

Step 3: Restore the system from the USB drive

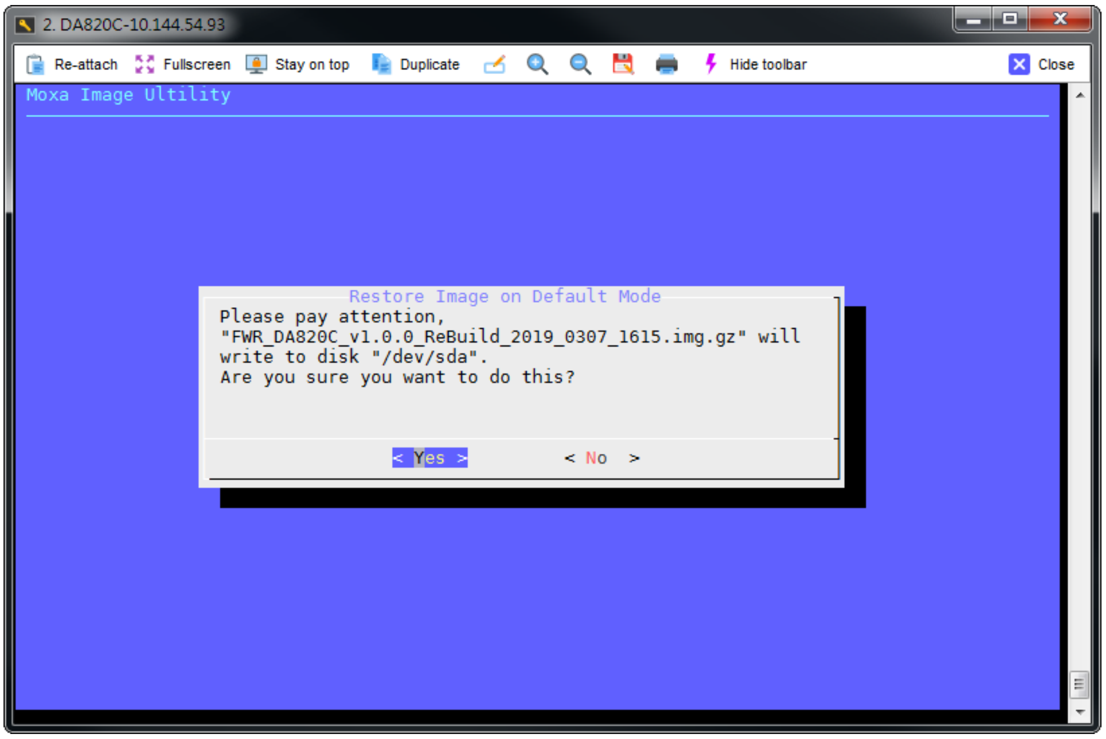
Connect the USB disk to any of the V2403C computer's USB ports and then reboot the computer. The system will boot from the USB disk and the Pre-installation Environment and the restore utility will appear.

[Default Mode]

Select "Default Mode" will write default image to default mSATA disk. **If you have multiple images or storage disks, to select "Advanced mode" is strongly suggested.**



```
Menu
Select Options

   0  Default Mode
   1  Advanced Mode
   2  Log File
   3  Reboot
   4  Exit

      <  OK  >        <Cancel>
```

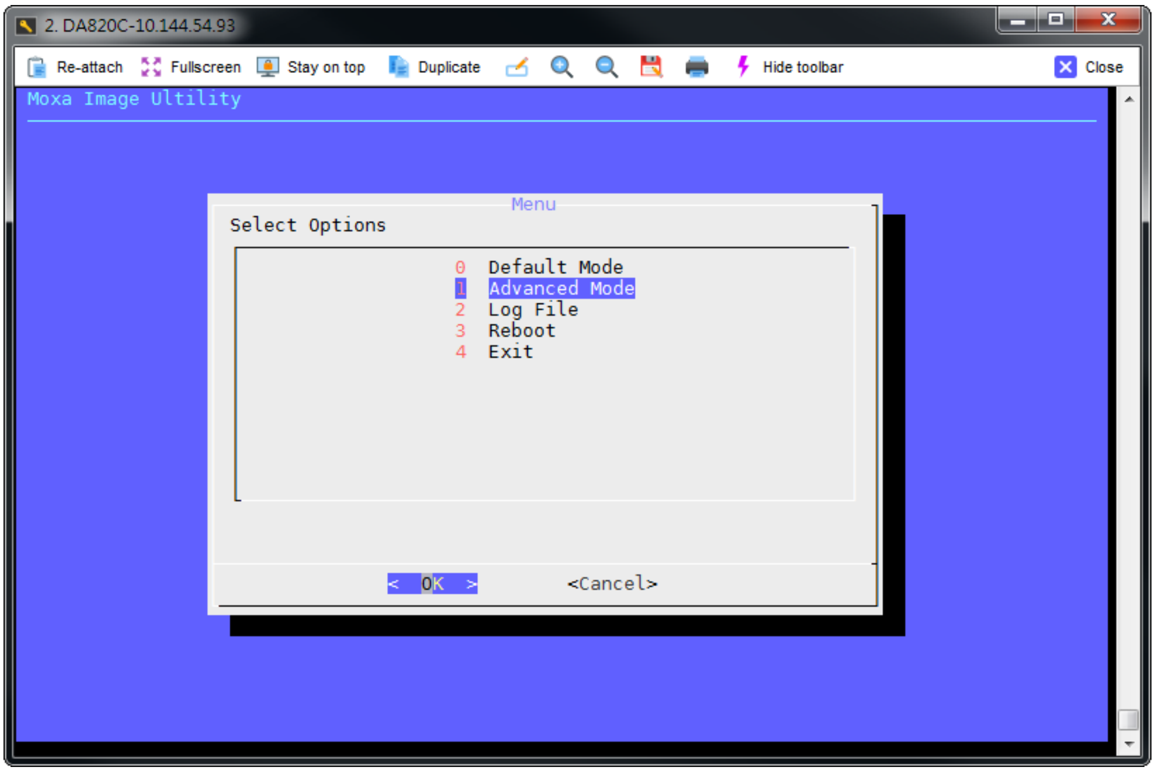


```
Moxa Image Ultility

Restore Image on Default Mode
Please pay attention,
"FWR_DA820C_v1.0.0_ReBuild_2019_0307_1615.img.gz" will
write to disk "/dev/sda".
Are you sure you want to do this?

      < Yes >        <  No  >
```

Press OK and wait for restore image process. If the process was finished, you can select to reboot, and remove the USB drive after the computer has been powered off and jump to Step4.

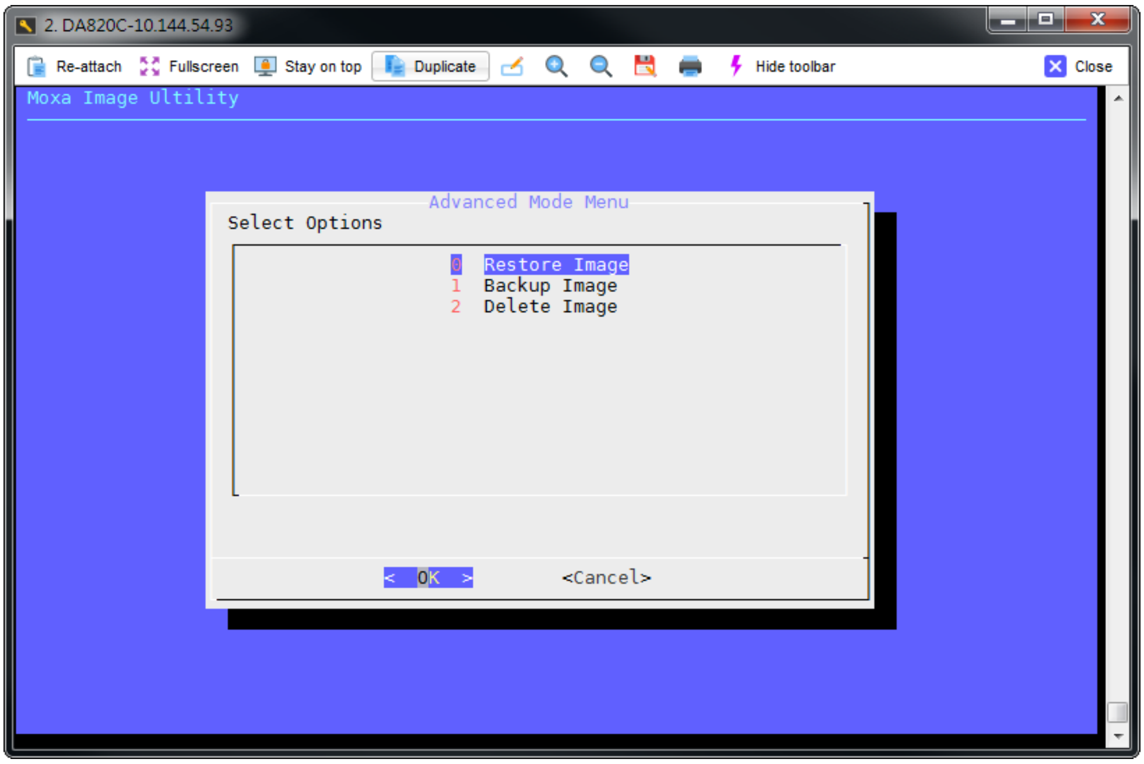
[Advanced Mode]

To select "Advanced Mode":



