Moxa Industrial Linux 3.x (Debian 11) Manual for Arm-based Computers

Version 3.0, November 2025

www.moxa.com/products

Moxa Industrial Linux 3.x (Debian 11) Manual for Arm-based Computers

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

Copyright Notice

Trademarks

Disclaimer

- Information in this document is subject to change without notice and does not represent a commitment on the part of Moxa.
- Moxa provides this document as is, without warranty of any kind, either expressed or implied, including, but not limited to, its particular purpose. Moxa reserves the right to make improvements and/or changes to this manual, or to the products and/or the programs described in this manual, at any time.
- Information provided in this manual is intended to be accurate and reliable. However, Moxa assumes no responsibility for its use, or for any infringements on the rights of third parties that may result from its use.
- This product might include unintentional technical or typographical errors. Changes are periodically made to the information herein to correct such errors, and these changes are incorporated into new editions of the publication.

Technical Support Contact Information

www.moxa.com/support

Table of Contents

1. Introduction

Moxa Industrial Linux 3

Moxa Industrial Linux 3 (MIL3) is an industrial-grade Linux distribution developed and maintained by Moxa to address the security, reliability, and long-term support needs of industrial automation systems such as transportation, energy, oil and gas, and manufacturing.

MIL3 is based on Debian 11 with kernel 5.10 and integrated with several feature sets designed to strengthen and accelerate user application development as well as ensure system reliability and security.

Secure and Standard Models

MIL3 provides two security levels in the form of **standard** and **secure** models.

- **Standard models** use the default Debian 11 security configuration, allowing users to customize and build their own security solutions.
- **Secure models** offer a hardened configuration with Secure Boot, predefined security settings, and preinstalled security tools/utilities. They are secure-by-default and aligned with recognized cybersecurity standards.

Cybersecurity Standard Compliance Table for Moxa Computers with Moxa Industrial Linux:

Moxa Computer Series	MIL Version	Cybersecurity Standard
UC-8200 Series	MIL3 Standard	• Europe RED (Directive 2014/53/EU) compliant
	MIL3 Secure	• Europe RED (Directive 2014/53/EU) compliant • IEC 62443-4-2 Security Leve 2 **certified**
UC-1222A Series UC-2222A Series UC-3400A Series UC-4400A Series	MIL3 Standard	• Europe RED (Directive 2014/53/EU) compliant
	MIL3 Secure	• Europe RED (Directive 2014/53/EU) compliant • IEC 62443-4-2 Security Leve 2 compliant
V1200 Series	MIL3 Secure	• Europe RED (Directive 2014/53/EU) compliant

To identify the security model that you have, use the **`mx-interface-mgmt deviceinfo`** command to display the information. Only secure models will have **`SECUREBOOT`** enabled.

```
moxa@moxa-tbzkb1090923: ~# mx-interface-mgmt deviceinfo
SERIALNUMBER=TBBBB1182827
MODELNAME=UC-8220-T-LX-US-S
SECUREBOOT=Enabled
```

The following table compares the main features in the standard and secure models.

	Standard Model	Secured Model
IEC 62443-4-2 SL2 Host Device Certified	N/A	✓
Security Configuration	Default Debian configuration	IEC 62443-4-2 SL-2 Certified
Secure Boot	N/A	✓
AppArmor and SELinux Support	✓	✓
Boot from SD or USB	✓	N/A (SD/USB is not secure as a boot source)
Disk Encryption	N/A	✓
Install Image via TFTP	✓	N/A (TFTP is not a secure protocol)
Secure Image Installation	N/A	✓
Secure Update	✓	✓
Intrusion Detection	✓ (AIDE preinstalled without pre-defined monitoring database)	✓ (AIDE with security monitoring database pre-defined)
Intrusion Prevention	✓ (Fail2ban)	✓ (Fail2ban)
Network Security Monitoring	✓ (Zeek)	✓ (Zeek)
Firewall	✓ (nftable disabled by default)	✓ (nftable with pre-configured security policy)
Security Diagnosis Tool (Moxa Guardian)	N/A	✓
Security Event Audit Log	✓ (Audit service disabled by default)	✓ (Audit service configured and running)
TPM 2.0	✓	✓
Backup, Decommission and Recovery	✓ (Moxa System Management)	✓ (Moxa System Management)
Network Management	✓ (Moxa Connection Management)	✓ (Moxa Connection Management)
Computer Interface Management	✓ (Moxa Computer Interface Manager)	✓ (Moxa Computer Interface Manager)

Eligible Computing Platforms

This user manual is applicable to Moxa's Arm-based computers listed below and covers the complete set of instructions applicable to all the supported models.

Some Moxa Arm-based computers come preinstalled with MIL1, while others come with MIL3 Standard. The table below lists the computer series that support MIL3, along with instructions on how to order the MIL3 Standard or Secure versions if the desired version is not preinstalled.

Moxa Computer Series	Preinstalled OS	How to Order MIL Versions
UC-8200 Series	MIL1 (Debian 9, kernel 4.4)	• Order MIL3 Standard or MIL3 Secure via CCS using the model "UC-8200 (CTO)"
UC-1222A Series	MIL3 Standard (Debian 11, kernel 5.10)	• MIL3 Standard preinstalled • Order MIL3 Secure via CCS using the model "UC-1200A (CTO)"
UC-2222A Series		• MIL3 Standard is preinstalled • Order MIL3 Secure via CCS using the model "UC-2200A (CTO)"
UC-3400A Series		• MIL3 Standard preinstalled • Order MIL3 Secure via CCS using the model "UC-3400A (CTO)"
UC-4400A Series		• MIL3 Standard preinstalled • Order MIL3 Secure via CCS using the model "UC-4400A (CTO)"
V1200 Series	MIL3 Secure (Debian 11, kernel 5.10)	• MIL3 Secure preinstalled • Only MIL3 Secure is available for this series

**Moxa's Computer Configuration System (CCS)*

2. Getting Started

Connecting to the Arm-based Computer

You will need another computer to connect to the Arm-based computer and log on to the command line interface. There are two ways to connect: locally through serial console or ethernet cable, or remotely via Secure Shell (SSH). Refer to the Hardware Manual to see how to set up the physical connections.

For default login username and password, please reference the Default Credentials and Password Strength.

The username and password are the same for all serial console and SSH remote log in actions. Root account login is disabled until you manually create a password for the account. The user **moxa** is in the **sudo** group so you can operate system level commands with this user using the **sudo** command. For additional details, see the Sudo Mechanism section in Chapter 7.

ATTENTION

For security reasons, we highly recommend that you disable the default user account and create your own user accounts.

Connecting through the Serial Console

This method is particularly useful when using the computer for the first time. The signal is transmitted over a direct serial connection, so you do not need to know either of its two IP addresses in order to connect to the Arm-based computer. To connect through the serial console, configure your PC's terminal software using the following settings.

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

Below we show how to use the terminal software to connect to the Arm-based computer in a Linux environment and in a Windows environment.

Linux Users

NOTE

These steps apply to the Linux PC you are using to connect to the Arm-based computer. Do NOT apply these steps to the Arm-based computer itself.

Take the following steps to connect to the Arm-based computer from your Linux PC.

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

   For Centos and Fedora:

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

   For Ubuntu and Debian:

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

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

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

3. Select **Serial port setup**.

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

4. Select **A** to change the serial device. Note that you need to know which device node is connected to the Arm-based computer.

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

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

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

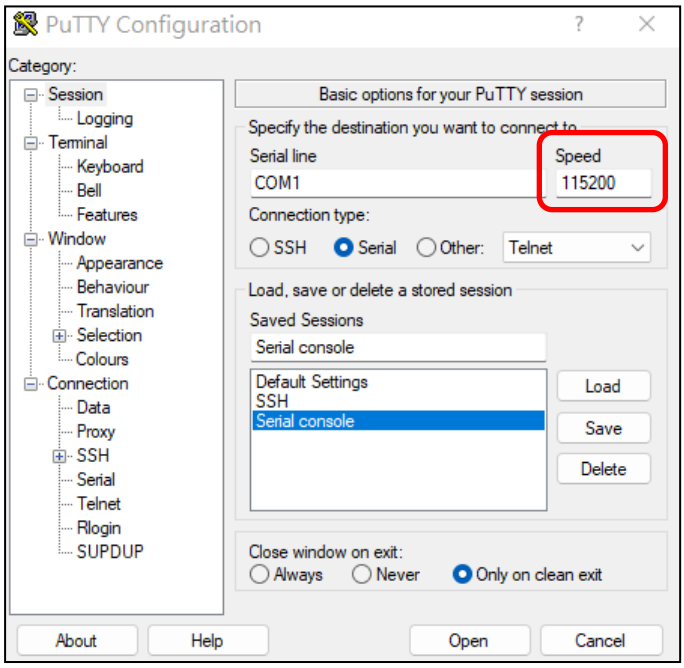
Windows Users

NOTE

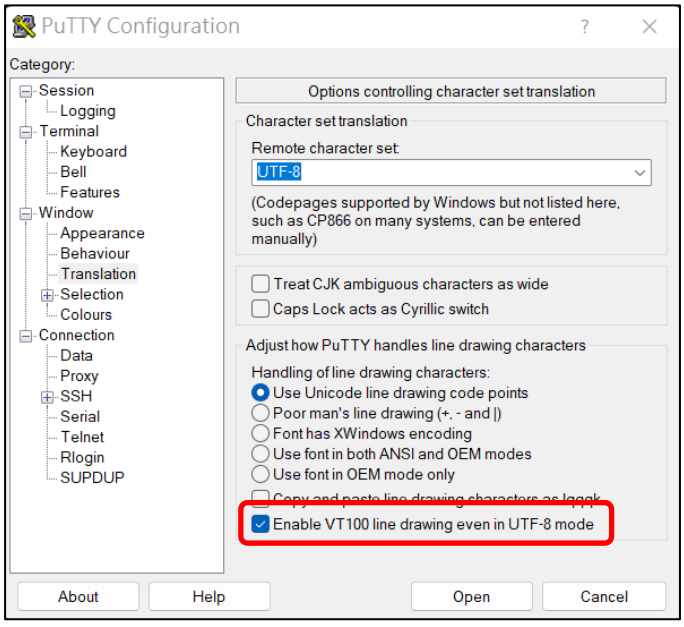
These steps apply to the Windows PC you are using to connect to the Arm-based computer. Do NOT apply these steps to the Arm-based computer itself.

Take the following steps to connect to the Arm-based computer from your Windows PC.

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



3. Select the **Serial** connection type and choose settings that are similar to the Minicom settings.
4. Enable **VT100 line drawing** option for the MCM GUI configurator to show correctly.



Connecting via the SSH

The Arm-based computer supports SSH connections remotely or over an Ethernet network. If you are connecting the computer using an Ethernet cable, refer to the following IP addresses information

Ethernet Port	Configuration	IP Address
LAN 1 (*)	DHCP (DHCP client)	Assigned by DHCP server. Link-local IP addresses will be assigned when DHCP server is not available
LAN 2	Static IP	192.168.4.127

*LAN 1 is by default for DHCP/link-local IP configuration and is managed by Moxa Connection Manger (MCM).

NOTE

Be sure to configure the IP address of your notebook/PC's Ethernet interface on the same subnet as the LAN port of Arm-based computer you plan to connect to. For example, 192.168.4.**126** for LAN2.

Linux Users

NOTE

These steps apply to the Linux PC you are using to connect to the Arm-based computer. Do NOT apply these steps to the Arm-based computer itself.

Use the **ssh** command from a Linux computer to access the computer's LAN2 port.

```
user@PC1:~ ssh moxa@192.168.4.127
```

Type **yes** to complete the connection.

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

To connect using LAN1, you need to use the IP offered by DHCP server from LAN1.

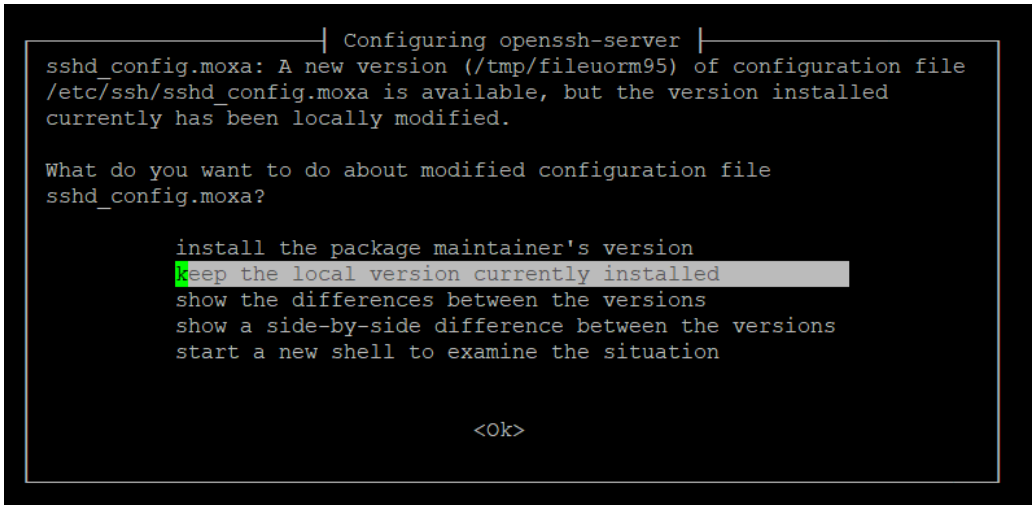
ATTENTION

Regenerate SSH key regularly

In order to secure your system, we suggest doing a regular SSH-rekey, as shown in the following steps:

```
moxa@moxa-tbzkb1090923:~$ cd /etc/ssh
moxa@moxa-tbzkb1090923:~$ sudo rm /etc/ssh/ssh_host_*
moxa@moxa-tbzkb1090923:~$ sudo dpkg-reconfigure openssh-server
moxa@moxa-tbzkb1090923:~$ sudo systemctl restart ssh
```

Select "**keep the local version currently installed**" following is prompt during rekey process



For more information about SSH, refer to the following link.

https://wiki.debian.org/SSH

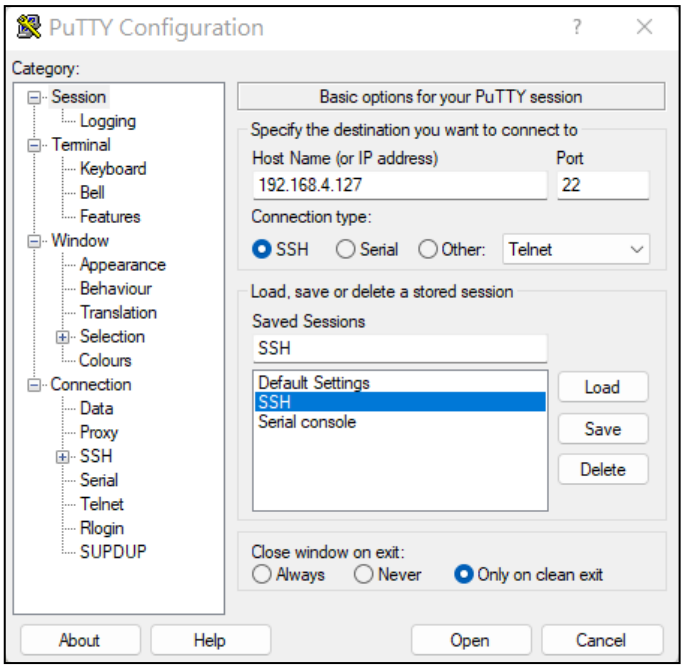
Windows Users

NOTE

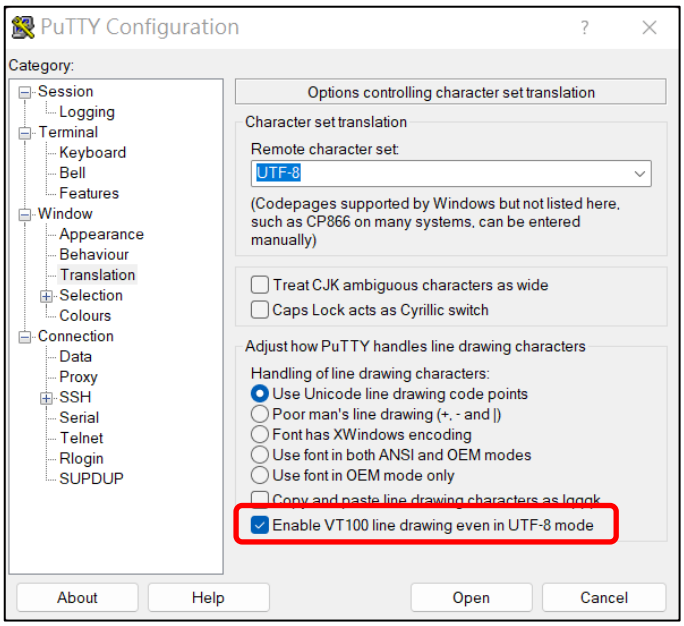
These steps apply to the Windows PC you are using to connect to the Arm-based computer. Do NOT apply these steps to the Arm-based computer itself.

Take the following steps from your Windows PC.

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



Enable **VT100 line drawing** option for the MCM GUI configurator to show correctly



Managing User Accounts

Default User Account and Password Policy

The default login username and password of Moxa Industrial Linux are both **moxa** for the first-time login. You will be prompted to set a new password before you can continue to login.

- Default Username: **moxa**
- Default Password: **moxa**

Password Strength Requirements:

- At least 8 characters in length
- Dictionary checking is enabled to prevent the use of common passwords

To modify the password strength policy, edit the **/etc/security/pwquality.conf.d/00-moxa-standard-pwquality.conf** file to configure the policy.

NOTE

Click the following link for more information on the password strength configuration.

https://manpages.debian.org/bullseye/libpwquality-common/pwquality.conf.5.en.html

For bootloader administrator password configuration, refers to the bootloader configuration section.

Creating and Deleting User Accounts

ATTENTION

DO NOT disable the default account before creating an alternative user account.

You can use the **useradd** and **userdel** commands to create and delete user accounts. Be sure to reference the manual pages (man) page of these commands to set relevant access privileges for the account. Following example shows how to create a **test1** user in the **sudo** group whose default login shell is **bash** and has home directory at **/home/test1**:

```
moxa@ moxa-tbzkb1090923:~# sudo useradd -m -G sudo -s /bin/bash test1
```

To change the password for test1, use the **passwd** option along with the new password. Retype the password to confirm the change.

```
moxa@moxa-tbzkb1090923:~# sudo passwd test1
New password:
Retype new password:
passwd: password updated successfully
```

To delete the user **test1**, use the **userdel** command.

```
moxa@ moxa-tbzkb1090923:# sudo userdel test1
```

Modifying User Accounts

You can use the **usermod** commands to create and modify the user account settings. Some examples of commonly used settings are listed here, including adding a user to a group, locking an account, activating an account and setting the password expiration date for the account.

1. Adding user test1 to the user group Moxa

```
moxa@ moxa-tbzkb1090923:# sudo usermod -a -G Moxa test1
```

2. Disabling or locking the user account test1

```
moxa@ moxa-tbzkb1090923:# sudo usermod -L test1
```

3. Activating the user account test1

```
moxa@ moxa-tbzkb1090923:# sudo usermod -U test1
```

4. Set a password expire date of 2023-11-01 for the user account test1.

```
moxa@ moxa-tbzkb1090923:# sudo usermod -e 2023-11-01 test1
```

NOTE

Refers to below link for complete usage of **usermod**

https://linux.die.net/man/8/usermod

Changing the Password

You can use the **passwd** commands to change the password of a user account. Changing the password will not have any impact on other functionalities.

An example of changing the password for user account **test1**.

```
moxa@ moxa-tbzkb1090923:# sudo passwd test1
New password:
Retype new password:
passwd: password updated successfully
```

Querying the MIL and OS Image Version

Use the **mx-ver** command to check the OS **image version** on your Arm-based computer.

1. Use the **mx-ver -M** command to check the **MIL version**.

```
moxa@moxa-imoxa1000042:~$ mx-ver -M
3.4.1
```

2. Use the **mx-ver** command to check the **OS image version**.

```
moxa@moxa-tbzkb1090923:# mx-ver
UC-8220-T-LX-US-S MIL3 version 1.0 Build 22052300
```

3. Use the **mx-ver -h** command to display additional options for querying product-related information.

```
moxa@moxa-tbzkb1090923:# mx-ver -h

Usage: mx-ver [OPTION]
       -a: show product information inline
       -b: show the build time
       -m: show the model name
       -s: show the product series
       -v: show the image version
       -A: show all information
       -M: show the MIL version
       -o: show the image option code
       -h: show the help menu
```

Querying the Device Information

Use the **# mx-interface-mgmt deviceinfo** command to retrieve general information for your Moxa Arm-based Computer

Command and Usage	Description
deviceinfo	Shows the following device information: • Serial number (S/N) • Model name • SECUREBOOT (Enabled/Disabled)

```
moxa@moxa-tbbbb1182827:~$ mx-interface-mgmt deviceinfo
SERIALNUMBER=TBBBB1182827
MODELNAME=UC-8220-T-LX-US-S
SECUREBOOT=Enabled
```

Determining Available Drive Space

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

```
moxa@moxa-tbzkb1090923:~$ sudo df -h

Filesystem      Size  Used  Avail   Use%  Mounted on
devtmpfs        485M     0   485M     0%  /dev
tmpfs           497M  7.1M   490M     2%  /run
/dev/mmcblk0p2  984M  150M   780M    17%  /boot_device/p2
/dev/mmcblk0p3  5.9G   39M   5.5G     1%  /boot_device/p3
/dev/mmcblk0p4  240M  2.8M   221M     2%  /var/log
/dev/loop0      147M  147M      0   100%  /boot_device/p2/lower
overlay         5.9G   39M   5.5G     1%  /
/dev/mmcblk0p1   54M   15M    36M    30%  /boot_device/p1
tmpfs           497M     0   497M     0%  /dev/shm
tmpfs           5.0M     0   5.0M     0%  /run/lock
tmpfs           497M     0   497M     0%  /sys/fs/cgroup
tmpfs           100M     0   100M     0%  /run/user/1000
```

Shutting Down the Device

To shut down the computer, first disconnect the power source. When the computer is powered off, main components such as the CPU, RAM, and storage devices are powered off, although an internal clock may retain battery power.

You can use the Linux command **shutdown** to close all software running on the device and halt the system. However, main components such as the CPU, RAM, and storage devices will continue to be powered after you run this command.

```
moxa@moxa-tbzkb1090923: ~# sudo shutdown -h now
```

3. Device Configuration

In this chapter, we describe how to configure the basic settings of Moxa Arm-based computers, including using the bootloader menu, configuring the network connections and power-saving settings, and localizing the computer. The instructions in this chapter cover all functions supported in Moxa Arm-based computers. Before referring to the sections in this chapter, ensure that they are applicable to and are supported by the hardware specification of your Arm-based computer.

Bootloader Configuration

Accessing the Bootloader Configuration Menu

To access bootloader menu, you must first connect to Moxa Arm-based computer via its serial console port. After powering on the Arm-based computer, press **Ctrl + Backspace** or **DEL** to enter the bootloader configuration menu

NOTE

If you cannot enter the bootloader menu by pressing <Ctrl + Backspace> or <DEL>, replace the PuTTy tool with the Tera Term terminal console tool (detailed information is available at: https://ttssh2.osdn.jp/index.html.en.)

```
--------------------------------------------------------------------------------
 Model: UC-8220-T-LX-US-S
 Boot Loader Version: 3.0.0S04
 Build date: May 13 2022 - 14:23:10    Serial Number: TBBBB1182827
 LAN1 MAC: 00:90:E8:A6:37:E1           LAN2 MAC: 00:90:E8:A6:37:E2
--------------------------------------------------------------------------------
 (0) Boot Management                   (1) Install System Image
 (2) Admin Password                    (3) Advance Setting
 (4) Exit and Reboot                   (5) Go To Linux
--------------------------------------------------------------------------------
```

Production and Developer Mode

The configurable options and operations in bootloader menu of Standard and Secure model are different for security consideration. Below is an overview of configuration options provided in Bootloader.

The Secure Model's bootloader menu has two modes (**Production** and **Developer mode**) where the **Production Mode** is the default mode with security configuration configured to comply with IEC 62443-4-2 security level 2 standard. **Developer Mode** provides addition operation and configuration that should only be used during development stage or maintenance.

For **Secure Model**, the administrator password to access bootloader menu is set by default. The Default Administrator Password is the **unique Serial Number(S/N)** printed on the sticker of Moxa Arm-based computer.

1. To switch to Developer Mode, run **`mx-bootloader-mgmt mode developer`**

```
root@moxa-tbbbb1182827:/# mx-bootloader-mgmt mode developer
Set device into developer mode Done
Mode info: prod_mode=1
root@moxa-tbbbb1182827:/# reboot
```

2. To switch to Production Mode, use **mx-bootloader-mgmt mode production**

```
root@moxa-tbbbb1182827:/# mx-bootloader-mgmt mode production
Set device into production mode Done
Mode info: prod_mode=0
root@moxa-tbbbb1182827:/# reboot
```

3. Reboot computer for setting to take effect
4. To check the current mode, run **mx-bootloader-mgmt mode info**

An overview of bootloader configuration options is listed in the following table:

Main Menu	Sub Menu	Secure Model		Standard Model
Main Menu	Sub Menu	Production Mode	Developer Mode	Production Mode
(0) Boot Management	(0) Set to Default	N/A	N/A	✓
	(1) Boot Option	N/A	N/A	✓
	(2) Advance Boot Option	N/A	N/A	✓
	(3) View Current Setting	N/A	N/A	✓
(1) Install System Image	(0) Install System Image from TFTP	N/A	✓	✓
	(1) Install System Image from SD	✓	✓	✓
	(2) Install System Image from USB	✓	✓	✓
	(3) TFTP Settings	N/A	✓	✓
(2) Admin Password	(0) Set to Default	✓	✓	✓
	(1) Enable/Disable Admin Password	✓ (enabled by default)	✓ (enabled by default)	✓ (disabled by default)
	(2) Configure Admin Password	✓	✓	✓
	(3) Configure Admin Password Policy	✓	✓	✓
(3) Advance Setting	(0) Set to Default	✓	✓	✓
	(1) Configure Auto Reboot	✓ (enabled by default)	✓ (enabled by default)	✓ (disabled by default)
	(2) Configure Login Message	✓	✓	✓
	(3) Configure Invalid Login Attempts	✓	✓	✓
	(4) Clear TPM	N/A	✓	✓
	(5) Configure Linux Security Modules	✓	✓	✓
	(6) Enable/Disable Interfaces	✓	✓	✓
	(7) View Bootloader log	✓	✓	✓
(3) Exit & Reboot	–	✓	✓	✓
(4) Go to Linux	–	N/A	N/A	✓

Boot Management

Boot Option

By default, Moxa Arm-based computers boot up from the embedded eMMC flash. Some models also provide an option to boot up from an external SD or USB.

The following is an example of changing first boot priority to SD card and setting the secondary boot option to SD card if the first option fails to boot.

1. Select **(0) Boot Management > (1) Boot Option**
2. Choose to first boot from an external storage.
3. Choose if the embedded storage should be disabled.

   If the embedded storage is disabled, Moxa Arm-based computers will only attempt to boot from the SD card. If embedded storage is set to eMMC, the computers will try to boot from SD; if that fails, they will boot from eMMC.

4. Set the External Storage to the SD card

```
--------------------------------------------------------------------------------
 Model: UC-8220-T-LX-US-S
 Boot Loader Version: 3.0.0S04
 Build date: May 13 2022 - 14:23:10    Serial Number: TBBBB1182827
 LAN1 MAC: 00:90:E8:A6:37:E1           LAN2 MAC: 00:90:E8:A6:37:E2
--------------------------------------------------------------------------------
 (0) Boot Management                   (1) Install System Image
 (2) Admin Password                    (3) Advance Setting
 (4) Exit and Reboot                   (5) Go To Linux
--------------------------------------------------------------------------------
Command>>1
Boot Management : Default
Boot Order : Embedded First
Embedded Storage : eMMC
External Storage : Disabled

Would you like to configure the Boot Option?
0 - No, 1 - Yes (0-1, Enter to abort): 1
Set Boot Order:
 0 - Embedded First, 1 - External First (0-1, Enter to abort): 1
Set Embedded Storage:
 0 - Disabled, 1 - eMMC (0-1, Enter to abort): 1
Set External Storage:
 0 - Disabled ,1 - SD (0-1, Enter to abort): 1
```

The table below lists all possible combinations of boot options configuration and the corresponding boot action

Set Boot Order	Set Embedded Storage	Set External Storage	Boot Action
0 – Embedded First	1 – eMMC	0 – Disabled	Boot from eMMC
1 – External First	0 – Disabled	1 – SD or 2 – USB	Boot from the external storage
0 – Embedded First	1 – eMMC	1 – SD or 2 – USB	First boot from eMMC; if it fails, boot from the external storage
1 – External First	1 – eMMC	1 – SD or 2 – USB	Boot from the external storage; if this fails, boot from eMMC

Advance Boot Option

Allow advanced users to edit the **bootargs** and **bootcmd** parameters to customize the boot process.

- **bootargs:** Used to tell the kernel how to configure various device drivers and where to find the root filesystem.
- **bootcmd:** Bootloader will execute the commands listed sequentially. Commands should be separated by semicolons.

Installing the System Image

Installing System Image From TFTP

1. Prepare a TFTP server
2. Set up a TFTP server.
3. Make sure the image (*.img) file is in your TFTP server directory.

> **IMPORTANT!**
>
> Use this method to install a system image on your computer if the size of the image file is less than 2 GB. If the file size is larger than 2 GB, use the SD card or USB to install the system image.

4. Select **Install System Image > TFTP Settings** and configure the following:
   - The LAN port to be used for TFTP transfer
   - Local IP address of LAN port
   - TFTP server IP
5. Press **ESC** to exit and select **Install System Image from TFTP**.

   If you want to change the TFTP IP address, enter 1 to set up the local LAN port IP address and the TFTP server IP address, and then choose an image (*.img) file.

```
Current IP Address

Local IP Address : 192.168.1.2
Server IP Address : 192.168.2.3
Using LAN2 to download data.
Do you want to change the ip address?
0 - No, 1 - Yes(0-1, Enter to abort):1
Local IP Address : 192.168.31.134
Server IP Address : 192.168.31.132
Saving Environment to SPI Flash...
Erasing SPI flash...Writing to SPI flash...done
Valid environment: 2
System Image File Name (system image.img): IMG_UC-8200_MIL3_V1.0.img
```

6. After the system image installation process is complete, unplug the power supply and reboot the system.
7. After rebooting the system, you can use the following command to check if the system image is up-to-date.

```
moxa@moxa-tbzkb1090923:# sudo mx-ver
UC-8220-T-LX-US-S MIL3 version 1.0 Build 22052300
```

Installing the System Image From SD or USB

The system image on the Moxa Arm-based computers can be installed through an external SD or USB disk. Prepare a USB or SD disk in the FAT32 or ext4 format with the system image and plug it into the USB or SD port of the computer.

1. Select **Install System Image > Install System Image from SD** or **Install System Image from USB**
2. Type in the system image file name.

```
--------------------------------------------------------------------------------
Model: UC-8220-T-LX-US-S
Boot Loader Version: 3.0.0S04
Build date: May 13 2022 - 14:23:10   Serial Number: TBBBB1182827
LAN1 MAC: 00:90:E8:A6:37:E1           LAN2 MAC: 00:90:E8:A6:37:E2
--------------------------------------------------------------------------------
 (0) Install System Image from TFTP    (1) Install System Image from SD
 (2) Install System Image from USB     (3) TFTP Settings
--------------------------------------------------------------------------------
Command>>2

System Image File Name (system image.img): IMG_UC-8200_MIL3_V1.0.img
```

NOTE

Make sure to put **the hash file of the system image** in the same folder as image as integrity validation is required

3. After the system image installation process is complete, unplug the power supply and reboot the system.
4. After rebooting the system, you can use the following command to check if the system image is up-to-date.

```
moxa@moxa-tbzkb1090923:# sudo mx-ver
UC-8220-T-LX-US-S MIL3 version 1.0 Build 22052300
```

Administrator Password

Enabling/Disabling Admin Password

For the **Secure Model**, the administrator password to access the bootloader menu is set by default. The

Default Administrator Password is the **unique Serial Number(S/N)** printed on the sticker of Moxa Arm-based computer.

For **Standard Model**, the bootloader menu is not password-protected by default. o enhance the security of your Moxa ARM-based computer, it is strongly recommended to set up an administrator password if physical unauthorized access is a possibility. To setup an administrator password, follow the below procedures:

1. Select **Admin Password > Enable/Disable Admin Password**.
2. Select **1** to set up an administrator password. The currently set password will be cleared if 0 (disable) is selected.

3. Enter the password you would like to set twice; the password strength requirement is at least 8 characters in length.

```
--------------------------------------------------------------------------------
Model: UC-8220-T-LX-US-S
Boot Loader Version: 3.0.0S04
Build date: May 13 2022 - 14:23:10   Serial Number: TBBBB1182827
LAN1 MAC: 00:90:E8:A6:37:E1           LAN2 MAC: 00:90:E8:A6:37:E2
--------------------------------------------------------------------------------
 (0) Set to Default                      (1)Enable/Disable Admin Password
 (2) Configure Admin Password            (3)Configure Admin Password Policy
--------------------------------------------------------------------------------
Command>>2
Current Mode: Disabled

0 - Disable, 1 - Enable (0-1, Enter to abort): 1

The current password is empty, please set one.

Enter the Administrator password
Enter current password: ********

Admin Password Policy:
- Minimum length: 8

Enter new password: ********
Retype password: ********
Password set successfully
Password status : Enabled.
```

4. Once Administrator password is set, password authentication is required when accessing bootloader menu.

```
DRAM:  1 GiB
MMC:   OMAP SD/MMC: 0, OMAP SD/MMC: 1, OMAP SD/MMC: 2
Net:   cpsw0, cpsw1
Non-security model.
Model: 0x02
2.0 TPM (device-id 0x15D1, rev-id 16)
TPM2 Init OK!
TPM2 Startup (1) OK!

Press <DEL> To Enter BIOS configuration Setting

Enter the Administrator password
Enter current password: ********
```

WARNING

It is important to save the password in a secure location. If the password is lost and access to bootloader menu is needed, you will have to contact Moxa technical support to send your Arm-based computer to Moxa for password reset.

Configuring the Admin Password Policy

To change the administrator password policy, select **Admin Password > Configure Admin Password Policy** and follows the on-screen instructions. Changing the password will not have any impact on functionalities.

```
--------------------------------------------------------------------------------
Model: UC-8220-T-LX-US-S
Boot Loader Version: 3.0.0S04
Build date: May 13 2022 - 14:23:10   Serial Number: TBBBB1182827
LAN1 MAC: 00:90:E8:A6:37:E1           LAN2 MAC: 00:90:E8:A6:37:E2
--------------------------------------------------------------------------------
 (0) Set to Default                      (1)Enable/Disable Admin Password
 (2) Configure Admin Password            (3)Configure Admin Password Policy
--------------------------------------------------------------------------------
Command>>3
Current setting:
Admin Password Policy:
 - Minimum length: 8
********************************************************************************
*
 Do you want to configure admin password policy setting?
********************************************************************************
*
0 - No, 1 - Yes (0-1, Enter to abort): 1
- Minimum length (6-16, Enter to abort): 6
- Minimum numeric numbers (0-16, Enter to abort): 1
- Minimum lowercase or uppercase letters combined (0-16, Enter to abort): 1
```

Minimum Length

Setting	Description	Factory Default
Input from 6 to 16	It allows users to decide the minimum length of the password.	8

Minimum Numeric Numbers

Setting	Description	Factory Default
Input from 0 to 16	It allows users to decide the minimum of numeric number that the password must contain	0

Minimum Lowercase or Uppercase Letters Combined

Setting	Description	Factory Default
Input from 0 to 16	It allows users to decide the minimum letters (lowercase or uppercase combined) that the password must contain.	0

Configuring Admin Password

To change the administrator password, select **Admin Password > Configure Admin Password** and follows the on-screen instructions

Resetting the Admin Password to Default

If you lost your password, follow the below steps to reset the password to the factory default

1. After powering on the Arm-based computer, press **Ctrl + Backspace** or **DEL** to enter the Bootloader configuration menu that prompts for a password.

```
DRAM:  1 GiB
MMC:   OMAP SD/MMC: 0, OMAP SD/MMC: 1, OMAP SD/MMC: 2
Net:   cpsw0, cpsw1
Non-security model.
Model: 0x02
2.0 TPM (device-id 0x15D1, rev-id 16)
TPM2 Init OK!
TPM2 Startup (1) OK!

Press <DEL> To Enter BIOS configuration Setting

Enter the Administrator password
Enter current password:
```

2. Immediately press and hold the **FN** button on the Moxa Arm-based computer for over 5 seconds will trigger the password reset process. You must complete this step within **10 seconds** after step one for the reset process to initiate.