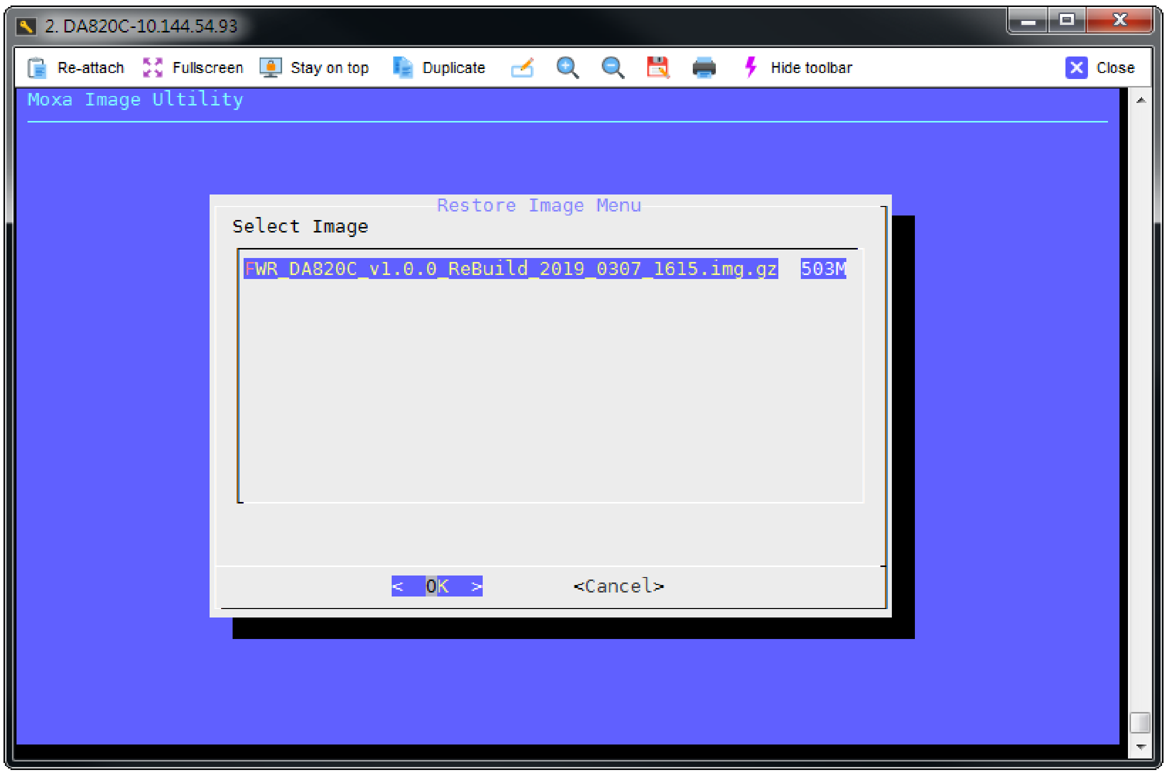
```
Moxa Image Ultility

                         Menu
 Select Options

              0  Default Mode
              1  Advanced Mode
              2  Log File
              3  Reboot
              4  Exit

        <  OK  >          <Cancel>
```

To select "Restore Image":



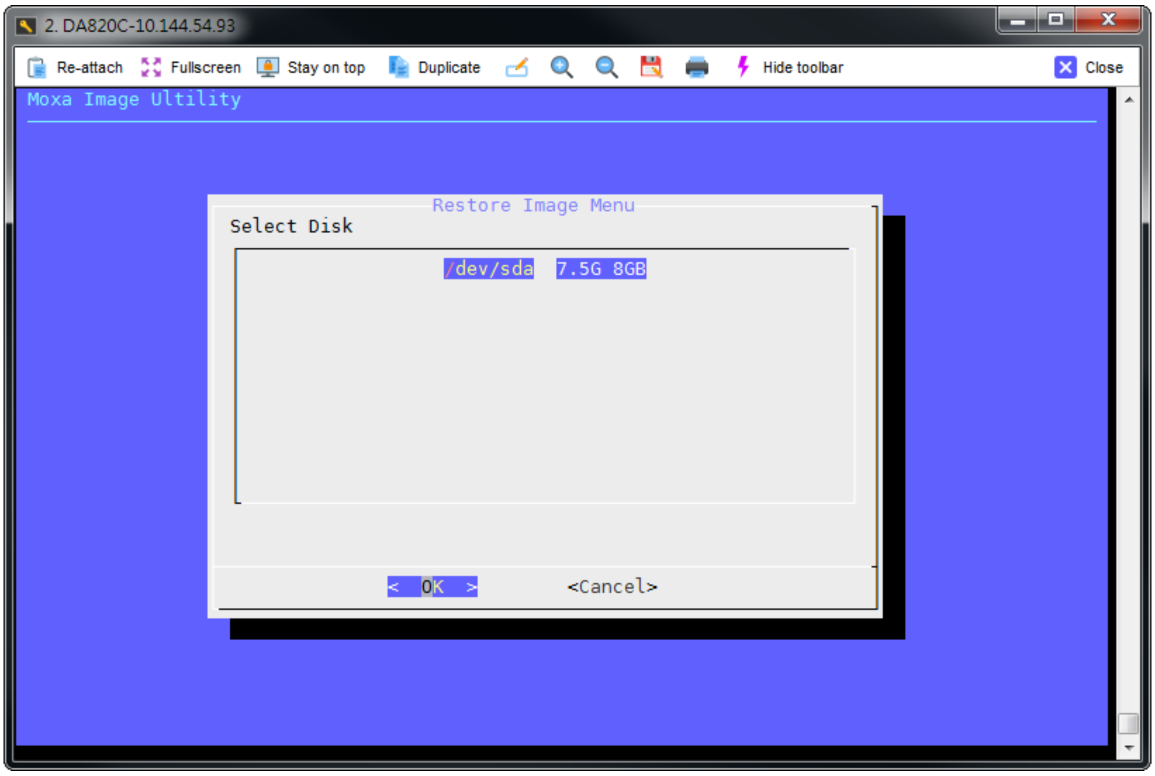
```
Moxa Image Ultility

                   Advanced Mode Menu
 Select Options

              0  Restore Image
              1  Backup Image
              2  Delete Image

        <  OK  >          <Cancel>
```

To select the target image:



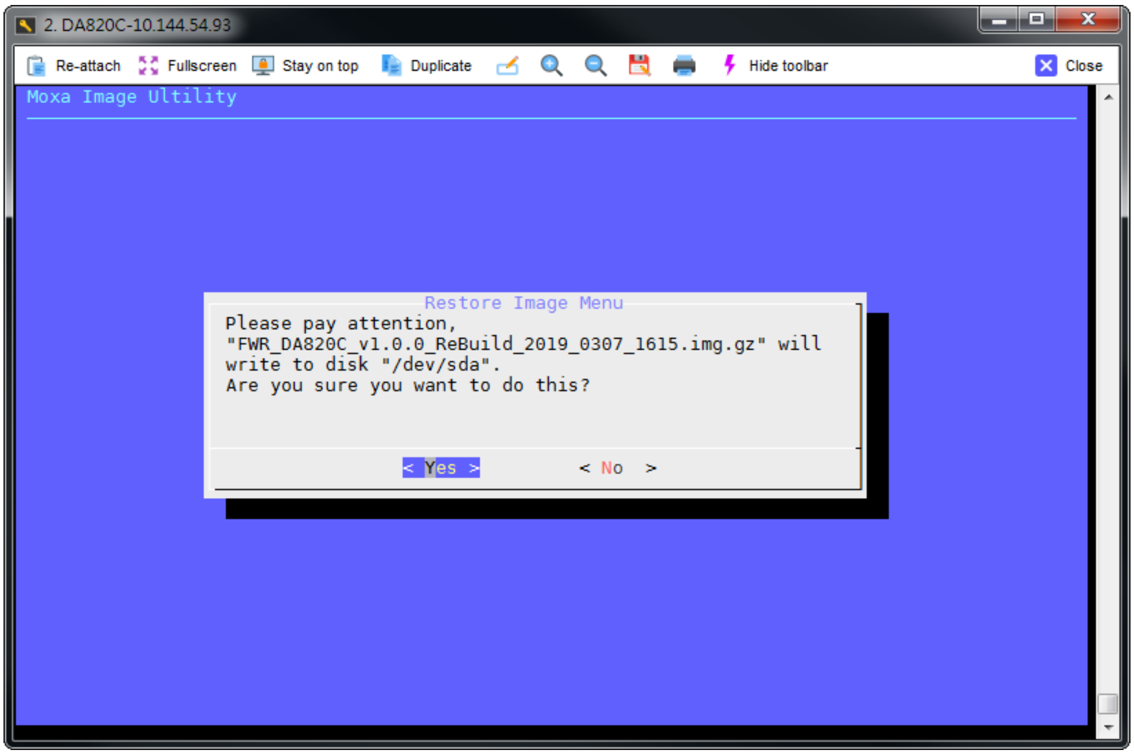
```
Moxa Image Ultility

                    Restore Image Menu
 Select Image
 FWR_DA820C_v1.0.0_ReBuild_2019_0307_1615.img.gz  503M

             <  OK  >          <Cancel>
```

To select the target storage disk:



```
Moxa Image Ultility

                    Restore Image Menu
 Select Disk
                  /dev/sda  7.5G 8GB

             <  OK  >          <Cancel>
```

And make sure again, this step will erase all partitions in the disk.



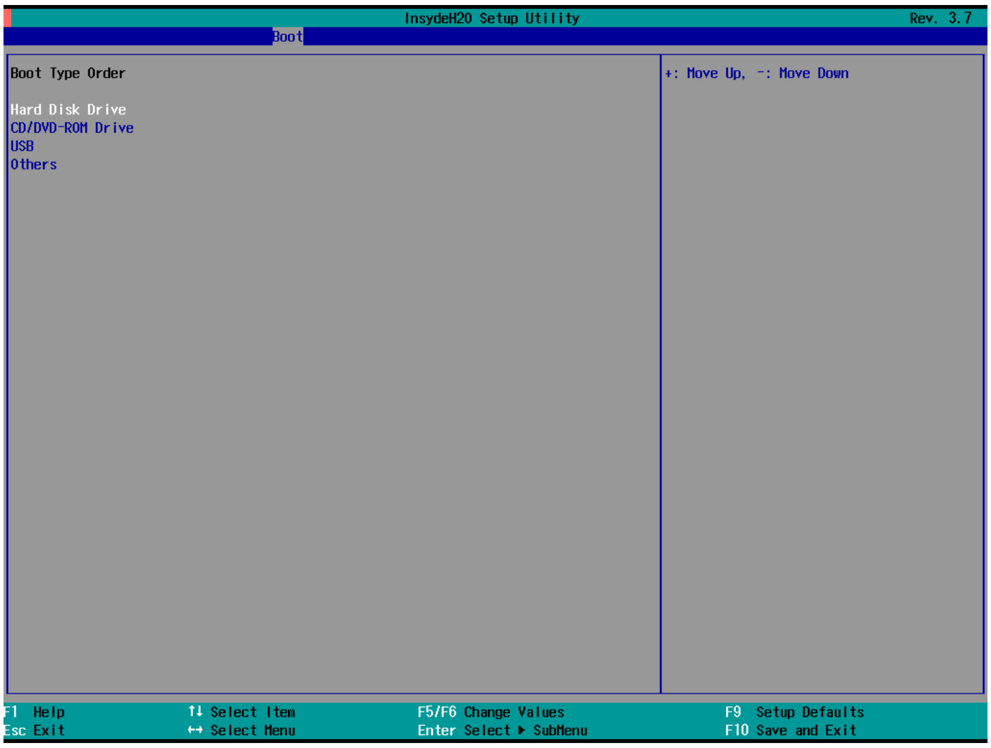
```
Moxa Image Ultility

                 Restore Image Menu
Please pay attention,
"FWR_DA820C_v1.0.0_ReBuild_2019_0307_1615.img.gz" will
write to disk "/dev/sda".
Are you sure you want to do this?

              < Yes >          < No  >
```

Press OK and wait for restore image process. If the process was finished, you can select to reboot, and remove the USB drive after the computer has been powered off and jump to Step4.