Login Policy

Invalid Login Attempts

This determines the **maximum consecutive failure login attempts** allowed during the specified **time period** and the duration to block users from accessing bootloader configuration menu when failure login attempts and time period is over the defined threshold.

To configure this policy, select **Advance Setting > Configure Invalid Login Attempts** and follow the on-screen instructions.

```
--------------------------------------------------------------------------------
 Model: UC-8220-T-LX-US-S
 Boot Loader Version: 3.0.0S04
 Build date: May 13 2022 - 14:23:10    Serial Number: TBBBB1182827
 LAN1 MAC: 00:90:E8:A6:37:E1           LAN2 MAC: 00:90:E8:A6:37:E2
--------------------------------------------------------------------------------
 (0) Set to Default                     (1) Configure Auto Reboot
 (2) Configure Login Message            (3) Configure Invalid Login Attempts
 (4) View Bootloader log
--------------------------------------------------------------------------------
Command>>3
Current setting: [5] consecutive invalid login within [60] seconds will reboot
and disable access to bootloader menu for [300] seconds.
********************************************************************************
*
 Do you want to configure the invalid login attempts setting?
********************************************************************************
*
0 - No, 1 - Yes (0-1, Enter to abort): 1

Input 0 to any of the configuration below will disable invalid login check

Consecutive invalid login attempts (0-5, Enter to abort):
Within how many seconds (0-60, Enter to abort):
Disable access for how many seconds (0-900, Enter to abort):
```

Consecutive Invalid Login Attempts

Configuration	Setting	Factory Default
Consecutive invalid login attempts	Input from 0 to 5	0 (Standard model) 5 (Secure model)
Within how many Seconds	Input from 0 to 60	0 (Standard model) 60 (Secure model)
Disable access for how many seconds	Input from 0 to 900	0 (Standard model) 300 (Secure model)

NOTE

Input 0 to any of the above configuration will disable the invalid login check.

Auto Reboot After Inactivity

This determines the time period for auto reboot when users do not do any action.

To set the time period, select **(2) Advance Setting > (1) Configure Auto Reboot** and follow the on-screen instructions.

Setting	Description	Factory Default
Input from 0 to 900 (seconds)	This determines the time period for auto reboot when users do not do any action	0 (Standard model) 900 (Secure model)

Login Banner Message

This allows users to customize the login message before prompting the administrator password.

To configure the message, select **Advance Setting > Configure Login Message** and follow the on-screen instructions.

```
U-Boot 2020.04-ga174fe3ef0-dirty (May 13 2022 - 14:23:01 +0800)
DRAM:  2 GiB
PMIC: PFUZE3000 DEV_ID=0x31 REV_ID=0x11
MMC:   FSL_SDHC: 0, FSL_SDHC: 2
Loading Environment from SPI Flash... SF: Detected mx25l12805d with page size
256 Bytes, erase size 64 KiB, total 16 MiB
OK
In:    serial
Out:   serial
Err:   serial
SEC0: RNG instantiated
Net:   eth0: ethernet@30be0000 [PRIME]Get shared mii bus on ethernet@30bf0000
FEC0:1 is connected to ethernet@30be0000.  Reconnecting to ethernet@30bf0000
, eth1: ethernet@30bf0000
Model: 0x00
Normal Boot

Press <DEL> To Enter BIOS configuration Setting

Enter the Administrator password
Enter current password:
```

Enable AppArmor and SELinux

The bootloader menu includes an option to enable support for AppArmor and SELinux, which are both disabled by default. Selecting this option will add the corresponding boot parameters to the kernel during startup.

To enable AppArmor or SELinux, navigate to **Advanced Settings > Configure Linux Security Modules** and follow the on-screen instructions.

> **NOTE**
>
> Enabling these options in the bootloader only passes the parameters to the kernel.
>
> The user-space tools for AppArmor and SELinux are not pre-installed on the system. If full functionality is required, you will need to install the respective user-space tools and configure the appropriate security policies.

Clearing the TPM Module

Clearing the TPM will erases information stored on the TPM. You will lose all created keys and access to data encrypted by these keys.

To clear the TPM, select **Advance Setting > Clear TPM** and follow the directions.

Changing the Default Hostname

The default hostname of UC computer with Moxa Industrial Linux 3 is unique for each computer. The hostname is in a format of moxa-[serial number].

If you would like to change the default hostname, follow the below procedure:

1. Modify the hostname by editing /etc/hostname
2. Disable the moxa-hostname service with '**systemctl disable moxa-hostname**' command. moxa-hostname is a service designed to execute automatically during system startup, setting the hostname to a default unique value.
3. Reboot the computer.

Localizing Your Arm-based Computer

Adjusting the Time

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

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

MM = Month
DD = Date
hhmm = hour and minute

```
moxa@moxa-tbzkb1090923:# sudo date 102900282021
Fri 29 Oct 2021 12:28:00 AM GMT
```

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

```
moxa@moxa-tbzkb1090923:# sudo hwclock -w
moxa@moxa-tbzkb1090923:# sudo hwclock
2021-10-28 16:25:04.077432+00:00
```

NOTE

Click the following links for more information on date and time:

https://www.debian.org/doc/manuals/system-administrator/ch-sysadmin-time.html

https://wiki.debian.org/DateTime

NTP Time Synchronization

The Moxa Industrial Linux (MIL) uses Network Time Security (NTS) to secure NTP, which provides a handshake (TLS) before using a NTP server and authentication of the NTP time synchronization packets using the results of the TLS handshake.

The default NTP client in MIL is **`Chrony`**. MIL disabled NTP server without NTS support by default and uses the following public NTP servers that support NTS.

- Cloudflare
- Netnod
- System76
- PTB

The default server list is configured in the **/etc/chrony/sources.d/moxa-nts.sources** file.

```
# prefer nts over ntp server
server time.cloudflare.com nts iburst prefer
server sth1.nts.netnod.se nts iburst prefer
server sth2.nts.netnod.se nts iburst prefer
server virginia.time.system76.com nts iburst prefer
server ohio.time.system76.com nts iburst prefer
server oregon.time.system76.com nts iburst prefer
server ptbtime1.ptb.de nts iburst prefer
server ptbtime2.ptb.de nts iburst prefer
server ptbtime3.ptb.de nts iburst prefer
```

The configuration file for Chrony is at **/etc/chrony/chrony.conf**.

The following example show some basic functions to monitor the current status of the Chrony's chronyc tool and make changes if necessary.

1. Check the time synchronization status between the local system and reference server using the command:

   `# chronyc tracking`

   ```
   moxa@moxa-tbbbb1182827:~$ chronyc tracking
   Reference ID    : A29FC801 (time.cloudflare.com)
   Stratum         : 4
   Ref time (UTC)  : Sun Jul 31 18:27:42 2022
   System time     : 0.000334575 seconds slow of NTP time
   Last offset     : +0.000226902 seconds
   RMS offset      : 0.005672113 seconds
   Frequency       : 27.766 ppm fast
   Residual freq   : -0.065 ppm
   Skew            : 3.403 ppm
   Root delay      : 0.203054637 seconds
   Root dispersion : 0.006750254 seconds
   Update interval : 517.4 seconds
   Leap status     : Normal
   ```

2. Check the time source configured in the **/etc/chrony/chrony.conf** file using the **# chronyc sources** command.

```
moxa@moxa-tbbbb1182827:~$ chronyc sources

MS Name/IP address         Stratum Poll Reach LastRx Last sample
===============================================================================
^+ ohio.time.system76.com      2   9   377   147    +18ms[  +18ms] +/-  141ms
^+ oregon.time.system76.com    2   9   377   203    +14ms[  +14ms] +/-  137ms
^- ptbtime1.ptb.de             1   9    21   682  -2780us[-2417us] +/-  166ms
^- ptbtime2.ptb.de             1   9    21   674  -5243us[-4882us] +/-  169ms
^- ptbtime3.ptb.de             1   9    21   687    +17ms[  +17ms] +/-  192ms
^+ sth1-ts.nts.netnod.se       1   9   377   220    -12ms[  -12ms] +/-  162ms
^- sth2-ts.nts.netnod.se       1   8   377    91  -3843us[-3843us] +/-  171ms
^* time.cloudflare.com         3   9   377   230    +13ms[  +13ms] +/-  129ms
^+ virginia.time.system76.c>   2   9   377   226  -8753us[-8753us] +/-  116ms
```

3. Manually synchronize the time using the **# chronyc makestep** command.

NOTE

For additional details on Chrony, check the following links:
https://linux.die.net/man/8/chronyd
https://linux.die.net/man/1/chronyc

Setting the Time Zone

There are two ways to configure the Moxa Arm-based computer's time zone. One is using the **TZ** variable. The other is using the **/etc/localtime** file.

Using the TZ Variable

The format of the TZ environment variable looks like this:

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

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

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

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

In the second case, the reference time is Eastern Standard Time and the only conversion performed is for Daylight Saving Time. Therefore, there is no need to adjust the hardware clock for Daylight Saving Time twice per year.

In the third case, the reference time is always the time reported. You can use this option if the hardware clock on your machine automatically adjusts for Daylight Saving Time or you would like to manually adjust the hardware time twice a year.

```
moxa@Moxa-tbzkb1090923:~$ TZ=EST5EDT
moxa@Moxa-tbzkb1090923:~$ export TZ
```

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

The following table lists other possible values for the TZ environment variable:

Hours From Greenwich Mean Time (GMT)	Value	Description
0	GMT	Greenwich Mean Time
+1	ECT	European Central Time
+2	EET	European Eastern Time
+2	ART	
+3	EAT	Saudi Arabia
+3.5	MET	Iran
+4	NET	
+5	PLT	West Asia
+5.5	IST	India
+6	BST	Central Asia
+7	VST	Bangkok
+8	CTT	China
+9	JST	Japan
+9.5	ACT	Central Australia
+10	AET	Eastern Australia
+11	SST	Central Pacific
+12	NST	New Zealand
-11	MIT	Samoa
-10	HST	Hawaii
-9	AST	Alaska
-8	PST	Pacific Standard Time
-7	PNT	Arizona
-7	MST	Mountain Standard Time
-6	CST	Central Standard Time
-5	EST	Eastern Standard Time
-5	IET	Indiana East
-4	PRT	Atlantic Standard Time
-3.5	CNT	Newfoundland
-3	AGT	Eastern South America
-3	BET	Eastern South America
-1	CAT	Azores

Using the localtime File

The local time zone is stored in the **/etc/localtime** and is used by GNU Library for C (glibc) if no value has been set for the TZ environment variable. This file is either a copy of the **/usr/share/zoneinfo/** file or a symbolic link to it. The Arm-based computer does not provide **/usr/share/zoneinfo/** files. You should find a suitable time zone information file and write over the original local time file in the Arm-based computer.

Configuring Device Discovery

Moxa provides device discovery through an mDNS service. This discovery service is designed to locate devices within a trusted local network environment during the provisioning stage. The service is enabled by default. You can use the following commands to stop or disable the service.

```
root@moxa-imoxa1000038:/# systemctl stop moxa-mdns.service
root@moxa-imoxa1000038:/# systemctl disable moxa-mdns.service
Removed /etc/systemd/system/multi-user.target.wants/moxa-mdns.service.
root@moxa-imoxa1000038:/#
```

Configuring Power Saving

Setting the Power Modes

The Moxa Power Management tool allows setting the following three power modes enabling savings on power consumption of devices. The power savings could be limited by various field sites.

- Active: Sets the scaling governor to performance, allowing the system to run at full speed.
- Conservation: Sets the scaling governor to powersave, reducing the CPU frequency to save power.
- Standby: Freezes all processes and puts the CPU into a waiting state until an asynchronous interrupt event such as RTC, Wake-on-LAN occurs.

NOTE

1. The Moxa Power Management tool is currently available only for the **UC-3400A Series**.
2. All additional user-developed drivers must support the Power Manager when the system is set to Standby mode.

To see the list of commands for the power management tool, open the console and run the **sudo mx-pwr-mgmt** command.

```
moxa@moxa-tbdjb1028636:~$ sudo mx-pwr-mgmt -h
MOXA Power Management Command-line Utility

Usage:
  mx-pwr-mgmt [command]

Available Commands:
  info         Show current power mode.
  set          Affect system performance and power usage.
  wakeupctl    Control the wakeup sources.
Flags:
 -n, --dry-run Does everything, but suspend.
 -h, --help    Display help for mx-pwr-mgmt.
 -v, --version Display version for mx-pwr-mgmt.
moxa@moxa-tbdjb1028636:~$
```

Run the **mx-pwr-mgmt info** command to see the current configurations.

```
moxa@moxa-tbdjb1028636:~$ sudo mx-pwr-mgmt info
Current Power Mode:
  Active
Current CPU Frequency:
  1400000
Current Scaling Governor:
  performance

Supported Power Mode:
  Active Conservation Standby

Peripherals are turned off in Conservation mode:
  WiFi1 Cellular1 LAN1 LAN2 USB SD LEDs
Peripherals are turned off in Standby mode:
  WiFi1 Cellular1 USB SD LEDs
LEDs are Turned off in Conservation or Standby modes:
  RDY_Green

moxa@moxa-tbdjb1028636:~$
```

```
moxa@moxa-tbdjb1028636:~$ sudo mx-pwr-mgmt set -h

Usage:
  mx-pwr-mgmt set

Available Commands:
  Affect system power-saving policies.
    $ mx-pwr-mgmt set --mode <active|conservation|standby>
  Set power to conservation mode and wake up from Timer.
    $ mx-pwr-mgmt set --mode conservation --second <second>
  Set power to standby mode and wake up from RTC.
    $ mx-pwr-mgmt set --mode standby --second <second>
moxa@moxa-tbdjb1028636:~$
```

Command and Usage	Description
info	Display all Moxa Power Manager information. • Current Power Mode. • Current CPU Frequency. • Current Scaling Governor. • Supported Power Mode: Active, Conservation, Standby. • Peripherals are turned off in Conservation mode. • Peripherals are turned off in Standby mode. • LESs are turned off in Conservation or Stadnby modes.
set --mode <active|conservation|standby>	• active: set the CPU into **performance** scaling governor. • conservation: set the CPU into **power save** scaling governor. • standby: freeze the CPU.
set --mode conservation --second <second>	Wake up from an assigned timer to the active mode.
set --mode standby --second <second>	Wake up from an assigned RTC to the standby mode.

Wake Up Configuration

You can enable/disable a wake up on sources such as LAN and RTC using this command.

```
moxa@moxa-tbdjb1028636:~$ sudo mx-pwr-mgmt wakeupctl -h

Usage:
  mx-pwr-mgmt wakeupctl

Available Commands:
  List all available wakeup sources.
    $ mx-pwr-mgmt wakeupctl
  Enable a wakeup source.
    $ mx-pwr-mgmt wakeupctl --enable <dev>
  Disable a wakeup source.
    $ mx-pwr-mgmt wakeupctl --disable <dev>
moxa@moxa-tbdjb1028636:~$
```

```
moxa@moxa-tbdjb1028636:~$ sudo mx-pwr-mgmt wakeupctl
NAME  STATE
RTC0  disabled
LAN1  enabled
LAN2  enabled
moxa@moxa-tbdjb1028636:~$ sudo mx-pwr-mgmt wakeupctl --enable RTC0
```

Command and Usage	Description
mx-pwr-mgmt wakeupctl -h	Display the help menu.
mx-pwr-mgmt wakeupctl	Display all available wakeup sources and their status.
mx-pwr-mgmt wakeupctl --enable	Enable the specified wakeup source.
mx-pwr-mgmt wakeupctl --disable	Disable the specified wakeup source.

System Configuration for Power Management and Savings

Additional configurations commands for power mode management can be included on demand in the **/etc/moxa/moxa-power-manager/model.con.d/UC-3400A.conf** file.

```
root@moxa-imoxa1000038:/etc/moxa/moxa-power-manager/model.conf.d# cat UC-3400A.conf
[conservation_mode]
# Available Governors are: powersave, ondemand, or schedutil
governor=powersave

[spnd_peripheral]
# Default: All peripherals are power off during 'conservation' and 'standby' mode
conservation=WiFi1 Cellular1 LAN1 LAN2 USB SD LED
standby=WiFi1 Cellular1 USB SD LED

[LED]
name=RDY_Green

[spnd_systemd]
MCM=moxa-connection-manager.service
root@moxa-imoxa1000038:/etc/moxa/moxa-power-manager/model.conf.d#
```

Command and Usage	Description
conservation_mode	Specifies the governor to be used when the system enters Conservation mode. Available governors are: powersave, ondemand, or schedutil.
Spnd_peripheral	Lists the peripherals that should be powered off when the system enters Conservation or Standby mode.
LED	Lists the LEDs that should be turned off when the system enters Conservation or Standby mode. Note: Please refer to the MCIM tool to get the LED list.
spnd_system	Lists the systemd services that should be stopped when the system enters Conservation or Standby mode.

Further customization of power management is possible by adding to the scripts in the **custom-suspend.sh** and **custom-resume.sh** files in the follow path:

```
root@moxa-imoxa1000038:/etc/moxa/moxa-power-manager/scripts# ls
custom-resume.sh  custom-suspend.sh
```

Command and Usage	Description
custom-suspend.sh	The script will be executed before the system leaving Active mode.
custom-resume.sh	The script will be executed when the system returning to Active mode.

4. Using and Managing Computer Interfaces

In this chapter, we include more information on the Arm-based computer's interfaces, such as the serial interface, storage, diagnostic LEDs, and the wireless module. The instructions in this chapter cover all functions supported in Moxa's Arm-based computers. Before referring to the sections in this chapter, make sure that they are applicable to and are supported by the hardware specification of your Arm-based computer.