Step 4: Change the BIOS Settings to Boot from the Original Disk

Now you will need to change the boot priority so that it can boot from the original disk. As the system reboots, press **F2** to enter the BIOS setup menu.

1. Select **Hard Disk Drive** and then press + to move to the first boot device position, and then press **Enter**. Make sure the hard disk has first boot priority.



```
                          InsydeH20 Setup Utility                          Rev. 3.7
                          Boot
Boot Type Order                                  +: Move Up, -: Move Down

Hard Disk Drive
CD/DVD-ROM Drive
USB
Others

F1  Help        ↑↓ Select Item     F5/F6 Change Values        F9  Setup Defaults
Esc Exit        ↔ Select Menu      Enter Select ▸ SubMenu     F10 Save and Exit
```

2. Press **F10** and then press **Enter** to save and exit BIOS settings.

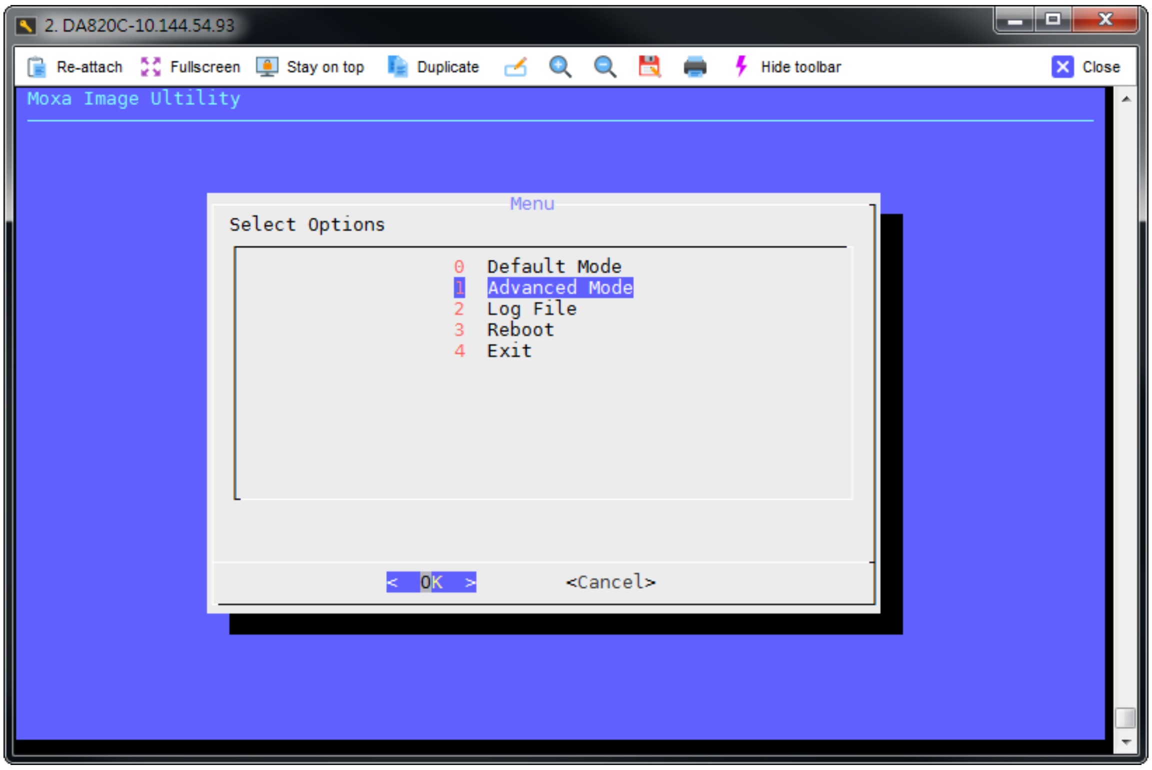
Step 5: Reboot the Computer

You need to wait about 10 to 15 minutes for the system to restart, since the system configuration files will be initiated while booting up for the first time. **Do not turn off the computer or shut down the computer** while the system is restarting; otherwise, the IIS service will be terminated. When the operating system has successfully launched, you will need to restart your computer so that the new settings can be activated.

Backing Up the System to a USB Drive

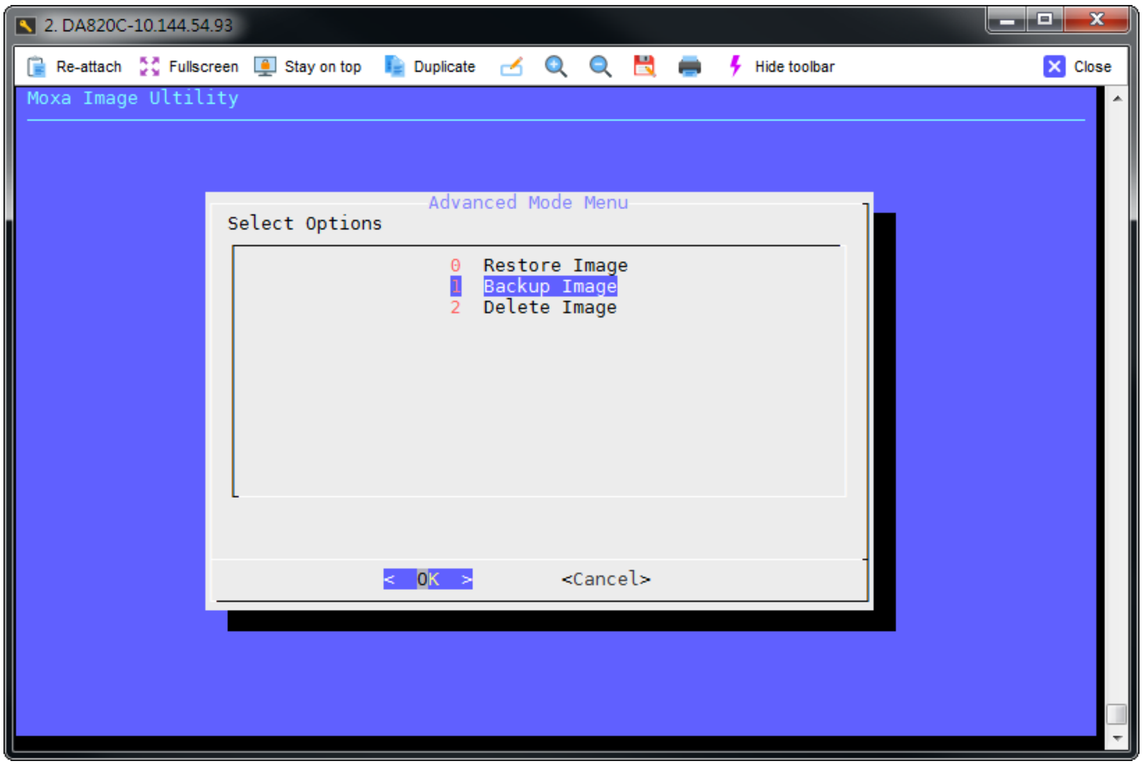
You may also backup the current system to the USB drive for system restore in case the system crashes. Change the BIOS settings to make the USB drive the first boot priority. When the system has been launched, take the following steps.

To select "Advanced Mode":



```
Moxa Image Ultility

Menu
Select Options

0  Default Mode
1  Advanced Mode
2  Log File
3  Reboot
4  Exit

<  OK  >          <Cancel>
```

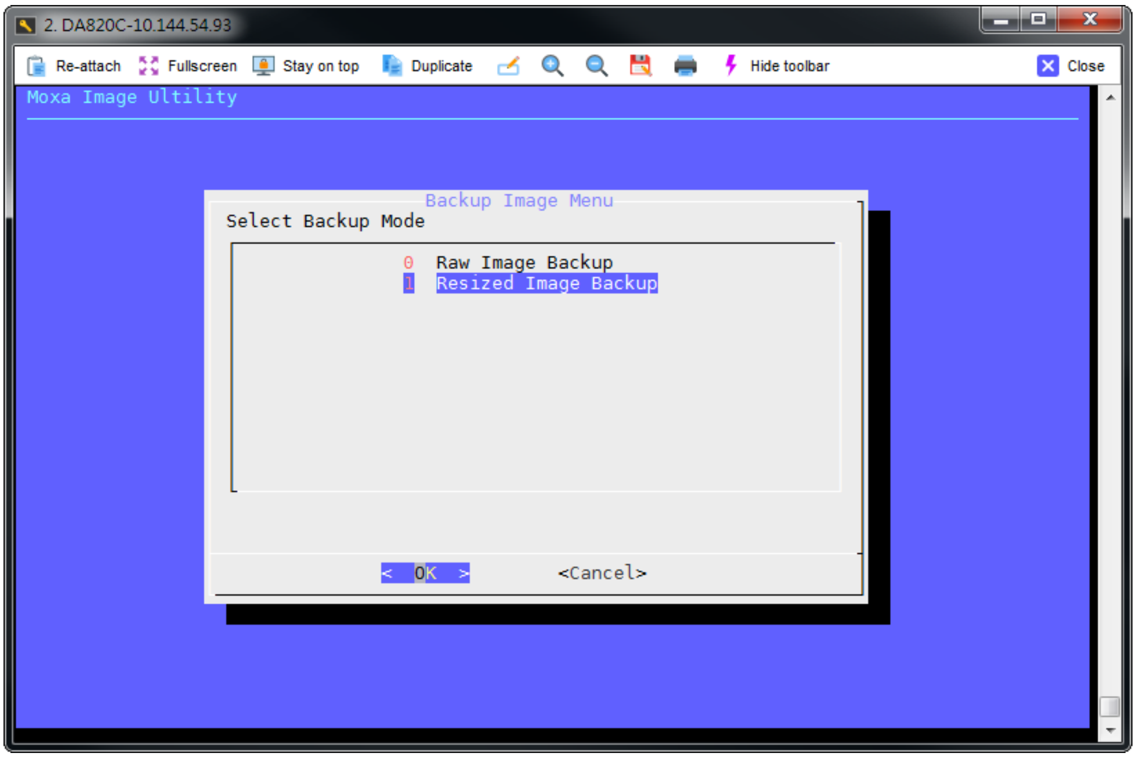
To select "Backup Image":


```
2. DA820C-10.144.54.93
Re-attach  Fullscreen  Stay on top  Duplicate  Hide toolbar  Close
Moxa Image Ultility

                 Advanced Mode Menu
 Select Options
                    0  Restore Image
                    1  Backup Image
                    2  Delete Image

              <  OK  >          <Cancel>
```