Moxa Computer Interface Manager (MCIM)

On many occasions, there isn't one standard method to access and configure specific interfaces on Moxa Arm-based computers because the hardware varies. Hence, programing across different Moxa Arm-based computer models can be difficult and time consuming. The goal of MCIM is to provide a unified software interface to access and configure non-standard computer interfaces. For example, MCIM can change the serial port interface mode (e.g., RS-232, RS-485-2W, RS-485-4W, and RS-422). However, configuring the serial port baudrate is not possible in MCIM because Linux provides a standard method to set the baudrate

MCIM is a command-line interface (CLI) Moxa utility designed to access and manage Moxa Arm-based computers' interfaces. Use the **`# sudo mx-interface-mgmt`** command to display the menu page.

Configuring the Log Level

To set the log level of MCIM, edit the configuration file **`/etc/moxa/MoxaComputerInterfaceManager/MoxaComputerInterfaceManager.con`**

Key	Value	Description
LOG_LEVEL	debug/info/warn/error	The log-level settings for the logs generated by MCIM for debugging and troubleshooting. The default level is "info"

Device Information

Use the **`# mx-interface-mgmt deviceinfo`** command to get information on your Moxa Arm-based computer.

Command and Usage	Description
deviceinfo	Show the following information: • Serial number (S/N) • Model name • SECUREBOOT (Enabled / Disabled)

LED Indicators

Use **# sudo mx-interface-mgmt led** command to get the list of controllable LEDs on your Arm-based computer.

Below is an example of the available LEDs on the UC-8220 Series. The returned NAME "L1" refers to the yellow LED for cellular signal, labeled "L1" on the device. For **LEDs** with **multiple colors** such as USR (yellow and green), 2 LED names will appears (USR_Yellow and USR_Green). For this type of LEDs, you must set the state of a color to "off" before setting another color to "on" or "blinking".

```
moxa@moxa-tbzkb1090923:~$ sudo mx-interface-mgmt led

NAME          LABEL                        STATE   ALIAS
W3             W3:yellow:signal            off     N/A
USR_Yellow     USR:yellow:programmable     off     N/A
USR_Green      USR:green:programmable      off     N/A
L1             L1:yellow:signal            off     N/A
W1             W1:yellow:signal            off     N/A
L2             L2:yellow:signal            off     N/A
W2             W2:yellow:signal            off     N/A
L3             L3:yellow:signal            off     N/A
```

The MCIM commands for LED indicator controls are listed in the following table:

Command and Usage	Description
led	Shows the following information for all controllable LEDs • Name (as labeled on the device) • Model series of the device • Color of the LED • Description of the LED • LED state (on/off/heartbeat)
led *<led_name>*	Show the above information of a **specified** LED
led *<led_name>* get_state	Get the current state (on/off/heartbeat) of a **specified** LED
led *<led_name>* set_state *<led_state>*	Set the state of a **specified** LED. Value of <state> can be **on**, **off**, or **heartbeat**

If an LED is common across multiple Moxa computer series, an ALIAS will be provided for that LED. You can use the alias in place of **<led_name>**.

An example of changing the current state of USR LED from **yellow** (steady) to **yellow** (blinking) is given below:

```
moxa@moxa-tbzkb1090923:~# sudo mx-interface-mgmt led USR_Yellow
NAME=SYS
LABEL= USR_Yellow
STATE=on
moxa@moxa-tbzkb1090923:~# sudo mx-interface-mgmt led USR_Yellow set_state
heartbeat
moxa@moxa-tbzkb1090923:~# sudo mx-interface-mgmt led USR_Yellow get_state
heartbeat
```

Storage and Partitions

Use **`# sudo mx-interface-mgmt disk and # sudo mx-interface-mgmt partition`** commands for managing the storage device and partitions.

Command and Usage	Description
`disk`	Show the following information of **all** embedded and external storage • Name (e.g., eMMC, USB, SD) • Device node (e.g., /dev/mmcblk0) • System disk (Y/N), if 'Y', it is the disk with MIL installed. • Number of partitions • Automount enabled/disabled (Y/N) • I/O state (enabled/disabled)
`disk <`*`disk_name`*`>`	Show the following information of a **specified** storage device • Name (e.g., eMMC, USB, SD) • Device node (e.g., /dev/mmcblk0) • System disk (Y/N), if 'Y', it is the disk with MIL installed. • Partition name and device node • Automount enabled/disabled (Y/N) • I/O state (enabled/disabled)
`disk <`*`disk_name`*`>` **`set_automount <`*`value`*`>`**	Set a **specified** external storage device (e.g., USB, SD) to automount when attach to device; <value> is true/false
`disk <disk_name>` **`set_io_state <io_state>`**	Set the I/O state for a specified USB or SD interface: • Enabled (default) • Disables the interface at the GPIO/driver level to reduce the attack surface when the interface is not in use *Note: Changing the I/O state requires a system reboot*
`partition`	Show the following information for partitions on **all** embedded and external storage devices: • Name (e.g., eMMC_p1, eMMC_p2, USB_p1) • Device node (e.g., /dev/mmcblk0p1) • Partition mounted (Y/N) • Partition mount point (e.g., /boot_device/p1) • Filesystem (e.g., ext4, FAT32)
`partition <`*`partition_name`*`>`	Show the above information of a **specified** partition
`partition <`*`partition_name`*`>` **`mount`**	Mount a **specified** partition
`partition <`*`partition_name`*`>` **`unmount`**	Unmount a **specified** partition
`partition <partition_name>` **`initialize_luks`**	Encrypts a non-system disk partition (e.g., USB, SD) using LUKS. The encrypted disk will only be mountable on a Moxa computer with the corresponding LUKS key file. *Note: The user will be prompted to set a minimum 8-character password. This password can be used to recreate the LUKS key file if needed.* **Recommendation:** Before Running the Command: • Ensure that the to-be-encrypted disk partition is not currently mounted or in use. • Do not run this command from within the directory where the partition is mounted, as it may interfere with the encryption process or cause unexpected errors. Password Security: • For enhanced security, it is recommended to use this command interactively, where the user is prompted to enter the password. This prevents the password from being exposed in system logs or command history.
`partition <partition_name>` **`initialize_luks -i`** **`<password>`**	Performs the above encryption function, but with the password provided as a parameter, bypassing the password prompt.

Command and Usage	Description
partition <partition_name> remap_luks	Remaps the encrypted disk to regenerate the LUKS key file. This is useful when you need to mount the encrypted disk on another Moxa computer that does not have the corresponding LUKS key file. **Recommendation:** For enhanced security, it is recommended to use this command interactively, where the user is prompted to enter the password. This prevents the password from being exposed in system logs or command history.
[Deprecated] **partition <partition_name> remap_luks -i <password>** **partition <partition_name> remap_luks p|--password <password>**	Performs the remapping function, but with the password provided as a parameter and bypassing the password prompt.

Below is an example of how to query available storage devices and set USB storage drive to automount:

```
moxa@moxa-tbzkb1090923:~$ mx-interface-mgmt disk
NAME   DEVICE         SYSTEM_DISK  NUMBER_OF_PARTITIONS  AUTOMOUNT_SETTING
USB    /dev/sdb       N              1                       false
eMMC   /dev/mmcblk0   Y              4                       false
moxa@moxa-tbzkb1090923:~$ sudo mx-interface-mgmt disk USB set_automount true
```

To query available partitions and mount the partition 1 of the USB storage drive, use the following command:

```
moxa@moxa-tbzkb1090923:~# mx-interface-mgmt partition
NAME       DEVICE           IS_MOUNTED  FS_TYPE  MOUNTPOINT
eMMC_p1    /dev/mmcblk0p1   Y           ext4     /boot_device/p1
eMMC_p2    /dev/mmcblk0p2   Y           ext4     /boot_device/p2
eMMC_p3    /dev/mmcblk0p3   Y           ext4     /boot_device/p3
eMMC_p4    /dev/mmcblk0p4   Y           ext4     /boot_device/p4
USB_p1     /dev/sdb1        N           N/A       N/A

moxa@moxa-tbzkb1090923:~# sudo mx-interface-mgmt partition USB_p1 mount
moxa@moxa-tbzkb1090923:~$ mx-interface-mgmt partition USB_p1
NAME=USB_p1
DEVICE=/dev/sdb1
IS_MOUNTED=Y
FS_TYPE=vfat
MOUNTPOINT=/media/USB_p1
```

WARNING

Setting external storage device to automount may expose your device to cybersecurity risks. It is strongly recommended that you not automount storage device unless your device is placed is in a highly secure environment.

Creating an Encrypted External Storage (e.g., USB, SD)

Below is an example of how to create an encrypted USB storage device:

1. Insert a USB card and use **`mx-interface-mgmt partition`** command to check the name of the available USB partition.

```
moxa@moxa-imoxa0920070:/home/moxa# sudo mx-interface-mgmt partition

NAME    DEVICE         IS_MOUNTED FS_TYPE MOUNTPOINT      MAPPER_DEVI UUID
SD_p1   /dev/mmcblk1p1 N          N/A     N/A             N/A         3e9f8825-1f7...
USB_p1  /dev/sda1      N          N/A     N/A             N/A         N/A
USB_p2  /dev/sda2      N          N/A     N/A             N/A         f4d582eb-f54...
USB_p3  /dev/sda3      N          N/A     N/A             N/A         N/A
eMMC_p1 /dev/mmcblk2p1 Y          ext4    /boot_device/p1 N/A         9ee65098-22a...
eMMC_p2 /dev/mmcblk2p2 Y          ext4    /boot_device/p2 N/A         ae9ebafe-629...
eMMC_p3 /dev/mmcblk2p3 Y          ext4    /boot_device/p3 N/A         fd7f9645-6b7...
eMMC_p4 /dev/mmcblk2p4 Y          ext4    /boot_device/p4 N/A         ab1aad4a-8a9...
```

2. Select a partition on the USB (e.g., USB_p1 for the 1st partition of the USB) to encrypt and set a password with a minimum length of 8 characters.

```
moxa@moxa-imoxa0920070:/home/moxa# sudo mx-interface-mgmt partition USB_p1
initialize_luks

[Warning]: Initializing a partition as LUKS will erase all data on the partition.
Enter password:
Re-enter password:
```

3. Now, USB_p1 is LUKS encrypted, and the corresponding LUKS key file is securely hashed using SHA-512 and stored on this computer. As a result, USB_p1 can only be mounted on this specific computer.
4. If the computer is ever restored to factory default or a new system image is installed, resulting in the loss of the LUKS key file, you can regenerate the key file using the **remap_luks** command by entering the password set in step #2. The same method can also be used when you want to mount the encrypted USB on a different Moxa computer with MIL3.

```
moxa@moxa-imoxa0920070:/home/moxa# sudo mx-interface-mgmt partition USB_p1 mount

Error: GDBus.Error:com.moxa.ComputerInterfaceManager.Error.Core.Failed: LUKS open
process failed: cannot get passphrase from config

moxa@moxa-imoxa0920070:/home/moxa# sudo mx-interface-mgmt partition USB_p1 remap_luks
Enter password:
moxa@moxa-imoxa0920070:/home/moxa# sudo mx-interface-mgmt partition USB_p1 mount
root@moxa-imoxa0920070:/home/moxa# sudo mx-interface-mgmt partition

NAME    DEVICE         IS_MOUNTED FS_TYPE MOUNTPOINT      MAPPER_DEVICE  UUID
SD_p1   /dev/mmcblk1p1 N          N/A     N/A             N/A            3e9f8825-1f7...
USB_p1  /dev/sda1      Y          ext4    /media/USB_p1   sda1_encrypted 44efd7d6-7ea...
USB_p2  /dev/sda2      N          N/A     N/A             N/A            f4d582eb-f54...
USB_p3  /dev/sda3      N          N/A     N/A             N/A            N/A
eMMC_p1 /dev/mmcblk2p1 Y          ext4    /boot_device/p1 N/A            9ee65098-22a...
eMMC_p2 /dev/mmcblk2p2 Y          ext4    /boot_device/p2 N/A            ae9ebafe-629...
eMMC_p3 /dev/mmcblk2p3 Y          ext4    /boot_device/p3 N/A            fd7f9645-6b7...
eMMC_p4 /dev/mmcblk2p4 Y          ext4    /boot_device/p4 N/A            ab1aad4a-8a9...
```

Serial Port

Configuring the Serial Interface via MCIM

Depending on the Moxa computer series, the serial ports support various operation modes, including RS-232, RS-422, RS-485 2-wire, and RS-485 4-wire, with flexible baudrate settings. The default operation mode is RS-232.

Use the **`# sudo mx-interface-mgmt serialport`** command to configure the operation mode, enable or disable the serial port, and adjust the resistor and terminator settings.

Command and Usage	Description
`serialport`	Shows the following information for **all** serial ports on the device: Name (as labeled on device) Device node (e.g., /dev/ttyM0) Current operation mode configured I/O state (enabled/disabled) Resistor ➢ Enabled: 1k-ohm pull-up/pull-down resistor applied ➢ Disabled (default): 150k-ohm pull-up/pull-down resistor applied ➢ N/A: This current operation mode (e.g., RS-232) doesn't support resistor Terminator ➢ Enabled: 120-ohm termination resistor applied ➢ Disabled (default): 120-ohm termination resistor not applied ➢ N/A: This current operation mode (e.g., RS-232) doesn't support terminator
`serialport <serialport_name>`	Shows the following information for a **specified** serial port: All the information described above Supported baudrates
`serialport <serialport_name> get_interface`	Gets the current operation mode for a **specified** serial port
`serialport <serialport_name> set_interface <serial_interface>`	Sets the operation mode for a **specified** serial port.
`serialport <serialport_name> set_io_state <io_state>`	Set the I/O state for a **specified** serial port: • Enabled (default) • Disables the interface at the GPIO/driver level to reduce the attack surface when the interface is not in use *Note: Changing the I/O state requires a system reboot*
`serialport <serialport_name> set_pull_up_down <state>`	Set the pull-up/pull-down resistor for a **specified** serial port: • Enabled: 1k-ohm pull-up/pull-down resistor applied • Disabled (default): 150k-ohm pull-up/pull-down resistor applied
`serialport <serialport_name> set_terminator <state>`	Set the 120-ohm termination resistor for a **specified** serial port: • Enabled: 120-ohm termination resistor applied • Disabled (default): 120-ohm termination resistor not applied

Changing the Serial Port Operation Mode

Use the **`serialport <port> set_interface`** command to change the operation mode of a serial port. For example, to change the mode of COM1 serial port from default RS-232 mode to the RS-422 mode, use the following commands:

```
moxa@moxa-imoxa0000005:~$ mx-interface-mgmt serialport
NAME  DEVICE      INTERFACE  IO_STATE  PULL_UP_DOWN_RESISTOR  TERMINATOR
P1    /dev/ttyM0  RS-422     enabled   N/A                    N/A
P2    /dev/ttyM1  RS-232     enabled   N/A                    N/A
moxa@moxa-imoxa0000005:~$ mx-interface-mgmt serialport P1
NAME=P1
DEVICE=/dev/ttyM0
SUPPORTED_INTERFACES=RS-232,RS-485-2W,RS-422,RS-485-4W
SUPPORTED_BAUDRATES=50,300,600,1200,1800,2400,4800,9600,19200,38400,57600,115200,230400,460800,921600
INTERFACE=RS-422
IO_STATE=enabled
PULL_UP_DOWN_RESISTOR=N/A
TERMINATOR=N/A
moxa@moxa-imoxa0000005:~$ sudo mx-interface-mgmt serialport P1 set_interface RS-422
moxa@moxa-imoxa0000005:~$ sudo mx-interface-mgmt serialport P1 get_interface
RS-422
moxa@moxa-imoxa0000005:~$
```

Changing Other Serial Interface Settings With STTY

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

Displaying All Settings

Use the following example to display all serial terminal settings of COM1 serial port.

```
moxa@moxa-tbzkb1090923:/# mx-interface-mgmt serialport
NAME  DEVICE
COM1  /dev/ttyM0
COM2  /dev/ttyM1

moxa@moxa-tbzkb1090923:/#  sudo stty -a -F /dev/ttyM0
speed 9600 baud; rows 0; columns 0; line = 0;
intr = ^C; quit = ^\; erase = ^?; kill = ^U; eof = ^D; eol = <undef>;
eol2 = <undef>; swtch = <undef>; start = ^Q; stop = ^S; susp = ^Z; rprnt = ^R;
werase = ^W; lnext = ^V; flush = ^O; min = 1; time = 0;
-parenb -parodd cs8 hupcl -cstopb cread clocal -crtscts
-ignbrk -brkint -ignpar -parmrk -inpck -istrip -inlcr -igncr icrnl ixon -ixoff
-iuclc -ixany -imaxbel -iutf8
opost -olcuc -ocrnl onlcr -onocr -onlret -ofill -ofdel nl0 cr0 tab0 bs0 vt0 ff0
isig icanon iexten echo echoe echok -echonl -noflsh -xcase -tostop -echoprt
echoctl echoke
```

Configuring Serial Settings

The following example changes the baudrate to 115200.

```
moxa@moxa-tbzkb1090923:~$ sudo stty 115200 -F /dev/ttyM0
```

Check the settings to confirm that the baudrate has changed to 115200.

```
moxa@moxa-tbzkb1090923:~$ sudo stty -a -F /dev/ttyM0
speed 115200 baud; rows 0; columns 0; line = 0;
intr = ^C; quit = ^\; erase = ^?; kill = ^U; eof = ^D; eol = <undef>;
eol2 = <undef>; swtch = <undef>; start = ^Q; stop = ^S; susp = ^Z; rprnt = ^R;
werase = ^W; lnext = ^V; flush = ^O; min = 1; time = 0;
-parenb -parodd cs8 hupcl -cstopb cread clocal -crtscts
-ignbrk -brkint -ignpar -parmrk -inpck -istrip -inlcr -igncr icrnl ixon -ixoff
-iuclc -ixany -imaxbel -iutf8
opost -olcuc -ocrnl onlcr -onocr -onlret -ofill -ofdel nl0 cr0 tab0 bs0 vt0 ff0
isig icanon iexten echo echoe echok -echonl -noflsh -xcase -tostop -echoprt
echoctl echoke
```

NOTE

Detailed information on the **stty** utility is available at the following link:

https://manpages.debian.org/bullseye/coreutils/stty.1.en.html

Ethernet Interface

Use **# sudo mx-interface-mgmt ethernet** command to configure the Ethernet ports.