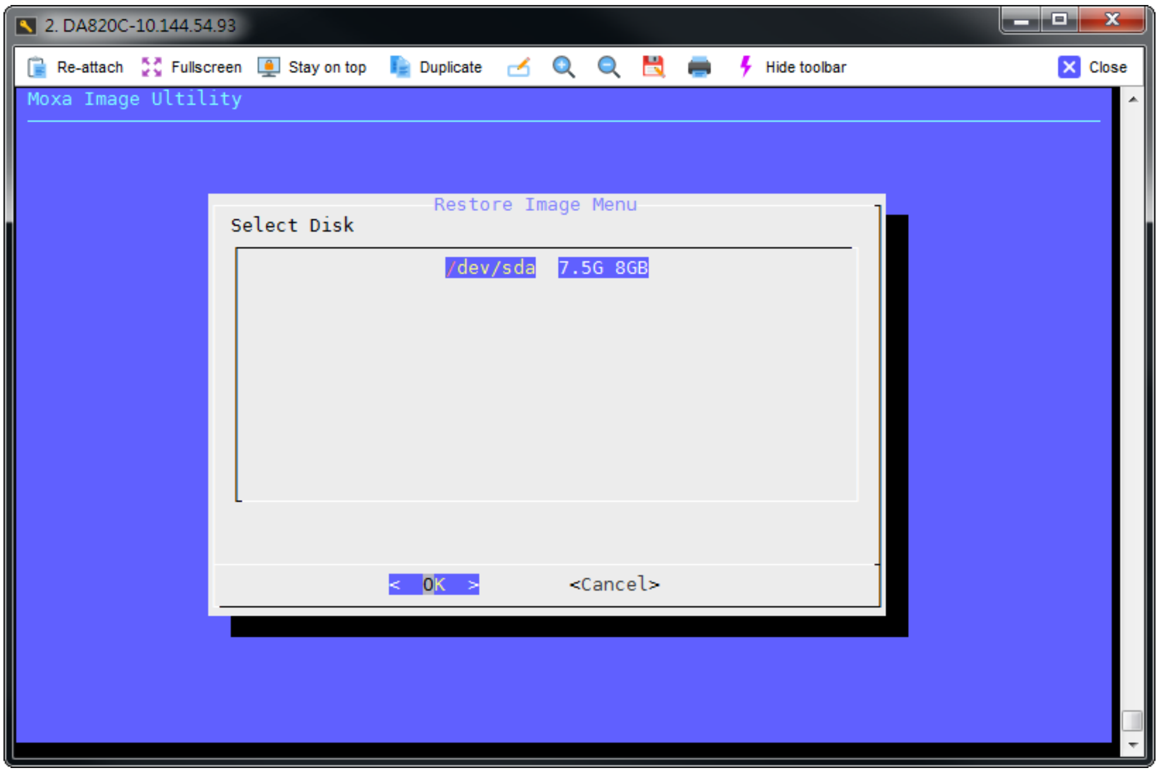
If you want to backup raw image, please select "Raw Image Backup".

If you want to backup image and resize disk size, please select "Resized Image Backup".

For example, to select "Resized Image Backup".

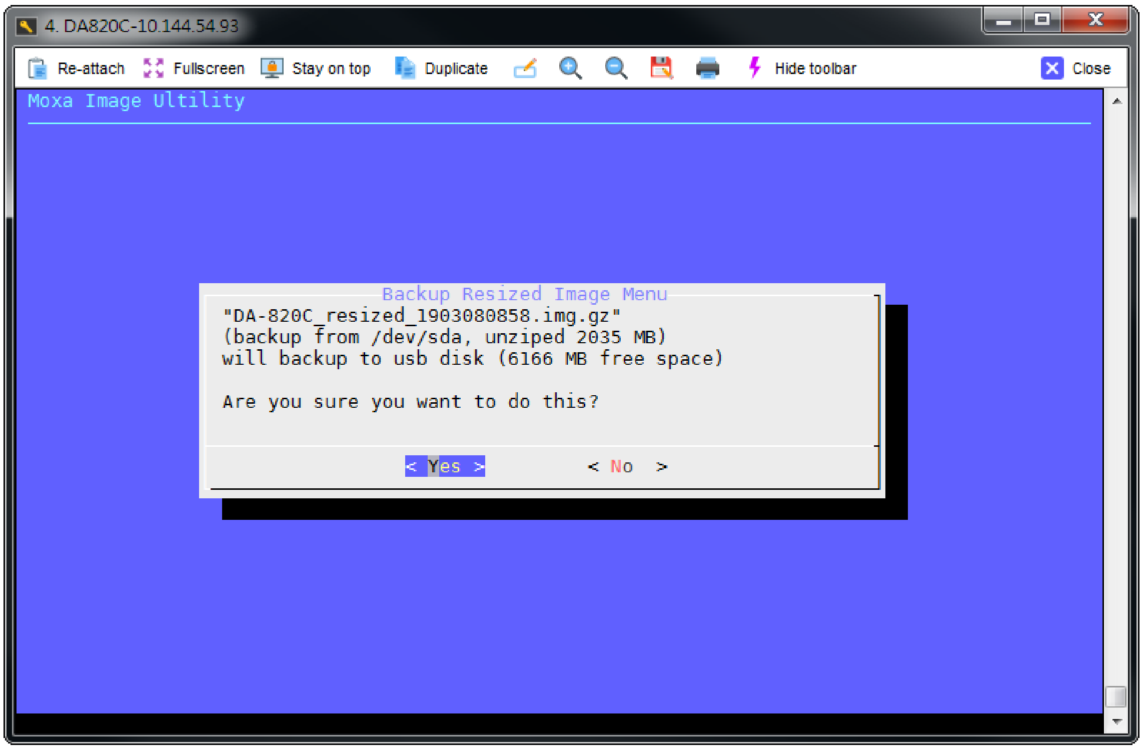

```
2. DA820C-10.144.54.93
Re-attach  Fullscreen  Stay on top  Duplicate  Hide toolbar  Close
Moxa Image Ultility

                 Backup Image Menu
 Select Backup Mode
                0  Raw Image Backup
                1  Resized Image Backup

              <  OK  >          <Cancel>
```

To select the target storage disk to backup:



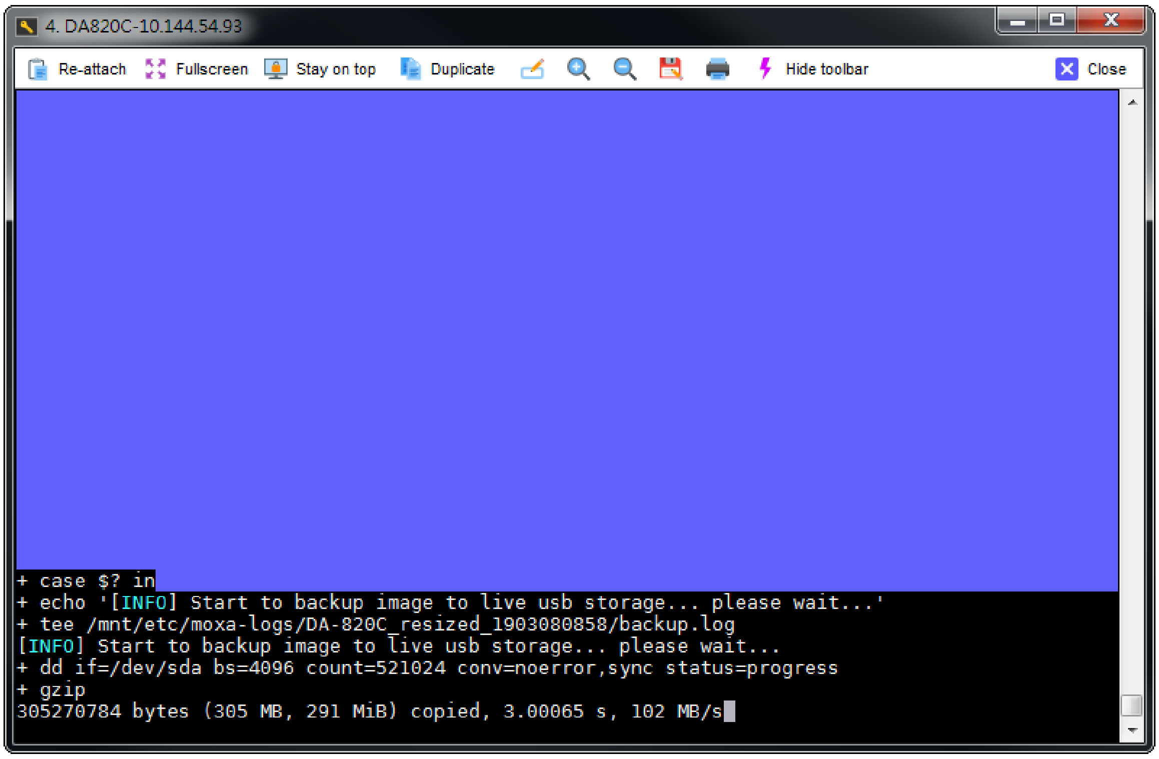
```
Moxa Image Ultility

                 Restore Image Menu
 Select Disk

            /dev/sda   7.5G 8GB

          <  OK  >          <Cancel>
```

The check box will show the backup information, including image size and the rest free space of USB live disk.



```
Moxa Image Ultility

              Backup Resized Image Menu
"DA-820C_resized_1903080858.img.gz"
(backup from /dev/sda, unziped 2035 MB)
will backup to usb disk (6166 MB free space)

Are you sure you want to do this?

          < Yes >          <  No  >
```

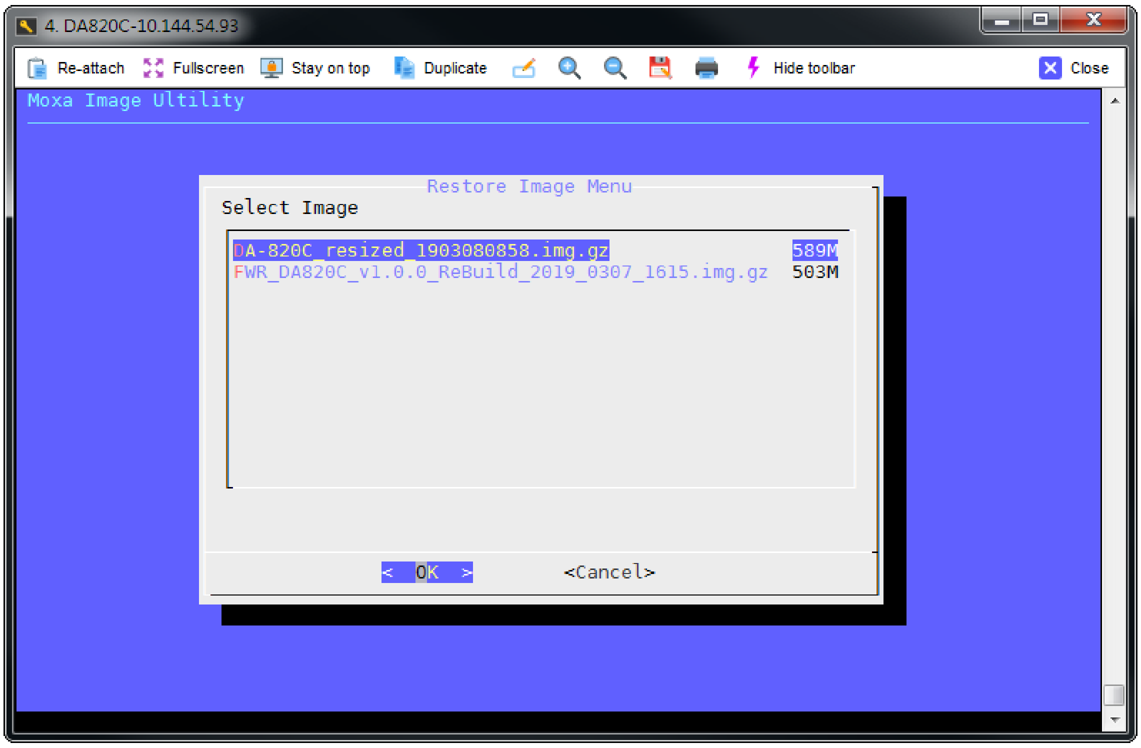
Select “Yes” button and start to backup disk.



```
+ case $? in
+ echo '[INFO] Start to backup image to live usb storage... please wait...'
+ tee /mnt/etc/moxa-logs/DA-820C_resized_1903080858/backup.log
[INFO] Start to backup image to live usb storage... please wait...
+ dd if=/dev/sda bs=4096 count=521024 conv=noerror,sync status=progress
+ gzip
305270784 bytes (305 MB, 291 MiB) copied, 3.00065 s, 102 MB/s
```

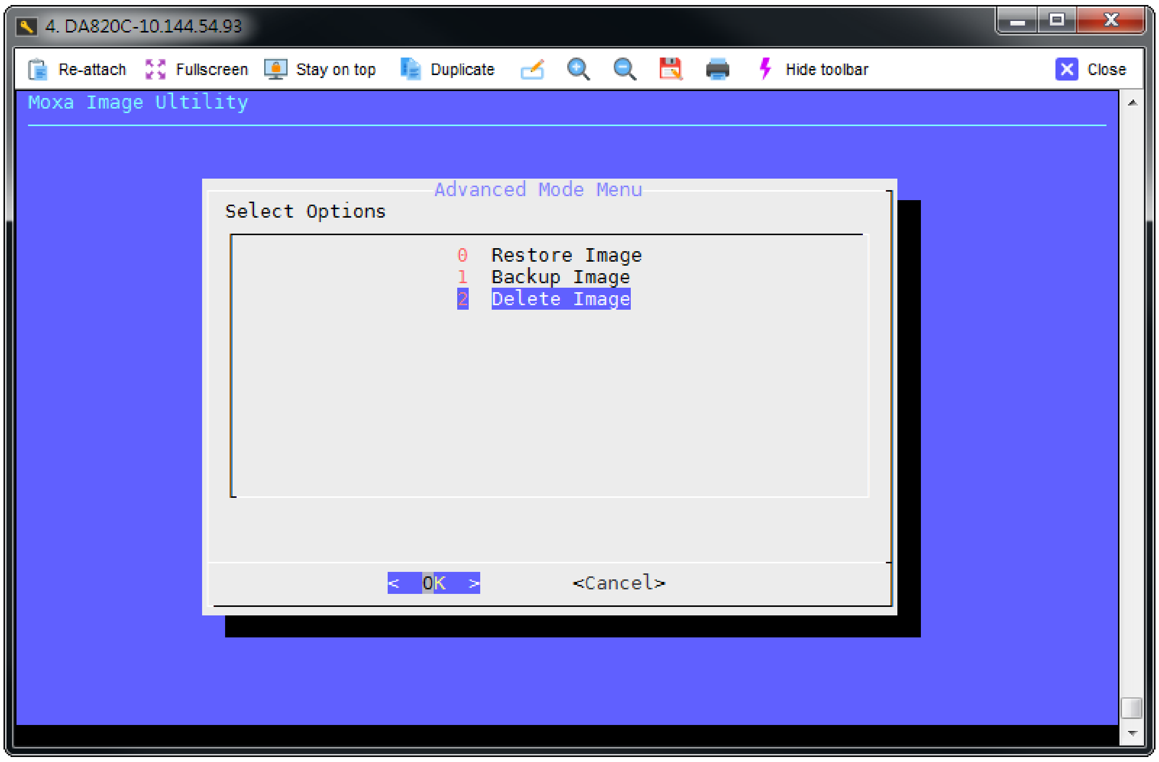
After backup process, the backup image was stored at live USB storage.

To select restore page, you can find out the backup image is located on USB storage.

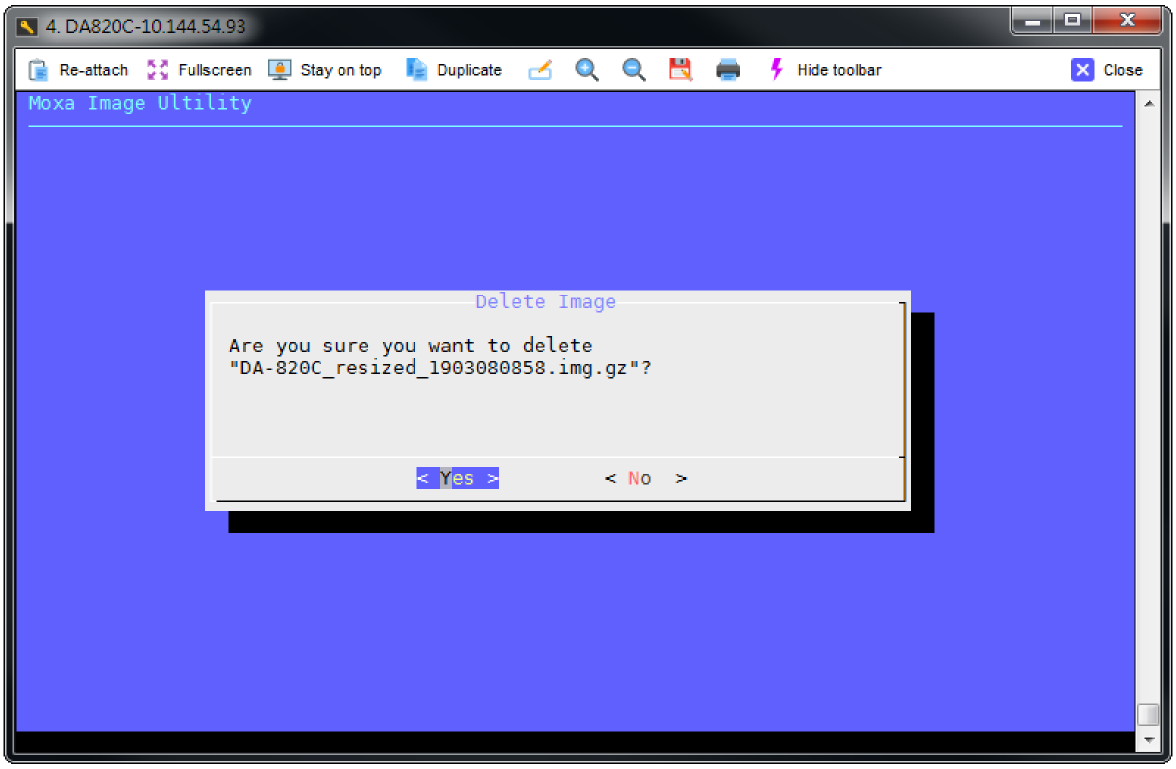


```
Moxa Image Ultility

                 Restore Image Menu
Select Image

DA-820C_resized_1903080858.img.gz                589M
FWR_DA820C_v1.0.0_ReBuild_2019_0307_1615.img.gz  503M

         <  OK  >          <Cancel>
```