Command and Usage	Description
ethernet	Show the following information of **all** ethernet ports on the device. • Name (as labeled on device) • Network interface name (eth0, eth1, etc.) • I/O state (enabled/disabled)
ethernet <ethernet_name>	Show the above information of a **specified** ethernet port
ethernet <ethernet_name> set_io_state <io_state>	Set the I/O state for a **specified** ethernet port: • Enabled (default) • Disables the interface at the GPIO/driver level to reduce the attack surface when the interface is not in use *Note: Changing the I/O state requires a system reboot*

```
moxa@moxa-tbzkb1090923:~$ mx-interface-mgmt ethernet
NAME  DEVICE_NAME
LAN1  eth0
LAN2  eth1
moxa@moxa-tbzkb1090923:~$ mx-interface-mgmt ethernet LAN1
NAME=LAN1
DEVICE_NAME=eth0
moxa@moxa-tbzkb1090923:~$
```

Serial Console Interface

Use the **# mx-interface-mgmt console** command to configure the serial console port.

Command and Usage	Description
console	Show the following information for the console port. • Name (as labeled on the device) • Device node (e.g., /dev/ttyS0)
Console <console_name>	Show the above information of a specified serial console interface
Console <console_name> set_io_state <io_state>	Set the I/O state for a **specified** console port: • Enabled (default) • Disables the interface at the GPIO/driver level to reduce the attack surface when the interface is not in use *Note: Changing the I/O state requires a system reboot*

NOTE

If both the serial console interface and ethernet are disabled, and you cannot access Linux through the console port to enable the interface, you can access the bootloader menu and navigate to **Advanced Settings > Enable/Disable Interfaces** to enable the serial console port.

Following is an example of showing the console port device node.

```
root@moxa-tbzkb1090923:~# mx-interface-mgmt console
NAME     DEVICE
Console  /dev/ttyS0
root@moxa-tbzkb1090923:~#
```

Digital Input/Output (DIO)

Use the **`# sudo mx-interface-mgmt dio`** command to query and configure the state for each digital input/output (DIO) interface, and also configure the hook script.

Command and Usage	Description
`dio`	Shows the following information of **all** DIO interfaces: • Name (as labeled on device) • State (high/low) • Event (high/low/change) • GPIO pin • Direction (input/output)
`dio <dio_name>`	Shows the following information for a **specified** DI or DO interface • Name (as labeled on device) • State (high/low) • Event (high/low/change) • GPIO pin • Direction (input/output) • Hook name: the name of the hook script assigned to this DI interface. • Direct configurable: Indicates whether this digital interface can be configured as either DI or DO (true if configurable).
`dio <dio_name> get_state`	Gets the current state (high/low) of a **specified** DI or DO interface
`dio <dio_name> set_state <dio_state>`	Sets the state (high/low) of a **specified** DO interface
`dio <di_name> get_event`	Gets the current event setting (none, falling, rising, or change) for the specified DI interface
`dio <di_name> set_event <di_event>`	Sets the event (none, falling, rising, or change) for the specified DI interface
`dio <dio_name> get_direction`	Gets the direction (input/output) for the specified DIO interface
`dio <dio_name> set_direction <direction>`	Sets the direction (input/output) for the specified DIO interface. This function is only available on computers with a DIO interface that can be configured as either DI (Digital Input) or DO (Digital Output)
[Deprecated] `dio <dio_name> reload`	[Deprecated] Reload the DIO configuration after using the **set_event** or **set_direction** command
`dio add_hook <hook_name> <hook_path>`	Adds a user-defined hook script: • **hook_name:** The alias name of the hook script. Allowed characters include uppercase letters (A–Z), lowercase letters (a–z), digits (0–9), hyphens (-), and underscores (_) • **hook_path:** The file path of the source hook script. For example: /home/moxa/close_gate.sh. When a hook is added, the script will automatically be copied into the directory: /etc/moxa/MoxaComputerInterfaceManager/dio-scripts
`dio delete_hook <hook_name>`	Delete a hook script
`dio list_hook`	Lists the added hook scripts • **Name:** The alias name of the hook script • **Path:** The file path of the associated hook script
`dio <dio_name> set_hook <hook_name>`	Assigns a hook script to a specified button
[Deprecated] `dio <dio_name> add_hook <edge> <path>`	[Deprecated] Executes a script from a **specified** path when the dio state changes. • <edge>specifies whether the trigger should react to a "rising" or "falling" edge. • <path>is the path to the script that should be executed when the specified edge transition occurs.
[Deprecated] `dio <dio_name> remove_hook <edge>`	[Deprecated] Removes the edge script (rising/falling) of an interface

NOTE

- The predetermined state of the digital output interface is high (open circuit).
- Starting from MIL 3.3, the **set_event** and **get_event** function will support only the DI interface, as it makes more sense for customers to define the desired actions in their code when controlling the DO state.
- Starting from MIL 3.1, **add_hook** and **remove_hook** have been replaced by a more flexible configuration method, as described below.

Example: Adding a Hook Script for the DI1 Port

This example shows how to create and assign a hook script named **"ALERT"** to the first digital input (DI1) of the UC-4400A Series computer, and configure it to automatically execute when DI1 changes to a rising state (pulled high).

For MIL 3.4 and later

1. Create a script named **DI1_ALERT.sh** with following content that will log a line into /var/log/di1.log when executed

```
#!bin/bash
echo "The input value of Digital Input 1 (DI1) has changed" >>
/var/log/di1.log
```

2. Create a new hook named **DI1_ALERT** using **mx-interface-mgmt dio add_hook**

```
root@moxa-imoxa1000030:/# sudo mx-interface-mgmt dio add_hook
DI1_ALERT ./home/moxa/DI1_ALERT.sh
```

3. Assign the hook **DI1_ALERT** to first digital input (DI1) of UC-4400A

```
root@moxa-imoxa1000030:/# sudo mx-interface-mgmt dio DI1 set_hook DI1_ALERT
```

4. Check if the hook is set correctly for DI1

```
root@moxa-imoxa1000030:/# mx-interface-mgmt dio DI1
NAME=DI1
STATE=high
EVENT=none
ACTIVE=no
GPIO_PIN=496
DIRECTION=input
DIRECTION_CONFIGURABLE=no
HOOK_NAME=DI1_ALERT
```

5. Set DI1 to run **DI1_ALERT** script when the DI state is changed to rising (pull high)

```
root@moxa-imoxa1000030:/# sudo mx-interface-mgmt dio DI1 set_event rising
root@moxa-imoxa1000030:/# mx-interface-mgmt dio

NAME   STATE   EVENT    ACTIVE   GPIO_PIN   DIRECTION
DI1    high    rising   yes      496        input
DI2    high    none     no       497        input
DI3    high    none     no       498        input
DI4    high    none     no       499        input
DO1    high    none     no       500        output
DO2    high    none     no       501        output
DO3    high    none     no       502        output
DO4    high    none     no       503        output
```

For earlier MIL3 version

Starting with MIL 3.1, we have introduced a more flexible method for configuring the hook script for DI's edge transitions. Detailed instructions can be found in the 'README' file located in the directory '**/etc/moxa/MoxaComputerInterfaceManager/dio-scripts**'.

An example of setting up the Moxa Computer Interface Manager (MCIM) to automatically execute a script when the signal of the first digital input (DI1) changes from low to high (rising) or from high to low (falling) is outlined below:

1. Navigate to '**/etc/moxa/MoxaComputerInterfaceManager/dio-scripts/**' and create a script named 'DI1.script' :
2. Add the following content to DI1.script to log an event whenever DI1 changes:

```
#!bin/bash
echo "The input value of Digital Input 1 (DI1) has changed" >>
/var/log/di1.log
```

3. Make the script executable

```
root@moxa-tbzgb1057611:/etc/moxa/MoxaComputerInterfaceManager/dio-scripts#
chmod +x DI1.script
```

4. Open '**peripheral-settings.conf**' located in '**/etc/moxa/MoxaComputerInterfaceManager/**' and set the event value for [DIO/DI1] to 3 to detect both rising and falling edges:

   The event ID corresponds to the following actions:

   - Event=0: none (default)
   - Event=1: signal change from high to low (falling)
   - Event=2: signal change from low to high (rising)
   - Event=3: signal change from low to high (rising) or high to low (falling)

```
[DIO/DI1]
Event=3
```

5. Apply the changes by restarting the MoxaComputerInterfaceManager service:

```
root@moxa-tbzgb1057611: systemctl restart MoxaComputerInterfaceManager
```

NOTE

In MIL versions 3.2 and 3.3, steps 4 and 5 above can be replaced by the **set_event** and **reload** commands respectively.

6. Ensure the script is correctly configured and active by checking the settings. The '**EVENT**' should be set to '**change**' and '**ACTIVE**' should show '**yes**':

```
root@moxa-tbcib1078122: mx-interface-mgmt dio

NAME  STATE  EVENT   ACTIVE
DI1   high   change  yes
DI2   high   none    no
DI3   high   none    no
DI4   high   none    no
DO1   high   none    no
DO2   high   none    no
DO3   high   none    no
DO4   high   none    no
```

Buzzer

Use the **# sudo mx-interface-mgmt buzzer** command to query and set the state for buzzer alarm in Moxa Arm-based computers with a buzzer.

Command and Usage	Description
buzzer	Show the following information of **all** buzzers • Name • State (on/off) • Device Type • GPIO pin
buzzer <buzzer_name>	Show the following information of a **specified** buzzer • Name • State (on/off) • Device Type • GPIO pin
buzzer <buzzer_name> get_state	Get the current state (on/off) of a **specified** buzzer
buzzer <buzzer_name> set_state	Set the state (on/off) of a **specified** buzzer

Cellular Module Interface

Use **# sudo mx-interface-mgmt cellular** command to query and manage cellular module(s)

Command and Usage	Description
cellular	Show the following information for **all** cellular modules. • Name (e.g., Cellular1) • Network interface name (wwan0, wwan1, etc.) • Cellular module detected (true/false)
cellular <name>	Show the detail information of a **specified** cellular module • Name (e.g., Cellular1) • Network interface name (wwan0, wwan1) • Cellular module detected (true/false) • QMI Port (e.g., /dev/cdc-wdm0) • AT Port (e.g., /dev/ttyUSB4) • GPS Port (e.g., /dev/ttyUSB3) if GPS is supported • Cellular module power status (on/off) • Number of available SIM slots on the device • The SIM slot # that is currently used by the cellular module *Note: SIM slot # correspond to the labeled slot # on the device*
cellular <name> get_power	Get the cellular module power status (on/off).
cellular <name> set_power <power_state>	Set the cellular module power status (on/off). *Note: Module will power-on when device reboot*
cellular <name> get_sim_slot	Get the SIM slot # that is currently used by the cellular module
cellular <name> set_ sim_slot <sim_slot>	Set the SIM slot # used by cellular module. Module power off/on is required for SIM slot changed to take effect. *Note: SIM slot # will be set to default (slot 1) when the device reboot*

NOTE

1. Some cellular modules may not support power on/off or SIM slot control.
2. If you are using Moxa Connection Manager (MCM) to manage the cellular connection, do not use set_power or sim_slot commands as they might interrupt MCM's network failover/failback operations.

An example of using MCIM to query the cellular module information and changing the SIM slot # use by the module from slot 1 to 2 is given below:

```
moxa@moxa-tb10923:~$ mx-interface-mgmt cellular
NAME        DEVICE_NAME    DEVICE_DETECTED
Cellular1   wwan0          true
moxa@moxa-tb10923:~$ mx-interface-mgmt cellular Cellular1
NAME=Cellular1
DEVICE_NAME=wwan0
QMI_PORT=/dev/cdc-wdm0
AT_PORT=/dev/ttyUSB4
GPS_PORT=/dev/ttyUSB3
DEVICE_DETECTED=true
POWER=on
SIM_SLOT_NUMBER=2
SIM_SLOT=1
moxa@moxa-tb10923:~$ sudo mx-interface-mgmt cellular Cellular1 set_sim_slot 2
moxa@moxa-tb10923:~$ mx-interface-mgmt cellular Cellular1 get_sim_slot
2
```

Wi-Fi Module Interface

Use the **# sudo mx-interface-mgmt wifi** command to query and manage Wi-Fi modules.

Command and Usage	Description
wifi	Shows the following information of **all** Wi-Fi modules. • Name (e.g., WiFi1) • Network interface name (wlan0, wlan1) • Wi-Fi module detected (true/false)
wifi <name>	Shows the above information for a **specified** Wi-Fi module
wifi <name> get_power	Gets the Wi-Fi module power status (on/off).
wifi <name> set_power <power_state>	Set the Wi-Fi module power status (on/off). *Note: The module will power-on when the device reboots.*

NOTE

Some Wi-Fi modules may not support power on/off control.

Socket Interface

Use the **# sudo mx-interface-mgmt socket** command manage the Mini PCI-E sockets on the Moxa Arm-based Computer

Command and Usage	Description
socket	List all the available sockets' name (e.g., Socket1, Socket2)
socket <socket_name>	Shows the following information for a **specified** Mini PCI-E socket • Name (e.g., Socket1, Socket2) • Power status (on/off) • Number of available SIM slots if a cellular module is insert to this Mini PCI-E socket • Get the SIM slot # that is currently used by the cellular module on this Mini PCI-E socket *Note: SIM slot # correspond to the labeled slot # on the device.*
socket < socket_name> get_power	Gets the power status (on/off) for a **specified** Mini PCI-E socket
socket <name> set_power <power_state>	Set the power status (on/off) for a **specified** Mini PCI-E socket. *Note: The socket will power-on when the device reboots.*

CAN Port

The CAN ports on Moxa's Arm-based computers support CAN 2.0A/B standard.

Configuring the CAN Interface via MCIM

Use the **`# sudo mx-interface-mgmt can`** command disable/enable CAN interface and configure the 120-ohm termination resistor

Command and Usage	Description
`can`	Shows the following information of **all** CAN interfaces. • Name (as labeled on the device, e.g., P3) • Devic name (e.g., can0, can1, can2) • Bitrate • I/O state (enabled/disabled) • Terminator ➢ Enabled: 120-ohm termination resistor applied ➢ Disabled (default): 120-ohm termination resistor not applied
`can <can_name>`	Shows the following information for a **specified** CAN interface. • All the information described above • Maximum supported bitrate
`can <can_name> get_bitrate`	Gets the bitrate for a **specified** CAN interface
`can <can_name> set_io_state <io_state>`	Set the I/O state for a **specified** CAN interface: • Enabled (default) • Disables the interface at the GPIO/driver level to reduce the attack surface when the interface is not in use *Note: Changing the I/O state requires a system reboot*
`can <can_name> set_terminator <terminator_state>`	Set the 120-ohm termination resistor for a **specified** CAN interface: • Enabled: 120-ohm termination resistor applied • Disabled (default): 120-ohm termination resistor not applied

Configuring the CAN Interface via ip link

The CAN ports are initialized by default. If any additional configuration not supported by MCIM are required, use the **`ip link`** command to check the CAN device.

To check the CAN device status, use the **`ip link`** command.

```
# ip link
can0: <NOARP,UP,LOWER_UP,ECHO> mtu 16 qdisc pfifo_fast state UNKNOWN mode
DEFAULT group default qlen 10 link/can
```

To configure the CAN device, use **`# ip link set can0 down`** to turn off the device first

```
# ip link set can0 down
# ip link
can0: <NOARP,ECHO> mtu 16 qdisc pfifo_fast state DOWN mode DEFAULT group
default qlen 10 link/can
```

Here's an example with bitrate 12500:

```
# ip link set can0 up type can bitrate 12500
```

CAN Bus Programming Guide

The following code is an example of the SocketCAN API, which sends packets using the raw interface.

CAN Write

```
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <string.h>
#include <net/if.h>
#include <sys/types.h>
#include <sys/socket.h>
#include <sys/ioctl.h>
#include <linux/can.h>
#include <linux/can/raw.h>
int main(void)
{
    int s;
    int nbytes;
    struct sockaddr_can addr;
    struct can_frame frame;
    struct ifreq ifr;
    char *ifname = "can1";
    if((s = socket(PF_CAN, SOCK_RAW, CAN_RAW)) < 0) {
        perror("Error while opening socket");
        return -1;
    }
    strcpy(ifr.ifr_name, ifname);
    ioctl(s, SIOCGIFINDEX, &ifr);
    addr.can_family = AF_CAN;
    addr.can_ifindex = ifr.ifr_ifindex;
    printf("%s at index %d\n", ifname, ifr.ifr_ifindex);
    if(bind(s, (struct sockaddr *)&addr, sizeof(addr)) < 0) {
        perror("Error in socket bind");
        return -2;
    }
    frame.can_id = 0x123;
    frame.can_dlc = 2;
    frame.data[0] = 0x11;
    frame.data[1] = 0x22;
    nbytes = write(s, &frame, sizeof(struct can_frame));
    printf("Wrote %d bytes\n", nbytes);
    return 0;
}
```

CAN Read

The following sample code illustrates how to read the data.

```
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <string.h>
#include <net/if.h>
#include <sys/types.h>
#include <sys/socket.h>
#include <sys/ioctl.h>
#include <linux/can.h>
#include <linux/can/raw.h>
Int main(void)
{
    int i;
    int s;
    int nbytes;
    struct sockaddr_can addr;
    struct can_frame frame;
    struct ifreq ifr;
    char *ifname = "can0";
    if((s = socket(PF_CAN, SOCK_RAW, CAN_RAW)) < 0) {
        perror("Error while opening socket");
        return -1;
    }
    strcpy(ifr.ifr_name, ifname);
    ioctl(s, SIOCGIFINDEX, &ifr);
    addr.can_family = AF_CAN;
    addr.can_ifindex = ifr.ifr_ifindex;
    printf("%s at index %d\n", ifname, ifr.ifr_ifindex);
    if(bind(s, (struct sockaddr *)&addr, sizeof(addr)) < 0) {
        perror("Error in socket bind");
        return -2;
    }
    nbytes = read(s, &frame, sizeof(struct can_frame));
    if (nbytes < 0) {
        perror("Error in can raw socket read");
        return 1;
    }
    if (nbytes < sizeof(struct can_frame)) {
        fprintf(stderr, "read: incomplete CAN frame\n");
        return 1;
    }
    printf(" %5s %03x [%d] ", ifname, frame.can_id, frame.can_dlc);
    for (i = 0; i < frame.can_dlc; i++)
        printf(" %02x", frame.data[i]);
    printf("\n");
    return 0;
}
```

After you use the SocketCAN API, the SocketCAN information is written to the paths: **/proc/sys/net/ipv4/conf/can*** and **/proc/sys/net/ipv4/neigh/can***

Push-button

A push button is available on Moxa Arm-based computers. The default actions of this button are described below:

Button Action	LED Indicator Status	Resulting Action
Press and hold FN button and release within 1s	System LED blinks	Device reboot
Press and hold FN button and release between 7s to 9s	• System LED blinks for 1s to 6s • System LED is ON for 7s to 9s	Reset to factory default
Press and hold FN button and release after 9s	• System LED blinks for 1s to 6s • System LED is ON for 7s to 9s • System LED is OFF after 9s	Do nothing; cancel action

NOTE

The System LED may be labeled as **USR**, **RDY**, or **READY** depending on the model of your Moxa Arm-based computer.