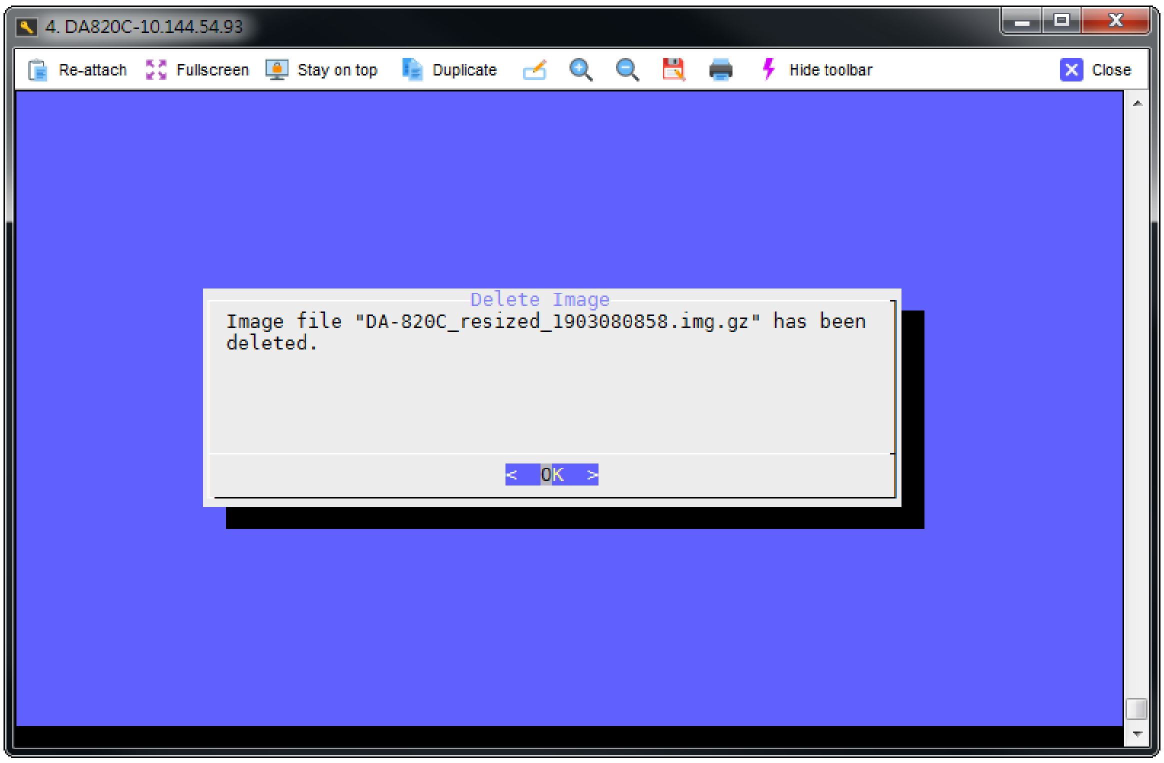
If you need to delete backup images in live USB storage, to select "Delete image" in advanced mode to delete image.



```
Moxa Image Ultility

                Advanced Mode Menu
 Select Options

                      0  Restore Image
                      1  Backup Image
                      2  Delete Image

              <  OK  >        <Cancel>
```



```
Moxa Image Ultility

                Delete Image

 Are you sure you want to delete
 "DA-820C_resized_1903080858.img.gz"?

             < Yes >        <  No  >
```



```
Delete Image
Image file "DA-820C_resized_1903080858.img.gz" has been
deleted.

< OK >
```

5. Advanced Configuration

Checking the Linux Version

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

```
moxa@Moxa:~$ uname -a
Linux Moxa 4.9.0-12-amd64 #1 SMP Debian 4.9.210-1 (2020-01-20) x86_64
GNU/LinuxGNU/Linux
```

Checking the Version

Querying the Firmware Version

The **kversion** program prints version information on the Linux system release, which is controlled by Moxa. Use the **-a** option to check the build date. The build date format is YYYYMMDDHHmm. You can use this program to check the version of the released image for troubleshooting issues. For example, the build date, 19080617 means it was built on 2019/08/06 at 17:00 hrs.

According to EPPROM information on IO board, use the -s option to show the PCBA serial number, -t option to show PCBA type number (00-CPU Board, 01-Carrier Board, 02-Riser card), -p option to show PCBA version, and -v option to show verbose information.

```
root@Moxa:/home/moxa# kversion
V2403C-KL5-T version 1.0
root@Moxa:/home/moxa# kversion -a
V2403C-KL5-T version 1.0 Build 19120515
```

Device Suspend

The V2403C supports ACPI S3 (suspend to RAM) function. To use this function, you should enable option S3 in the BIOS and then use the systemctl suspend command as follows:

```
root@Moxa:/home/moxa# systemctl suspend
```

When the device suspend function is in effect, press the power button to wake up the computer.

If you login in as administrator (root) in X windows, you can use **System → Shutdown → Suspend** to suspend your device. Note that this option is not available to non-root users.

Some components on Moxa's embedded computers may need to be reset after resuming. You can write a simple script and save it in the **/usr/lib/pm-utils/sleep.d/** directory to complete this procedure. For example, you can create the following script for your application.

```
#!/bin/sh

case "$1" in
        hibernate|suspend)
                    echo "close AP and tty ports which are opened"
                echo "operations before serial ports suspend"
                ;;
        thaw|resume)
                    echo "restart AP"
                echo "operations after serial ports resume"
                ;;
        *) exit $NA
                ;;
esac
```

NOTE

For more information, refer to the systemd suspend service man page at:
https://www.freedesktop.org/software/systemd/man/systemd-suspend.service.html

Wake on LAN

The V2403C supports wake on LAN (WoL), a feature used to wake up a device from suspend (S3) and shutdown (S5). To check the WOL support on an Ethernet port x, run the ethtool enpx command, where enpx is the network interface name.

```
root@Moxa:/home/moxa# apt update && apt install ethtool
root@Moxa:/home/moxa# ethtool enp0s31f6
Settings for enp0s31f6:
        Supported ports: [ TP ]
        Supported link modes:   10baseT/Half 10baseT/Full
                                100baseT/Half 100baseT/Full
                                1000baseT/Full
        Supported pause frame use: No
        Supports auto-negotiation: Yes
        Advertised link modes:  10baseT/Half 10baseT/Full
                                100baseT/Half 100baseT/Full
                                1000baseT/Full
        Advertised pause frame use: No
        Advertised auto-negotiation: Yes
        Speed: 1000Mb/s
        Duplex: Full
        Port: Twisted Pair
        PHYAD: 1
        Transceiver: internal
        Auto-negotiation: on
        MDI-X: on (auto)
        Supports Wake-on: pumbg
        Wake-on: g
        Current message level: 0x00000007 (7)
                               drv probe link
        Link detected: yes
```

The default WoL option is g (wake on magic packet). If the WoL setting is not g, we suggest that you modify the setting with the **ethtool -s enpx wol g** command.

The following example illustrates how to wake up a computer from suspend (S3):

1. On Moxa's embedded computer:
   - Enable the S3 option in the BIOS.
   - Get the MAC address by issuing the **ifconfig ethx** command (***x*** is the port number).
   - Suspend to RAM using the **pm-suspend --quirk-s3-bios** command.
2. A remote computer:

   Issue the **etherwake -b <*mac-address-of-the-embedded-computer*>** command to wake up the computer as per the following example:

```
etherwake -b 00:90:e8:00:d7:38
```

The following example illustrates how to wake up a computer from shutdown (S5):

1. Moxa's embedded computer:
   - Shut down the computer using the **shutdown -h now** command.

2. A remote computer:

   Issue the **etherwake -b <*mac-address-of-the-embedded-computer*>** command to wake up the computer as per the following example:

```
etherwake -b 00:90:e8:00:d7:38
```

Default Network Interface Name

Debian 9 Stretch adopts the systemd predictable network interface naming convention by default. The network interface name is no longer just based on ethX. The interface names depend on the hardware design and physical connections. You may see different types of interface names, for example:

1. Names incorporating Firmware/BIOS-provided index numbers for onboard devices (example: eno1)
2. Names incorporating Firmware/BIOS-provided PCI Express hotplug slot index numbers (example: ens1)
3. Names incorporating physical/geographical location of the connector of the hardware (example: enp2s0)
4. Names incorporating the interfaces' MAC address (example: enx78e7d1ea46da)
5. Classic, unpredictable kernel-native ethX naming (example: eth0)

For more details, refer to:
https://www.freedesktop.org/wiki/Software/systemd/PredictableNetworkInterfaceNames/

The V2403C default LAN port and network interface name mapping is as given below:

LAN port	Network Interface Name
LAN1	enp0s31f6
LAN2	enp6s0
LAN3	enp7s0
LAN4	enp8s0

Renaming the Network Interfaces

You can use the udev rules to rename the network interfaces. For example, if you would like to rename them to the classic "ethX" naming, you can create a rules file named **/etc/udev/rules.d/70-persistent-net.rules**, and edit the content of the file as given below:

Renaming the interfaces using the MAC address (example)

```
SUBSYSTEM=="net", ACTION=="add", ATTR{address}=="00:90:e8:00:d7:38",
NAME="eth0"
SUBSYSTEM=="net", ACTION=="add", ATTR{address}=="00:90:e8:00:d7:58",
NAME="eth1"
```

Viewing the Product Serial Number

The product information can be obtained using the dmidecode command as shown in the following example.

```
moxa@Moxa:~$ sudo dmidecode -t 1
# dmidecode 3.0
Getting SMBIOS data from sysfs.
SMBIOS 3.0.0 present.

Handle 0x0001, DMI type 1, 27 bytes
System Information
        Manufacturer: Moxa
        Product Name: V2403C
        Version:
        Serial Number: 123456789
        UUID: 12345678-1234-5678-90AB-CDDEEFAABBCC
        Wake-up Type: Power Switch
        SKU Number:
         Family:
```

Real-time Clock (RTC)

The V2403C supports standard Linux simple RTC control. The device node is located at **/dev/rtc**. You must include the file **<linux/rtc.h>**.

1. Function RTC_RD_TIME

   Reads time information from the RTC and returns the value to the third argument (**struct rtc_time *time**).

```
int ioctl(fd, RTC_RD_TIME, struct rtc_time *time);
```

2. Function RTC_SET_TIME

   Sets the RTC time. Argument 3 will be passed to RTC.

```
int ioctl(fd, RTC_SET_TIME, struct rtc_time *time);
```

MXview

Moxa's MXview network management software is designed for configuring, monitoring, and diagnosing networking devices in industrial networks. MXview provides an integrated management platform that can discover networking devices and SNMP/IP devices installed on subnets.

You can find more information about Moxa MXview at:

https://www.moxa.com/en/products/industrial-network-infrastructure/network-management-software/mxview-series

Setting Up the SNMP and LLDP Daemons Using moxa-snmpd-support-package

From MXview version 3.0 onwards, the moxa-snmpd-support-package is installed by default on the Moxa platform. The package includes snmpd and lldpd settings for MXview (based on the model name).

If you want to install **moxa-snmpd-support-package**, run the following commands:

```
moxa@Moxa:~$apt-get update
moxa@Moxa:~$apt-get install moxa-snmpd-support-package
```

Installing MXview 3.x

MXview 3.x is available for download on Moxa website:

https://www.moxa.com/en/products/industrial-network-infrastructure/network-management-software/mxview-series

Please follow the instructions to install MXview network management software.

For example, you can press "Topology" -> "Scan Range" to scan devices, then use "Topology" -> "Auto Topology" to draw the topology diagram. For more information, please refer to MXview user manual.