Configuring Actions for Buttons

Use **`# sudo mx-interface-mgmt button`** command to display all buttons on your computer and manage the button actions.

Command and Usage	Description
`button`	Shows the following information for **all** buttons on the device: • Name (as labeled on device) • Action: ➢ **default:** Button behavior is set to the factory default action. ➢ **snapshot:** When pressing and holding the button and releasing between 7s to 9s, the system restores the computer to the user-created snapshot instead of the factory default ➢ **disabled:** The button has no function when pushed ➢ **customized (user-defined):** Users can create their own action script and assign it to the button. The action name will appear as defined by the user (e.g., *shutdown, backup, sendlog*). Custom actions can be created with the **add_action** command and applied to a button with the **set_action** command (see the command descriptions below)
`button <name>`	Shows the above information for a **specified** button
`button add_action <action_name> <action_path>`	Adds a customized (user-defined) action: • **action_name:** The name of the action. Allowed characters include uppercase letters (A–Z), lowercase letters (a–z), digits (0–9), hyphens (-), and underscores (_) • **action_path:** The file path of the source bash script. For example: /home/moxa/shutdown_button.sh. When the action is added, the script will automatically be copied into the directory: /etc/moxa/MoxaComputerInterfaceManager/button-scripts/
`button delete_action <action_name>`	Deletes a customized (user-defined) action
`button list_action`	Lists the MIL pre-defined actions and user-defined actions: • **Name:** The name of the action • **Reserved:** Indicates whether the action is pre-defined by MIL (true) or created by the user (false) • **Path:** The file path of the associated bash script
`button <button_name> get_action`	Retrieves the action currently assigned to the **specified** button
`button <button_name> set_action <action_name>`	Assigns an action to a **specified** button

NOTE

The **button** commands shown in the table are based on MIL v3.4. Some of these commands may not be available in earlier MIL versions. Refer to the example usage section below for detailed instructions applicable to different MIL versions.

Following is an example of using MCIM to query an available button (labeled as FN on device) of the UC-8200 series.

```
root@moxa-tb10923:~# mx-interface-mgmt button
NAME  Action
FN    default
```

In contrast, on the UC-3400A the push button is labeled RESET. The example below shows that its action has been configured as snapshot. When pressing and holding the button, then releasing it between 7 and 9 seconds, the system restores the user-created snapshot instead of the factory default:

```
root@moxa-tb10226:/home/moxa# mx-interface-mgmt button
NAME   ACTION
RESET  snapshot
```

Example: Adding a Custom Action

This example demonstrates how to create and assign a customized button action named "SHUTDOWN" on the UC-3400A series, which performs the following function:

Button Action	LED Indicator Status	Resulting Action
Press and hold the "RESET" button and release within 1 second	READY LED blinks	Device reboot
Press and hold the "RESET" button and release between 7 to 9 seconds	• READY LED blinks for 1 to 6 seconds • READY LED is ON for 7 to 9 seconds	Restore the system from a snapshot
Press and hold the "RESET" button for longer than 15 seconds then release	• READY LED blinks for 1s to 6s • READY LED is ON for 7s to 9s • READY LED is blinks after 10s	Shutdown the computer

The contents of the script are shown below:

```
#!/bin/sh

ACTION="${1}"
SECONDS="${2}"

if [ "${ACTION}" = "press" ]; then
    /usr/bin/mx-interface-mgmt led RDY_Red set_state off
    /usr/bin/mx-interface-mgmt led RDY_Green set_state off

elif [ "${ACTION}" = "hold" ]; then
    if [ ${SECONDS} -eq 1 ]; then
        /usr/bin/mx-interface-mgmt led RDY_Green set_state heartbeat
    elif [ ${SECONDS} -eq 7 ]; then
        /usr/bin/mx-interface-mgmt led RDY_Green set_state on
    elif [ ${SECONDS} -eq 9 ]; then
        /usr/bin/mx-interface-mgmt led RDY_Green set_state off
    elif [ ${SECONDS} -ge 10 ]; then
        /usr/bin/mx-interface-mgmt led RDY_Green set_state heartbeat
    fi

elif [ "${ACTION}" = "release" ]; then
    if [ ${SECONDS} -lt 1 ]; then
        /usr/sbin/reboot
    elif [ ${SECONDS} -ge 7 ] && [ ${SECONDS} -lt 9 ]; then
        /usr/bin/mx-interface-mgmt led RDY_Green set_state heartbeat
```

```
        if /usr/sbin/mx-system-mgmt snapshot info > /dev/null 2>&1; then
            /usr/sbin/mx-system-mgmt snapshot restore --yes
            /usr/sbin/reboot
        else
            echo "Error: There is no snapshot information"
            /usr/bin/mx-interface-mgmt led RDY_Green set_state off
            exit 1
        fi
    elif [ ${SECONDS} -ge 15 ]; then
        /sbin/shutdown -h now
    fi
    /usr/bin/mx-interface-mgmt led RDY_Green set_state on
fi

exit 0
```

For MIL 3.4 and later

1. Create a script named **shutdown_button.sh** under path **/home/moxa/** with content above
2. Create a user-defined action named **SHUTDOWN** and specify the path of the script

```
root@moxa-tb10226:/# sudo mx-interface-mgmt button add_action SHUTDOWN
/home/moxa/shutdown_button.sh
```

3. Confirms the action is created successfully

```
root@moxa-tb10226:/home/moxa# mx-interface-mgmt button list_action
NAME         RESERVED PATH
SHUTDOWN     false    /etc/moxa/MoxaComputerInterfaceManager/button-
scripts/shutdown_button.sh
default      true     /etc/moxa/MoxaComputerInterfaceManager/button-
scripts/uc3400a-default.script
snapshot     true     /etc/moxa/MoxaComputerInterfaceManager/button-
scripts/uc3400a-snapshot.script
user-defined true     /etc/moxa/MoxaComputerInterfaceManager/button-
scripts/custom.script
```

4. Check the push button name. In this example, the result shows that the name is **RESET**

```
root@moxa-tb10226:/# mx-interface-mgmt button
NAME   ACTION
RESET  default
```

5. Set the **RESET** button to use the newly created action SHUTDOWN

```
root@moxa-tb10226:/# sudo mx-interface-mgmt button RESET set_action SHUTDOWN
root@moxa-tb10226:/# mx-interface-mgmt button
NAME   ACTION
RESET  SHUTDOWN
```

For MIL 3.2 and 3.3

The **add_action** command is not available in MIL 3.2 and 3.3.
Follow the instructions below to configure the button with a user-defined action:

1. Edit the content of **/etc/moxa/MoxaComputerInterfaceManager/button-scripts/custom.script** to implement the desired action, as shown in the example script above.
2. Set the **RESET** button to action user-defined

```
root@moxa-tb10226:/# sudo mx-interface-mgmt button RESET set_action user-
defined
root@moxa-tb10226:/# mx-interface-mgmt button
NAME   ACTION
RESET  SHUTDOWN
```

3. Once configured, the **RESET** button will execute the script defined in **/etc/moxa/MoxaComputerInterfaceManager/button-scripts/custom.script** whenever it is pressed.

NOTE

MIL 3.4 continues to support the above legacy method of editing **custom.script** and setting the action to **user-defined**. Therefore, existing button actions configured in MIL 3.2 or 3.3 will remain valid after upgrading to MIL 3.4. However, this method is **deprecated**. We strongly recommend using the new commands introduced in MIL 3.4.

For MIL 3.0 and 3.1

The **add_action** and **set_action** are not available in MIL 3.0 and 3.1.
Follow the instructions below to configure the button with a user-defined action:

1. Edit the content of **/etc/moxa/MoxaComputerInterfaceManager/button-scripts/custom.script** to implement the desired action, as shown in the example script above.
2. Edit **/etc/moxa/MoxaComputerInterfaceManager/peripheral-settings.conf**.
3. Configure the **Action** parameter to **2**. The device needs to be rebooted for the settings to take effect.

   The **Action** parameter in **peripheral-settings.conf** can have the following three values:

   - 0: Disable the button (no action when pressed)
   - 1: Run the default script
   - 2: Run the custom script

NOTE

You must reboot the system for the settings to take effect.

An example of the settings in the **peripheral-settings.conf** file is shown below:

```
[Button/FN]
Action=2
[Disk/eMMC]
AutoMount=false
[Disk/SD]
AutoMount=false
[Disk/USB]
AutoMount=true
[SerialPort/COM1]
Interface=1
[SerialPort/COM2]
Interface=0
```

4. Once configured, the button will execute the script defined in **/etc/moxa/MoxaComputerInterfaceManager/button-scripts/custom.script** whenever it is pressed.

NOTE

This legacy method is only supported in MIL 3.0 and 3.1. When upgrading to a newer MIL version, ensure that you switch to the new configuration commands.

Configuring the Real COM Mode

You can use Moxa's NPort Series serial device drivers to extend the number of serial interfaces (ports) on your UC computer. The NPort comes equipped with COM drivers that work with Windows systems and TTY drivers for Linux systems. The driver establishes a transparent connection between the UC computer and serial device by mapping the IP Port of the NPort's serial port to a local pseudo COM/TTY port on the UC computer.

In addition, the Real COM mode supports up to 4 simultaneous connections, so that multiple hosts (e.g., PC or UC computer) can collect data from a serial device at the same time.

One of the major conveniences of using the Real COM mode is that it allows users to continue using RS-232/422/485 serial communications software that was written for pure serial communications applications. The driver intercepts data sent to the host's COM port, packs it into a TCP/IP packet, and then redirects it through the host's Ethernet card. At the other end of the connection, the NPort accepts the Ethernet frame, unpacks the TCP/IP packet, and then sends it transparently to the appropriate serial device attached to one of the NPort's serial ports.

To install the Real COM driver on the UC computer, download the driver using apt from Moxa software repository that is accessible over the Internet. You will be able to view the driver-related files in the /usr/lib/npreal2/driver folder after a successful installation.

```
root@moxa-tb10923:~# sudo apt update && apt install moxa-nport-real-tty-utils
```

```
> mxaddsvr (Add Server, mapping tty port)
> mxdelsvr (Delete Server, unmapping tty port)
> mxloadsvr (Reload Server)
> mxmknod (Create device node/tty port)
> mxrmnod (Remove device node/tty port)
> mxuninst (Remove tty port and driver files)
```

Now, load the driver using the command:

```
root@moxa-tb10923:~# modprobe npreal2
```

To ensure the driver loads automatically at each system bootup, run the command below to create a configuration file:

```
root@moxa-tb10923:~# echo "npreal2" > /etc/modules-load.d/npreal2.conf
```

At this point, you will be ready to map the NPort serial port to the system **tty** port. For a list of supported NPort devices and their revision history, click https://www.moxa.com/en/support/search?psid=50278.

Mapping TTY Ports

First of all ensure that you set the operation mode of the desired NPort serial port to Real COM mode. After logging in as a super user, enter the directory /usr/lib/npreal2/driver and then run the mxaddsvr command to map the target NPort serial port to the host tty ports.

The syntax of **mxaddsvr** command is as follows:

mxaddsvr [NPort IP Address] [Total Ports] ([Data port] [Cmd port])

The **mxaddsvr** command performs the following actions:

- Modifies the npreal2d.cf
- Creates tty ports in the /dev directory with major & minor numbers configured in npreal2d.cf
- Restarts the driver.

Mapping TTY Ports (automatic)

To map tty ports automatically, run the **mxaddsvr** command with just the IP address and the number of ports, as shown in the following example:

```
# cd /usr/lib/npreal2/driver
# ./mxaddsvr 192.168.3.4 16
```

16 tty ports will be added, all with IP 192.168.3.4, consisting of data ports from 950 to 965 and command ports from 966 to 981.

ATTENTION

You must reboot the system after mapping tty ports with **mxaddsvr**.

Mapping TTY Ports (manual)

To map tty ports manually, run the **mxaddsvr** command and specify the data and command ports as shown in the following example:

```
 # cd /usr/lib/npreal2/driver
# ./mxaddsvr 192.168.3.4 16 4001 966
```

In this example, 16 tty ports will be added, all with IP 192.168.3.4; data ports from 4001 to 4016 and command ports from 966 to 981.

ATTENTION

You must reboot the system after mapping the tty ports with **mxaddsvr**.

Removing Mapped TTY Ports

After logging in as root, enter the directory /usr/lib/npreal2/driver and run the **mxdelsvr** command to delete a server. The syntax of **mxdelsvr** is as follows:

mxdelsvr [IP Address]

Example:

```
# cd /usr/lib/npreal2/driver
# ./mxdelsvr 192.168.3.4
```

The following actions are performed when you run the **mxdelsvr** command:

1. Modifies npreal2d.cf
2. The relevant tty ports ae removed from the /dev directory
3. Restarts the driver

If the IP address is not provided in the command line, the program will list the installed servers and total ports on the screen. You will need to choose a server from the list for deletion.

5. Configuring and Managing Networks

Moxa Connection Manager (MCM)

MCM is a network management utility developed by Moxa to manage the LAN and WAN network on your Moxa Arm-based computer, including Wi-Fi, cellular, and ethernet interfaces. With MCM, you can easily fill in the connection profile and priority in the configuration file; then MCM will automatically connect and keep the connection alive. Following are the major features of MCM:

- Cellular, Ethernet and Wi-fi connection
- Connection auto keep-alive, failover, and failback
- DHCP server
- Data usage monitoring
- Cellular connection diagnosis tool
- Cellular modem and network information
- Cellular modem firmware upgrade with failback

IMPORTANT!

You can find the detailed online user manual for the Moxa Connection Manager (MCM) at the following link: Moxa Connection Manager Reference Manual

Following is default configuration of Moxa Connection Manager (MCM):

Interface	Default Managed by MCM	Network Configuration
LAN1	Yes	• Set as DHCP WAN by default. • After boot up, if LAN1 cannot obtain IP from DHCP server for 20 seconds, then static IPv4, 192.168.3.127 is assigned. *Note: This process is achieved by setting profile-1 of LAN1 to WAN type with IPv4 DHCP, and profile-2 to IPv4 link-local. If profile-1 fails to obtain an IPv4 address from the DHCP server, it will automatically switch to profile-2.*
LAN2	No	Static IPv4, 192.168.4.127 retrieve from **/etc/network/interfaces**
Cellular/ Wi-Fi	No	Not configured

To run the MCM tool, use **# sudo mx-connect-mgmt**

```
MOXA Connection Management Command-line Utility

USAGE:
    mx-connect-mgmt [SUBCOMMAND]

FLAGS:
    -h, --help      Prints help information
    -V, --version   Prints version information

SUBCOMMANDS:
    GPS             Control GPS interface
    configure       MOXA Connection Management via GUI dialog
    datausage       Show interface data usage information and related functions
    default         Reset to default configuration
    debug           and diagnose cellular connection
    help            Show the help menu
    ls              List available network interfaces
    modem           Upgrade cellular modem firmware
    nwk_status      Show network and modem's information and connection status
    reload          configuration files and restart interfaces
    start           to control interfaces
    stop            to control interfaces
    unlock_pin      Unlock SIM PIN for the specified interface
    unlock_puk      Unlock PUK and reset SIM PIN for the specified interface
    wifi            Search Wi-Fi AP
```

NOTE

By default, only LAN1 port is managed by MCM.

There are 2 types of configuration files for MCM. One is main configuration file to manage the interrelationship between each interface, and one configuration files per each network interfaces available on Moxa Arm-based computer

Config Type	Description	File Location
Main Config.	Main configuration file which is to configure which network interface you would like MCM to manage and set the priority during failover/failback	/etc/moxa/MoxaConnectionManager/ **MoxaConnectionManager.conf**
Interface Config.	Per interface configuration file which is to configure properties of individual interfaces. Such as APN, PIN code of cellular connection or SSID and password of Wi-Fi.	/etc/moxa/MoxaConnectionManager /interfaces/ **[interface name].conf**

NOTE

- When modification is made to configuration file, you must use **# sudo mx-connect-mgmt reload** to make the change effective.
- You can find the detailed configuration file structure in the "Configuration File" chapter of the Moxa Connection Manager Reference Manual.
- We highly recommend using the GUI Configurator, described in the next section, instead of editing the configuration file directly, as it automatically checks for conflicts.

Instead of modifying the configuration file directly, we highly recommend you use the **GUI Configurator** described in next section to configure MCM.

Using MCM With CLI

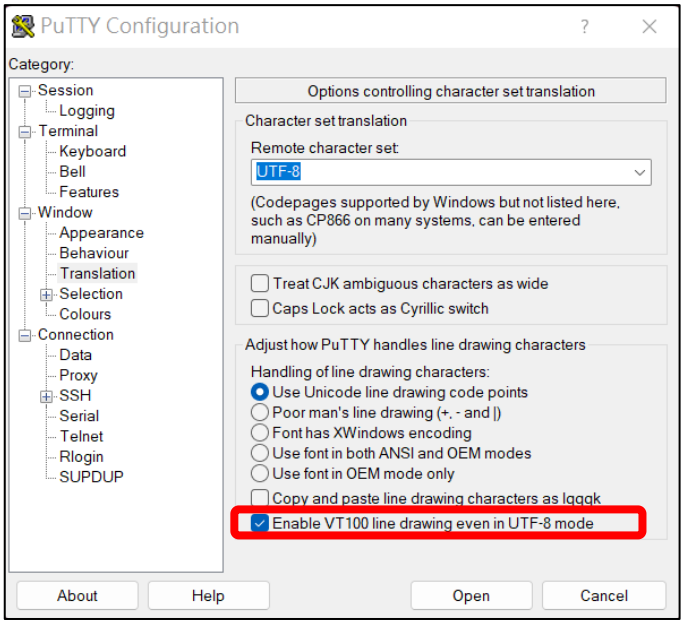
Like the **cell_mgmt** tool in Moxa Industrial Linux 1, the **Moxa Connection Manager (MCM)** also supports configuration through the command-line interface (CLI). For detailed, refer to the MCM CLI Reference Manual

Setting Up MCM with GUI Configurator

GUI Configurator Overview

To configure the WAN network through ethernet, Wi-Fi or cellular interface on the V2406C computer, you can use the simple GUI dialog provided by using **`# mx-connect-mgmt configure`** command.

If you are using PuTTY, enable **VT100 line drawing** option under **Windows > Translation** for the GUI to show correctly



1. Go to the main page.

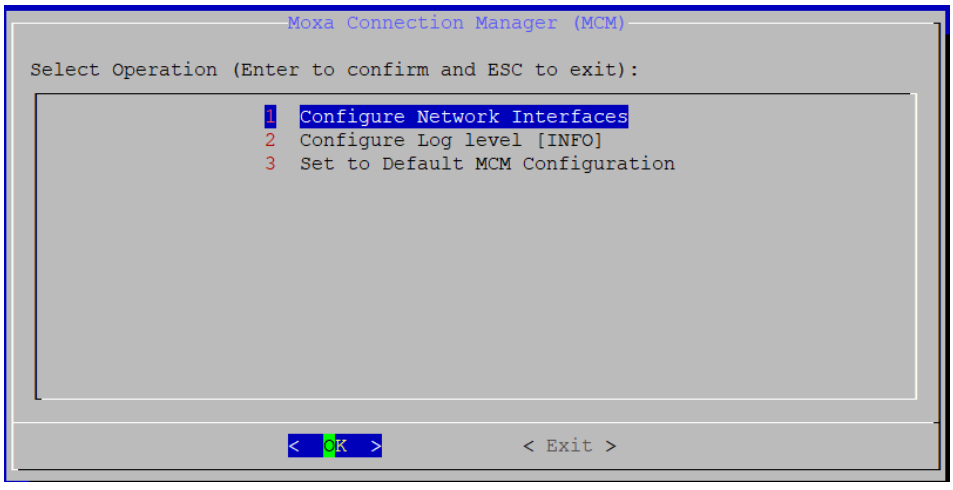


Figure 5.1 – Main page

Option Name	Description
Configure Network Interface	Configure network setting for each network interface
Configure Log Level	• Available syslog levels are ERR, WARN, INFO, DEBUG, TRACE • MCM log is save in /var/log/syslog
Set to Default MCM Configuration	Set all configuration to default

2. Configure network type for each interface and set the WAN connection priority for failover/failback.

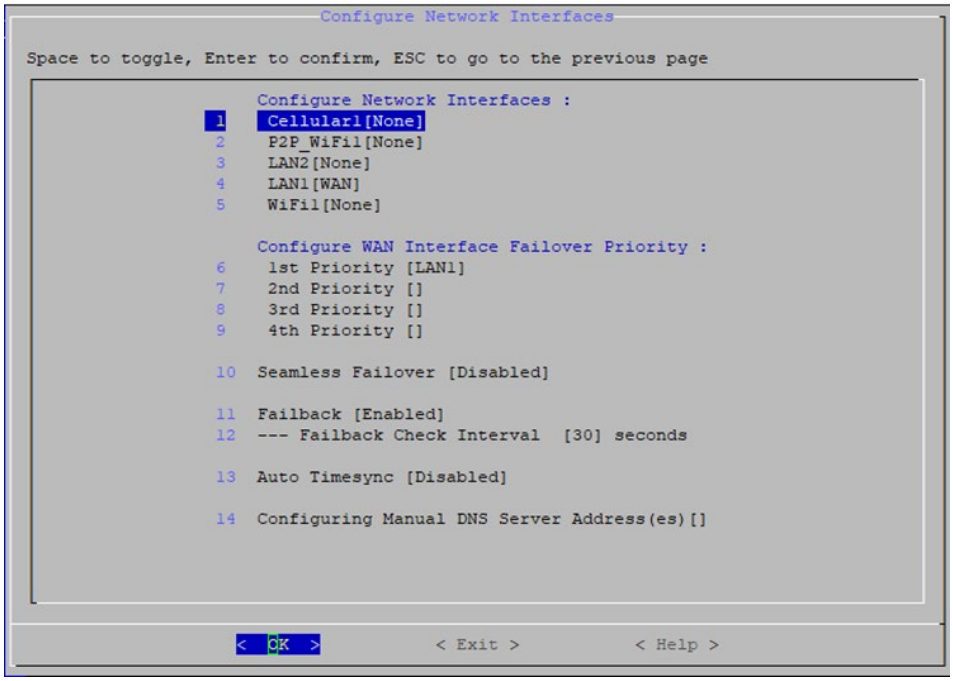


Figure 5.2 –Configure network interface

Option Name	Description
Configure Network Interfaces	A list of available network interfaces will show, where you can set the network type for each interface. The options are: • **WAN** - When set to WAN, this interface will be added to the default gateway list and allow MCM to apply automatic keep-alive and failover/failback control over it • **LAN** - When set to LAN, MCM will connect this interface using the network attributes defined in Profile-1 and DHCP server can be enabled for this interface • **LAN Bridge** - Bridge two or more LAN interfaces to construct a larger LAN • **Manual** - When set to Manual, it allows the user to have total control over this interface. MCM will connect this interface one-time only network attributes defined in Profile-1. MCM will not set these interfaces as the default gateway nor apply connection keep-alive and failover/failback control over it. • **Multi-WAN** - When set to Multi-WAN, it routes traffic to the interface from which it originates. • **None** - MCM will not manage this interface
Seamless Failover	• **Disabled (default)** - If the primary connection fails, MCM tries all pre-configured profiles before switching to the backup interface, causing some downtime during failback. • **Enabled** - If the primary connection fails, MCM will not attempt to try all the profiles configured for the primary connection. MCM will immediately switches to the connected backup interface, avoiding downtime. *Note: Using ping for the backup's keep-alive may incur data costs.*
Configure WAN Interface Priority	MCM will use the WAN interface set as 1st Priority as the default gateway. When the 1st priority interface becomes unavailable, MCM will automatically failover to the next priority interface.

Option Name	Description
Enable/Disable Failback	• When enabled, the backup connection will automatically failback to the higher priority connection when it became available again • **Failback Check Interval -** This value specifies how long (in seconds) the higher priority connection must remain stable before MCM triggers a failback, preventing frequent failover and failback due to instability
Auto Timesync	• **Disabled (default)** - Disables the auto time-sync function. • **GPS** - Syncs the system clock using GPS time. Requires a GPS antenna and the GPS function to be enabled. • **Chrony** - Uses the Chrony service to sync the system clock via an NTP server. • **Cellular** - Syncs the system clock using the cellular base station's time. A cellular connection is required.
Configure Manual DNS server Address(es)	This function allows you to manually specify DNS server addresses for the MCM to use for domain name resolution. If the DHCP server does not provide a DNS server, setting manual DNS addresses ensures that your system can still resolve domain names.

3. Configure individual network interface.

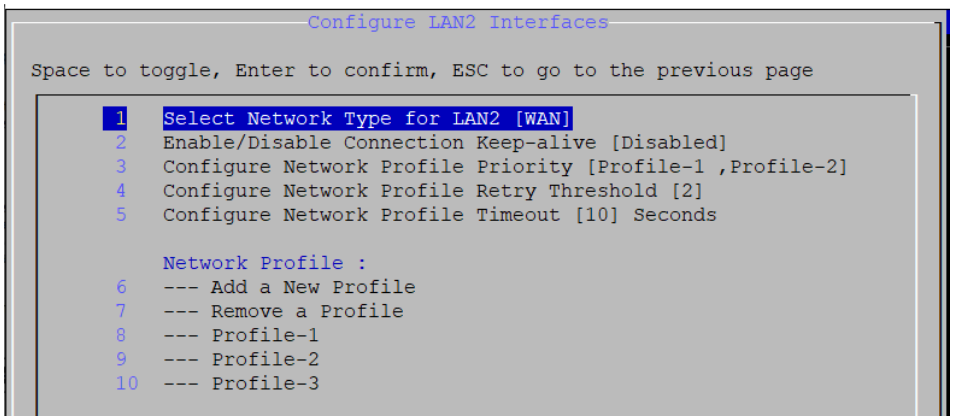


Figure 5.3 –Configurable options for WAN interface

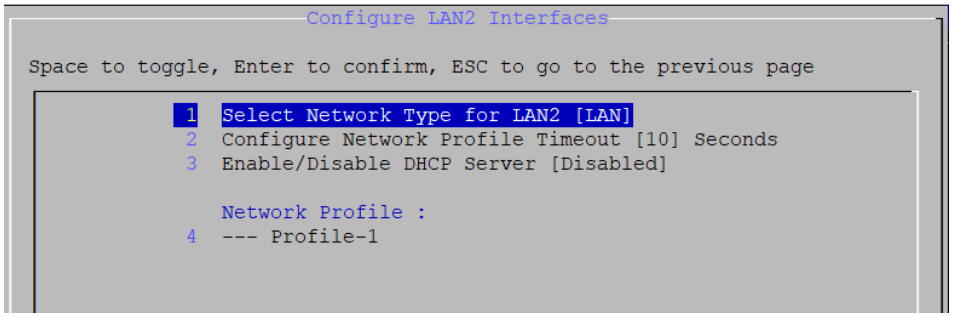


Figure 5.4 –Configurable options for LAN interface

Option Name	Network Type	Description
Select Network Type	All	Available options are WAN/LAN/LAN Bridge/Manual/Multi-WAN/None
Configure Network Profile Priority	WAN	When the 1st priority WAN network's profile cannot connect or becomes unavailable, MCM will automatically failover to the next profile in this priority list *Note: network profile failback is currently not supported*
Configure Network Profile Retry Threshold	WAN	This value determines the maximum attempts MCM will try to connect using the current WAN network profile before failover to the next profile in the priority list.
Configure Network Profile Timeout	All	This value (in seconds) determines the maximum time MCM will try to connect using the current network profile before determining the connection is unavailable
Bridge IPv4 Address	LAN-bridge	Assign a static IPv4 address for the bridged LAN interfaces
Bridge IPv4 Subnet Mask	LAN-bridge	Assign a static IPv4 subnet mask for the bridged LAN interfaces

Option Name	Network Type	Description
Enable/Disable DHCP Server	LAN, LAN-bridge	Configure a specific LAN or bridged LAN interfaces as DHCP server
Network Profile	WAN, LAN, Manual	• This section displays all network profile in a list with option to add, modify or remove a profile. • If network type is set to LAN or Manual, only profile-1 will be used because network profile failover is only available for WAN

4. Configure network profile of an interface.

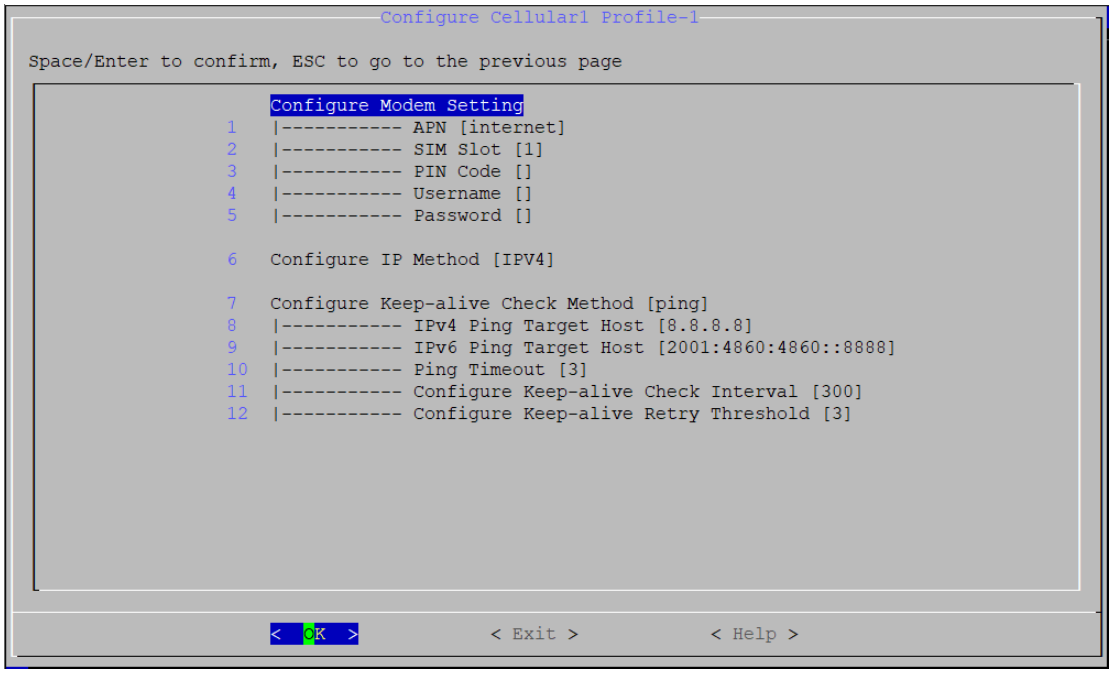


Figure 5.5 –Network profile setting (cellular interface as an example)

Option Name	Interface	Description
Configure Modem Setting	Cellular (WAN)	Configure cellular connection parameters including **APN, SIM slot** (which SIM slot number to use), **PIN Code, Username, Password**
	Wi-Fi (WAN)	Configure Wi-Fi connection parameters including **Mode** (only Wi-Fi client mode is supported), **SSID**, and **Password** *Note: make sure to leave the password field empty if you are connecting to a public Wi-Fi without password*
Configure IP Method	All interfaces	Configure IP related parameters including protocol version (IPv4, IPv6 or IPv4v6) and IP assignment method (DHCP, auto*, static IP or Link-local)
Configure Keep-alive Check Method	All interfaces	Select the method to check connection is alive • **Ping:** Connection is only considered alive if pinging the target server specified is successful ➢ Optionally, select "ping-signalmonitor" to also include signal strength as a criterion for a healthy connection. • **Check-ip-exist:** As long as an IP is assigned to the interface (e.g., the base station assigns IP to the cellular modem or DHCP server assigns IP to LAN port), are considered connection is alive ➢ Optionally, select "check-ip-exist-signalmonitor" to also include signal strength as a criterion for a healthy connection.

* IP assignment method “auto” is for IPv6 only, which support Stateless Address Auto-Configuration (SLACC) and Stateless for DHCPv6.

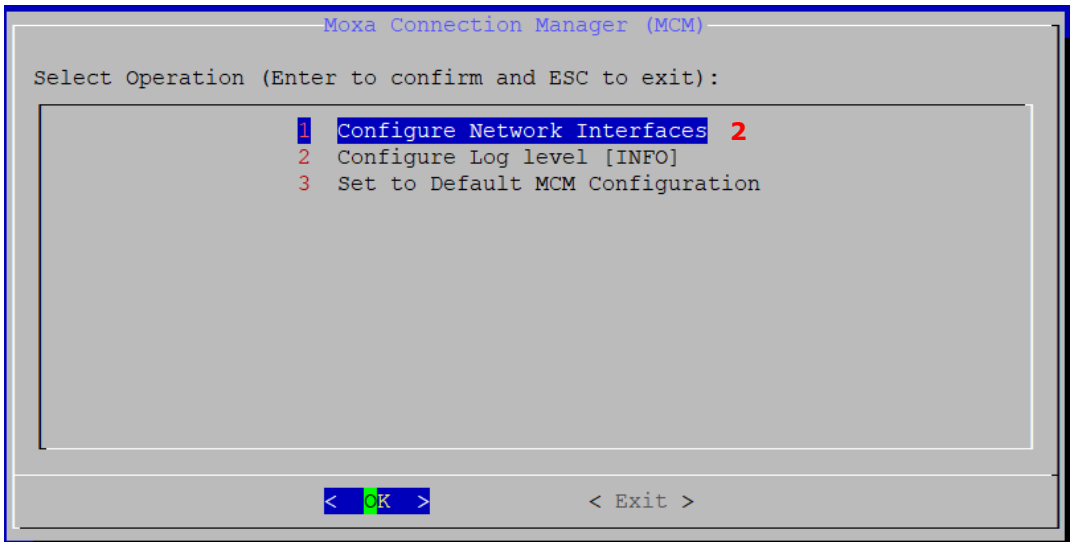
Cellular and Wi-Fi Failover/Failback

One of the key features in MCM is WAN connection auto-failover, where you can configure multiple backup WAN networks. When the primary connection becomes unavailable, MCM will automatically fail over to the backup network depending on the priority you set. You can even configure the connection to fall back to the primary one when it is back online.

In below example, we will set Wi-Fi interface as the primary WAN network and Cellular(4G/LTE) as the backup. MCM will automatically switch to using Cellular(4G/LTE) when Wi-Fi is down and back to Wi-Fi when it is back online.