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 RS-422. Use the **mx-uart-ctl** command to change the operation mode.

mx-uart-ctl -p *<port_number>* **-m** *<uart_mode>*

port_number:	0,1,2,...
uart_ mode:	As in the following table

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

For example, use the following commands to set port 0 to RS-485 4-wire mode:

```
root@Moxa:/home/moxa# mx-uart-ctl -p 0
Current uart mode is RS232 interface.
root@Moxa:/home/moxa# mx-uart-ctl -p 0 -m 2
Set OK.
```

Changing the Terminal Settings

The **stty** command is used to view and modify the serial terminal settings.

Displaying All Settings

The following example shows the **stty** command options used to display all the serial terminal settings.

```
root@Moxa:/home/moxa# sudo stty -a -F /dev/ttyM0
speed 9600 baud; rows 0; columns 0; line = 0;
intr = ^C; quit = ^\; erase = ^?; kill = ^U; eof = ^D; eol = <undef>; eol2 =
<undef>; swtch = <undef>; start = ^Q; stop = ^S; susp = ^Z; rprnt = ^R; werase
= ^W;
lnext = ^V; discard = ^O; min = 1; time = 0;
-parenb -parodd -cmspar 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 -flusho -extproc
```

Configuring Serial Settings

The following example changes the baudrate to 115200.

```
root@Moxa:/home/moxa# sudo stty 115200 -F /dev/ttyM0
```

```
root@Moxa:/home/moxa# sudo stty -a -F /dev/ttyM0
speed 115200 baud; rows 0; columns 0; line = 0;
intr = ^C; quit = ^\; erase = ^?; kill = ^U; eof = ^D; eol = <undef>; eol2 =
<undef>; swtch = <undef>; start = ^Q; stop = ^S; susp = ^Z; rprnt = ^R; werase
= ^W;
lnext = ^V; discard = ^O; min = 1; time = 0;
-parenb -parodd -cmspar 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 -flusho -extproc
```

Note

Detailed information on the stty utility is available at:

http://www.gnu.org/software/coreutils/manual/coreutils.html

DIP Switch

There are two DIP switches on the inner of the V-2406C. They are controller to switch cellular/Wi-Fi mode. In cellular mode, user can control form factor power on/off. In Wi-Fi mode, user cannot control power. 0/LOW (ON) for the Wi-Fi module, 1/HIGH (OFF) for the Cellular module.

The DIP switches device file is located at Linux file system path:

DIP slot 1: /sys/class/gpio/gpio455/value

DIP slot 2: /sys/class/gpio/gpio487/value

DIP switches status can be accessed via device node. These are the examples to show the DIP switches status.

Example to show the DIP switches status:

```
root@Moxa:/home/moxa# cat /sys/class/gpio/gpio455/value
0
root@Moxa:/home/moxa# cat /sys/class/gpio/gpio487/value
0
root@Moxa:/home/moxa# setinterface /dev/ttyM0
Now setting is RS485-2W mode
```

Moxa Module Control

The moxa-module-control utility is used to control modules, including power control, module detection, initialize setting, and SIM slot switching (for cellular modules), on a platform.

```
Moxa module control utility
Version: 1.3.1
---------------------------
Usage:
        mx-module-ctl [Options]

Operations:
        -v,--version
                Show utility version
        -l,--list
                List module slots
        -s,--slot <module_slot_id>
                Select slot
        -p,--power on|off
                Power on/off module
        -r,--reset on|off
                Reset module
        -i,--sim 1|2
                Select sim card slot

Example:
        Power on module 1
        # mx-module-ctl -s 1 -p on

        Reset module 2
        # mx-module-ctl -s 2 -r on

        Select SIM 2 for module 1
        # mx-module-ctl -s 1 -i 2
```

Switching the SIM Card

The following is an example for switching SIM card by mx-module-ctl. Please notice that the 'SIM card select' only for switching SIM card slot, for some cellular modules, they need to do power cycle to re-attach SIM card.

```
# Example for module slot 1 SIM card switch
## Switch to SIM card slot 1 (upper side)
root@Moxa:/home/moxa# mx-module-ctl -s 1 -i 1

## Switch to SIM card slot 2 (lower side)
root@Moxa:/home/moxa# mx-module-ctl -s 1 -i 2
```

Powering On/Off the Cellular Module

Example to control cellular module power by mx-module-ctl

```
# Power on cellular module slot 1
root@Moxa:/home/moxa# mx-module-ctl -s 1 -p on
# Power off cellular module slot 1
root@Moxa:/home/moxa# mx-module-ctl -s 1 -p off

# Power on cellular module slot 2
root@Moxa:/home/moxa# mx-module-ctl -s 2 -p on
# Power off cellular module slot 2
root@Moxa:/home/moxa# mx-module-ctl -s 2 -p off
```

Moxa Digital I/O Control

Digital Output channels can be set to high or low. The channels are controlled by **moxa-dio-ctl**, which is a C library for getting and setting the status of the digital-input/output ports.

Return all the GPIO sysfs files have been exported by /etc/systemd/system/multi-user.target.wants/V2403c_platform_init.service at boot sequence. You don't need to export the GPIO entry. The default status of the digital inputs is **high**.

```
root@Moxa:/home/moxa# mx-dio-ctl
action type is unset
Usage:
        mx-dio-ctl <-i|-o <#port number> [-s <#state>]>

OPTIONS:
        -i <#DIN port number>
        -o <#DOUT port number>
        -s <#state>
                Set state for target DOUT port
                0 --> LOW
                1 --> HIGH

Example:
        Get value from DIN port 0
        # mx-dio-ctl -i 0
        Get value from DOUT port 0
        # mx-dio-ctl -o 0

        Set DOUT port 0 value to LOW
        # mx-dio-ctl -o 0 -s 0
        Set DOUT port 0 value to HIGH
        # mx-dio-ctl -o 0 -s 1
```

To set DO-0 to status **high**, do the following:

```
root@Moxa:/home/moxa# mx-dio-ctl -o 0 -s 1
DOUT port 0 state: 1
```

Connect DO-0 to DI-0 to read the DI-0 status as follows:

```
root@Moxa:/home/moxa# mx-dio-ctl -i 0
DIN port 0 state: 1
```

To set DO-0 to status **low**, do the following:

```
root@Moxa:/home/moxa# mx-dio-ctl -o 0 -s 0
DOUT port 0 state: 0
```

Connect DO-0 to DI-0 to read the DI-0 status as follows:

```
root@Moxa:/home/moxa# mx-dio-ctl -i 0
DIN port 0 state: 0
```

Moxa Cellular Utility

The **cell_mgmt** utility is used to manage the cellular module in the computer. You must use the sudo command or run the cell_mgmt command with root permission.

Manual Page

```
Usage:
        /usr/sbin/cell_mgmt [-i <module id>] [-s <slot id>] <OPTIONS>

OPTIONS
        -i <module id>
                Module identifier, start from 0 and default to 0.
        -s <slot id>
                Slot identifier, start from 1 and default value depends
                on module interface.
                example: module 0 may in slot 2
        modules
                Shows module numbers supported.
        slot
                Shows module slot id
        interface [interface id]
                Switching and checking module interface(s)
        start [OPTIONS]
                Start network.

                OPTIONS:
                Phone - Phone number (especially for AT based modules)
                Auth - Authentication type(CHAP|PAP|BOTH), default=NONE.
                Username
                Password

                example:
                        cell_mgmt start
                        cell_mgmt start Phone=*99#
        stop
                Stop 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), Default=1.
        set_apn <APN>
                Set APN to configuration file and PDP profile.
                And use this APN as the default one.
        check_carrier
                Check current carrier.
        switch_carrier <Verizon|ATT|Sprint|Generic>
                Switching between carrier profiles.
        m_info (deprecated)
                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

Before dialing, ensure that the APN (Access Point Name) is set correctly, and the module is attached with the base station.

1. Unlock the PIN code if SIM locked by a PIN code

   Use cell_mgmt sim_status to check SIM card status and use cell_mgmt unlock_pin <PIN> to unlock SIM card if "SIM-PIN"

```
root@Moxa:/home/moxa# cell_mgmt sim_status
+CPIN: READY
```

2. Set the APN with cell_mgmt set_apn <APN>, this command will update the APN in profile ID 1

```
root@Moxa:/home/moxa# cell_mgmt set_apn internet
old APN=, new APN=internet
```

3. Check if the service attached with correct APN

   PS (packet-switched) should be attached for network connection.

```
root@Moxa:/home/moxa# cell_mgmt attach_status
CS: attached
PS: attached
```

4. Check cellular signal strength is good or not

```
root@Moxa:/home/moxa# cell_mgmt signal
4G Level 4 (Good)
```

5. Dial up with **cell_mgmt start** and check connection status with cell_mgmt status

```
root@Moxa:/home/moxa# cell_mgmt start
PIN code: Disabled or verified
Saving state... (PPP_ISP_NAME: wvdial-2)

root@Moxa:/home/moxa# cell_mgmt status
Status: connected
PPPIFName: ppp0
IFName: ppp0
IP: 100.72.98.84
IPRemote: 10.64.64.64
DNS: 61.31.233.1 8.8.8.8
```

The **cell_mgmt dial-up** function will automatically set the DNS and default gateway of the computer.

Cellular Management

cell_mgmt start

To start a network connection, use the default cellular module of the computer (using **cell_mgmt interface** to verify which module is selected by default if the computer supports multiple modules).

If you run the cell_mgmt start command with the APN, Username, Password, and PIN, all the configurations will be written into the configuration file **/etc/moxa-cellular-utils/moxa-cellular-utils.conf**.

This information is then used when you run the command without specifying the options.

Usage: **cell_mgmt start**

cell_mgmt stop

Stops/disables the network connection on the cellular module of the computer by **cell_mgmt stop** and check current status with **cell_mgmt status**.

```
root@Moxa:/home/moxa# cell_mgmt stop
Clearing state...
root@Moxa:/home/moxa# cell_mgmt status
Status: disconnected
```

cell_mgmt restart

```
Restarts the network connection on the cellular module of the computer.
root@Moxa:/home/moxa# cell_mgmt restart
Clearing state...
PIN code: Disabled or verified
Saving state... (PPP_ISP_NAME: wvdial-2)
root@Moxa:/home/moxa# cell_mgmt status
Status: connected
PPPIFName: ppp0
IFName: ppp0
IP: 100.72.98.84
IPRemote: 10.64.64.64
DNS: 61.31.233.1 8.8.8.8
```

cell_mgmt status

Provides information on the status of the network connection.

```
root@Moxa:/home/moxa# cell_mgmt status
Status: connected
PPPIFName: ppp0
IFName: ppp0
IP: 100.72.98.84
IPRemote: 10.64.64.64
DNS: 61.31.233.1 8.8.8.8
```

cell_mgmt sim_status

Query sim card status.

```
root@Moxa:/home/moxa# cell_mgmt sim_status
+CPIN: READY
```

cell_mgmt signal

Provides the cellular signal strength.

```
root@Moxa:/home/moxa# cell_mgmt signal
4G Level 4 (Good)
```

cell_mgmt operator

Provides information on the cellular service provider.

```
root@Moxa:/home/moxa# cell_mgmt operator
TW Mobile
```

Cellular Module

cell_mgmt -s <slot>

Slot identifier, start from 1 and default value depends on module interface.

```
root@Moxa:/home/moxa# cell_mgmt -s 1 module_info
SLOT: 1
Module: Sierra Wireless WP7607
WWAN_node: wwan0
AT_port: /dev/ttyUSB2
GPS_port: /dev/ttyUSB1
QMI_port: /dev/cdc-wdm0
Modem_port: NotSupport
AT_port (resvered): NotSupport

root@Moxa:/home/moxa# cell_mgmt -s 2 module_info
SLOT: 2
Module: Huawei ME909s-821
WWAN_node: enp0s20f0u7c2
AT_port: /dev/ttyUSB3
GPS_port: NotSupport
QMI_port: NotSupport
Modem_port: /dev/ttyUSB0
AT_port (resvered): NotSupport
```

cell_mgmt module_info

Provides information of the cellular module (AT port, GPS port, QMI port, and module name, etc.).

```
root@Moxa:/home/moxa# cell_mgmt module_info
SLOT: 2
Module: Huawei ME909s-821
WWAN_node: enp0s20f0u7c2
AT_port: /dev/ttyUSB2
GPS_port: NotSupport
QMI_port: NotSupport
Modem_port: /dev/ttyUSB0
AT_port (resvered): NotSupport
```

cell_mgmt interface [*id*]

Used to view the supported modules and default module on the computer with their IDs. Change the default module by specified the ID.

```
root@Moxa:/home/moxa# cell_mgmt interface
[0] enp0s20f0u7c2    <Current>
```

cell_mgmt power_cycle

Power cycle the cellular module in the computer. Some kernel message for module reloaded may be popped out.

```
root@Moxa:/home/moxa# cell_mgmt power_cycle
Clearing state...

[kernel message]
[ 3347.762575] usb 1-7: USB disconnect, device number 5
[ 3347.762799] cdc_ether 1-7:2.0 enp0s20f0u7c2: unregister 'cdc_ether' usb-
0000:00:14.0-7, CDC Ethernet Device
[ 3347.795175] option1 ttyUSB0: GSM modem (1-port) converter now disconnected
from ttyUSB0
[ 3347.795184] option 1-7:2.2: device disconnected
[ 3347.795273] option1 ttyUSB1: GSM modem (1-port) converter now disconnected
from ttyUSB1
[ 3347.795281] option 1-7:2.3: device disconnected
[ 3347.795374] option1 ttyUSB2: GSM modem (1-port) converter now disconnected
from ttyUSB2
[ 3347.795384] option 1-7:2.4: device disconnected
[ 3347.795514] option1 ttyUSB3: GSM modem (1-port) converter now disconnected
from ttyUSB3
[ 3347.795523] option 1-7:2.5: device disconnected
[ 3347.795645] option1 ttyUSB4: GSM modem (1-port) converter now disconnected
from ttyUSB4
[ 3347.795656] option 1-7:2.6: device disconnected
[ 3355.207128] usb 1-7: new high-speed USB device number 6 using xhci_hcd
[ 3355.348597] usb 1-7: New USB device found, idVendor=12d1, idProduct=15c1
[ 3355.348605] usb 1-7: New USB device strings: Mfr=1, Product=2,
SerialNumber=3
[ 3355.348610] usb 1-7: Product: HUAWEI Mobile V7R11
[ 3355.348615] usb 1-7: Manufacturer: Huawei Technologies Co., Ltd.
[ 3355.348619] usb 1-7: SerialNumber: 0123456789ABCDEF
[ 3355.353522] cdc_ether 1-7:2.0 usb0: register 'cdc_ether' at usb-
0000:00:14.0-7, CDC Ethernet Device, 02:1e:10:1f:00:00
[ 3355.354373] option 1-7:2.2: GSM modem (1-port) converter detected
[ 3355.354669] usb 1-7: GSM modem (1-port) converter now attached to ttyUSB0
[ 3355.355526] option 1-7:2.3: GSM modem (1-port) converter detected
[ 3355.355805] usb 1-7: GSM modem (1-port) converter now attached to ttyUSB1
[ 3355.356298] option 1-7:2.4: GSM modem (1-port) converter detected
[ 3355.356557] usb 1-7: GSM modem (1-port) converter now attached to ttyUSB2
[ 3355.356888] option 1-7:2.5: GSM modem (1-port) converter detected
[ 3355.356944] usb 1-7: GSM modem (1-port) converter now attached to ttyUSB3
[ 3355.357104] option 1-7:2.6: GSM modem (1-port) converter detected
[ 3355.357156] usb 1-7: GSM modem (1-port) converter now attached to ttyUSB4
[ 3355.368589] cdc_ether 1-7:2.0 enp0s20f0u7c2: renamed from usb0
[ 3355.403015] IPv6: ADDRCONF(NETDEV_UP): enp0s20f0u7c2: link is not ready
```

cell_mgmt check_carrier

This command helps to check if current carrier matched with the service (SIM card) provider.

```
root@Moxa:/home/moxa# cell_mgmt check_carrier
----------Carrier Info----------
preferred firmware=02.22.00.00
preferred carrier name=GENERIC
preferred carrier config=GENERIC_002.048_000
firmware=02.22.00.00
carrier name=GENERIC
carrier config=GENERIC_002.048_000
available carriers=ATT GENERIC VERIZON
--------------------------------
```

cell_mgmt switch_carrier <*carrier*>

Some modules provide multiple carrier support and use this command to switch between carriers. It may take some time (depends on module's mechanism) to switch between carriers.

```
root@Moxa:/home/moxa#  cell_mgmt switch_carrier
----------switch_carrier------------
Usage:
       switch_carrier <Verizon|ATT|Generic>

root@Moxa:/home/moxa# cell_mgmt switch_carrier ATT
----------switch_carrier------------
cmd=AT!IMPREF="ATT"

OK

OK

OK
wait for module reset ...

OK

root@Moxa:/home/moxa# cell_mgmt check_carrier
----------Carrier Info----------
preferred firmware=02.22.00.00
preferred carrier name=ATT
preferred carrier config=ATT_002.051_000
firmware=02.22.00.00
carrier name=ATT
carrier config=ATT_002.051_000
available carriers=ATT GENERIC VERIZON
--------------------------------
```

cell_mgmt at <'AT_COMMAND'>

Used to input an AT command. For example, use AT commands like ATI or AT+CSQ:

```
root@Moxa:/home/moxa# cell_mgmt at 'ATI'
ATI
Manufacturer: Sierra Wireless, Incorporated
Model: WP7607
Revision: SWI9X07Y_02.28.03.03 000000 jenkins 2019/05/21 03:33:04
IMEI: 359779080114230
IMEI SV:  6
FSN: VN829285061610
+GCAP: +CGSM

OK
root@Moxa:/home/moxa# cell_mgmt at 'AT+CSQ'
AT+CSQ
+CSQ: 99,99

OK
```

Watch Dog Timer (WDT)

Introduction

The WDT works like a watchdog function and can be enabled or disabled. When the WDT function is enabled and the application does not acknowledge it, the system will reboot. The watchdog driver is load with default timeout 60 seconds. The watchdog application should acknowledge in 60 seconds.

How the WDT Works

Debian project supports a watchdog daemon that can be installed from the APT repository using the apt-get command. The watchdog daemon checks if your system is still working. If programs are no longer executed, it will perform the hard reset of the system. The standard watchdog driver and package have been installed in the V-2406C.

To enable it, first modify the **/etc/watchdog.conf** to remove the '**#**' in front of the "**watchdog-device**" setting

```
…
watchdog-device = /dev/watchdog
…
```

Then enable the **watchdog** service via **systemctl**

moxa@Moxa:~$ sudo systemctl enable watchdog

The watchdog configuration file is located in **/etc/watchdog.conf**. You can configure the watchdog settings in this file based on your system requirements. The acknowledgement interval can be set between 2 seconds and 58 seconds. Currently we configure the watchdog daemon to acknowledge in 29 seconds so that it can acknowledge twice before the watchdog timer times out and the daemon goes to sleep. The **realtime** mode setting is to lock the watchdog timer into the computer memory, so that it is never swapped out to prevent delay in acknowledging the watchdog. The **priority** sets the schedule priority for the realtime mode.

```
…
interval                 = 29
realtime                 = yes
priority                 = 1
…
```

If you want to remove it from systemd, you can use this command.

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

Check the status of the watchdog daemon.

```
moxa@Moxa:~# sudo systemctl status watchdog
```

Watchdog Device IOCTL Commands

IOCTL	**WDIOC_GETSUPPORT**
Description	This returns the support of the card itself
Input	None
Output	(struct watchdog_info *) arg
Return	On success, return 0. Otherwise, return < 0 value.

IOCTL	**WDIOC_GETSTATUS**
Description	This returns the status of the card
Input	None
Output	(int *)arg
Return	On success, return 0. Otherwise, return < 0 value.

IOCTL	WDIOC_GETBOOTSTATUS
Description	This returns the status of the card that was reported at bootup.
Input	None
Output	(int *)arg)
Return	On success, return 0. Otherwise, return < 0 value.

IOCTL	WDIOC_SETOPTIONS
Description	This lets you set the options of the card. You can either enable or disable the card this way.
Input	None
Output	(int *)arg)
Return	On success, return 0. Otherwise, return < 0 value.

IOCTL	WDIOC_KEEPALIVE
Description	This pings the card to tell it not to reset your computer.
Input	None
Output	None
Return	On success, return 0. Otherwise, return < 0 value.

IOCTL	WDIOC_SETTIMEOUT
Description	Set the watchdog timeout
Input	arg: 2 ~ 255 seconds
Output	None
Return	On success, return 0. Otherwise, return < 0 value.

IOCTL	WDIOC_GETTIMEOUT
Description	Get the current watchdog timeout.
Input	None
Output	arg: 2 ~ 255 seconds
Return	On success, return 0. Otherwise, return < 0 value.

Example

The example file **watchdog-simple.c** acknowledges (acks) the watchdog every 10 seconds.

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

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