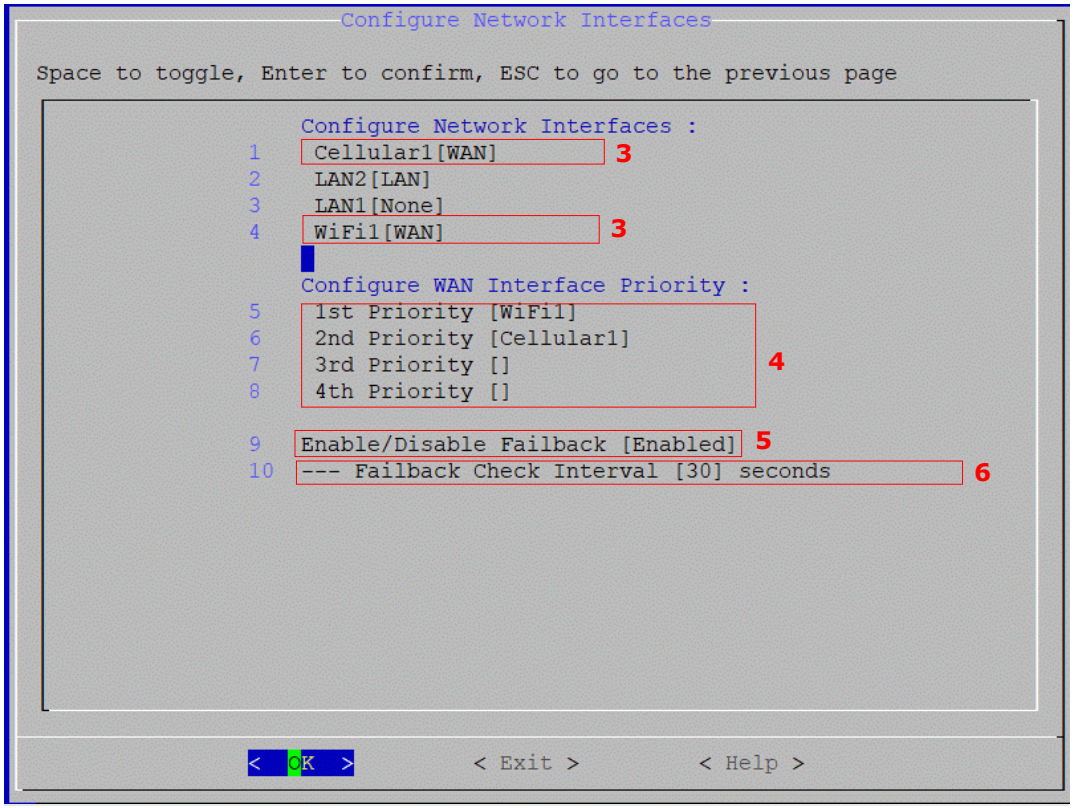
1. Run **# mx-connect-mgmt configure** to launch a simple GUI dialog configurator

```
root@moxa-tbbbb1182827:/# sudo mx-connect-mgmt configure
```



2. Select "Configure Network Interfaces"
3. Set interface Cellular1 and WiFi1 both to WAN, and
4. Set WiFi1 as the 1st priority and Cellular1 as 2nd priority
5. Make sure Failback is enabled if you would like MCM to automatically switch back to Wi-Fi from cellular when it is back online.

6. Failback Check Interval [30] seconds mean MCM will make sure Wi-Fi connection is alive and stable for 30 seconds before failback to use Wi-Fi as the primary connection (default gateway). The purpose is to avoid unstable connections causing frequent failover and failback.



7. Go to the interface configuration page of WiFi1 and Cellular1 (Figure 5.5 is an example of Cellular)
8. The option "**Enable/Disable Connection Keep-alive**" is disabled by default. It means there will be a short period without network during Wi-Fi to cellular failover process since MCM will only initiate the cellular connection when failover is triggered.

   You can enable this setting if a seamless failover experience is desired. When enabled, it allows MCM to failover to a ready-to-use backup connection without the initialization downtime.

NOTE

Enable/Disable Connection Keep-alive setting in this page has been replaced by "Seamless Failover" configuration in the main page since MCM v1.3.x, see Figure 5.2 –Configure network interface

9. MCM also supports network profile failover. For example, on a Moxa Arm-based computer with dual SIM slots, you can set up two profiles for cellular interface; each uses a different SIM slot and SIM card.
   - **Network Profile Priority:** in this example, MCM will use profile-1 by default and failover to use profile-2 when it cannot establish a connection with profile-1.
   - **Network Profile Timeout and Retry Threshold:** in this example, MCM will try to connect with profile-1 two times, each with a maximum of 90 seconds timeout before switching to profile-2.

10. You can modify the default profile-1 and profile-2 or add/remove a profile.

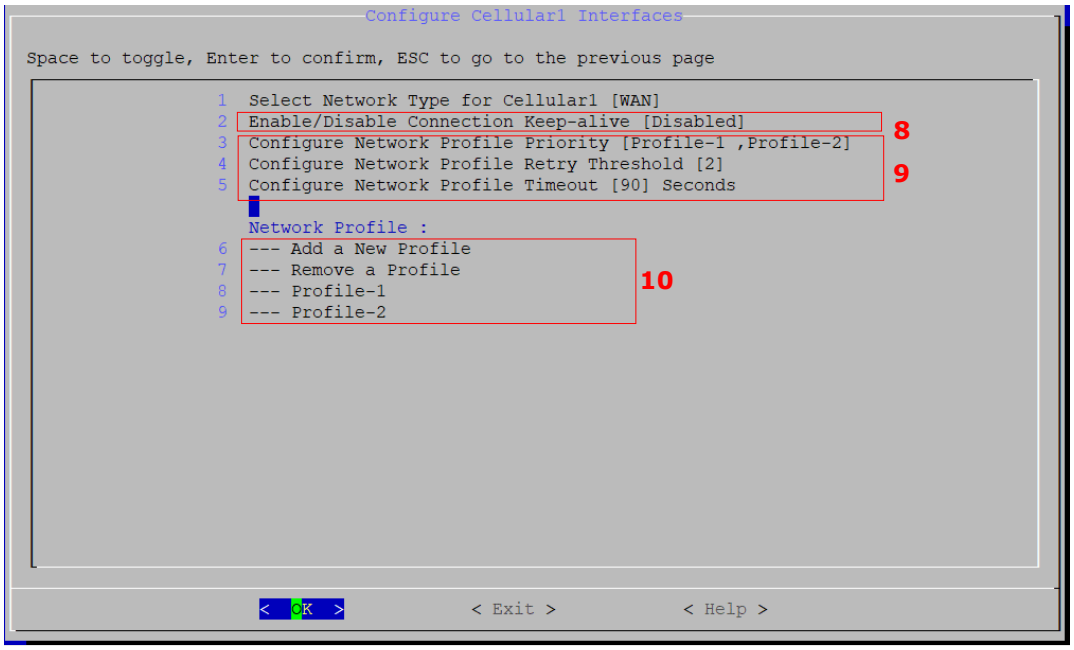


Figure 5.6 –Interface configuration page of Cellular1

11. Go to profile configuration page.
12. Configure the cellular modem related attribute. In this example, a SIM card in SIM slot 1 with PIN code "0000" and APN "internet" is used for Profile-1
13. Select the IP protocol generation. IPv4, IPv6, and IPv4v6 are the available options.
14. Select how MCM determine the connection is alive. Currently, only "ping' method is supported for WAN network. In this example, following configuration are set for Profile-1 of Cellular1 interface
    - MCM will ping the Google DNS once every 300 seconds.
    - MCM will try to ping the target host maximum 3 times (Retry Threshold) before concluding profile-1 cannot connect. For each ping attempt, MCM will consider ping fails if server doesn't response in 3 seconds (Ping timeout).
15. Once completed the configuration, exit MCM and select save and reload configuration file for the configuration to take effect

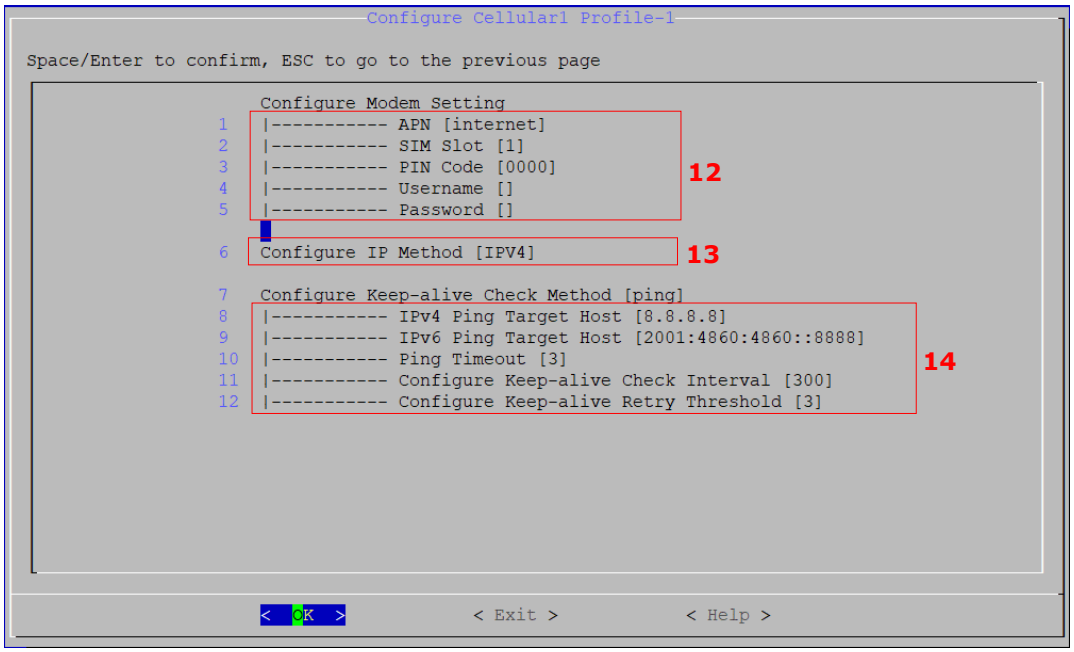
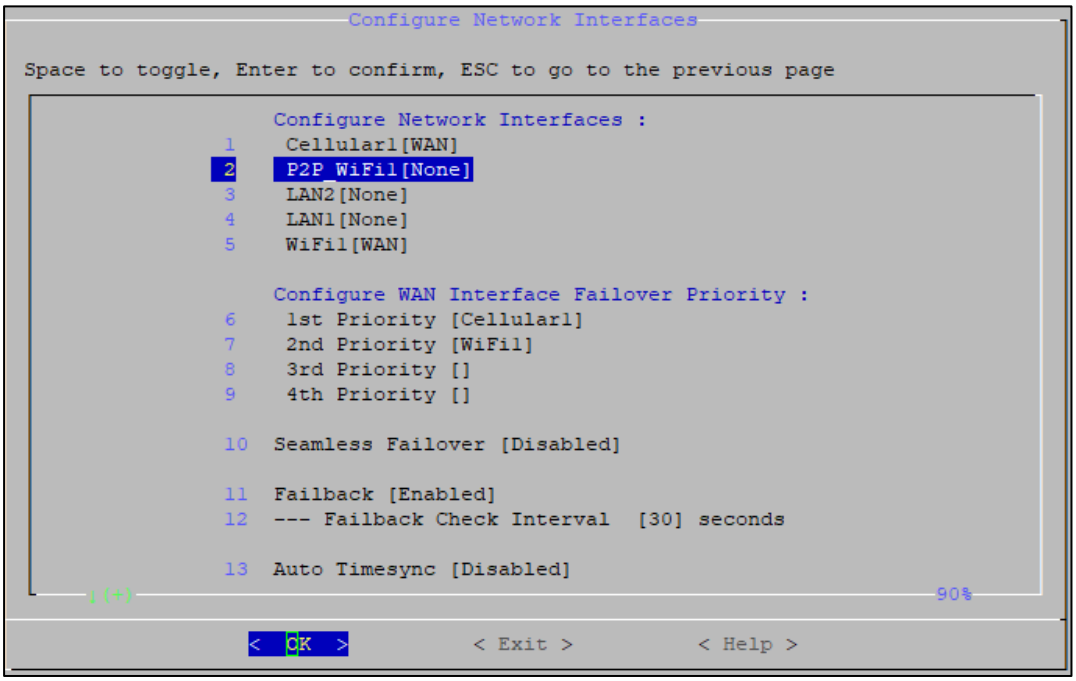


Figure 5.7–network profile configuration page of Cellular1 interface

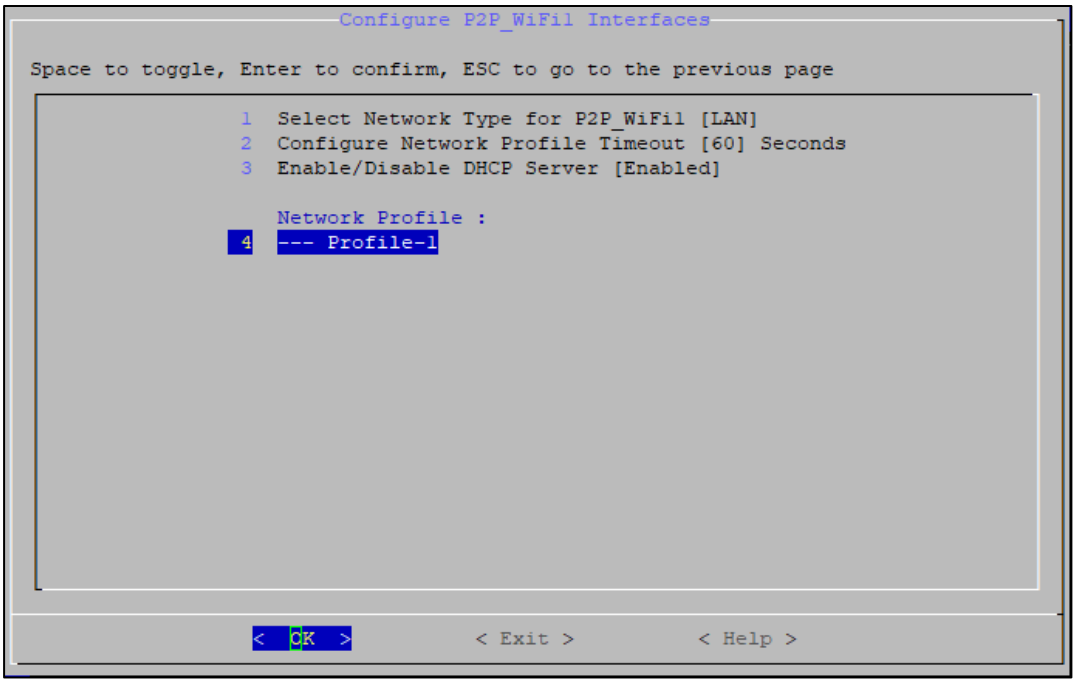
Connecting via Wi-Fi P2P for Remote Access

Starting from MIL 3.2, MCM includes a feature that allows remote access to Moxa computers via WiFi P2P. This is useful for remote debugging when a cellular connection is unavailable, and the device is in a difficult-to-access location without wired connections. WiFi P2P can be enabled alongside WiFi client mode, allowing simultaneous peer-to-peer communication and internet access through a WiFi network, providing flexibility in maintaining connectivity while troubleshooting.

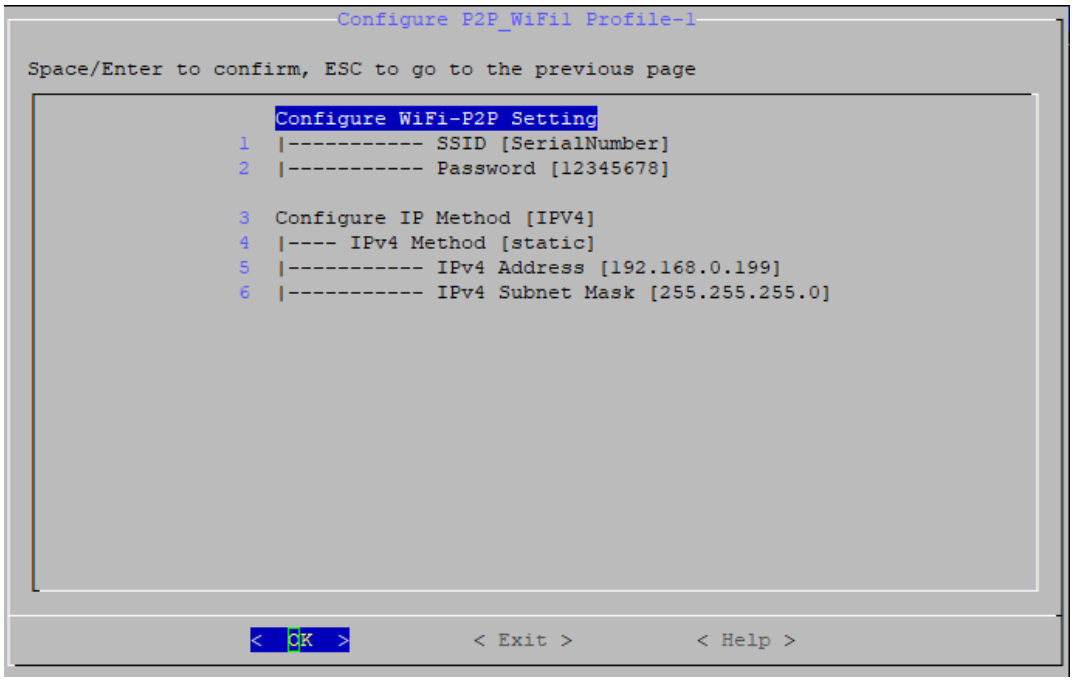
1. If your Moxa computer has a supported WiFi module installed, P2P WiFi will show up as an interface



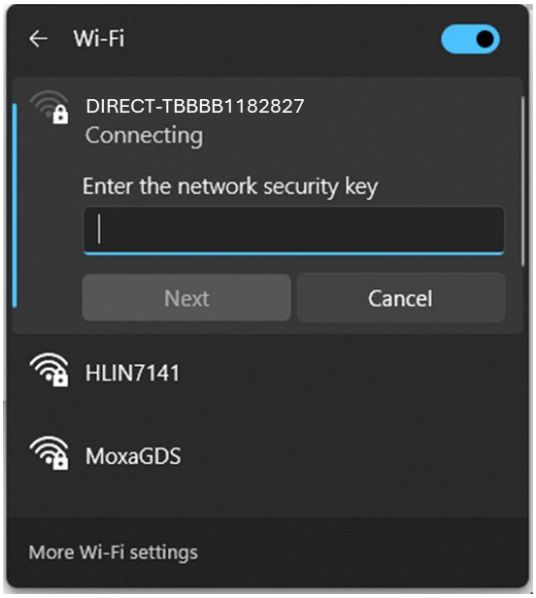
2. Enable P2P WiFi interface and configure the profile



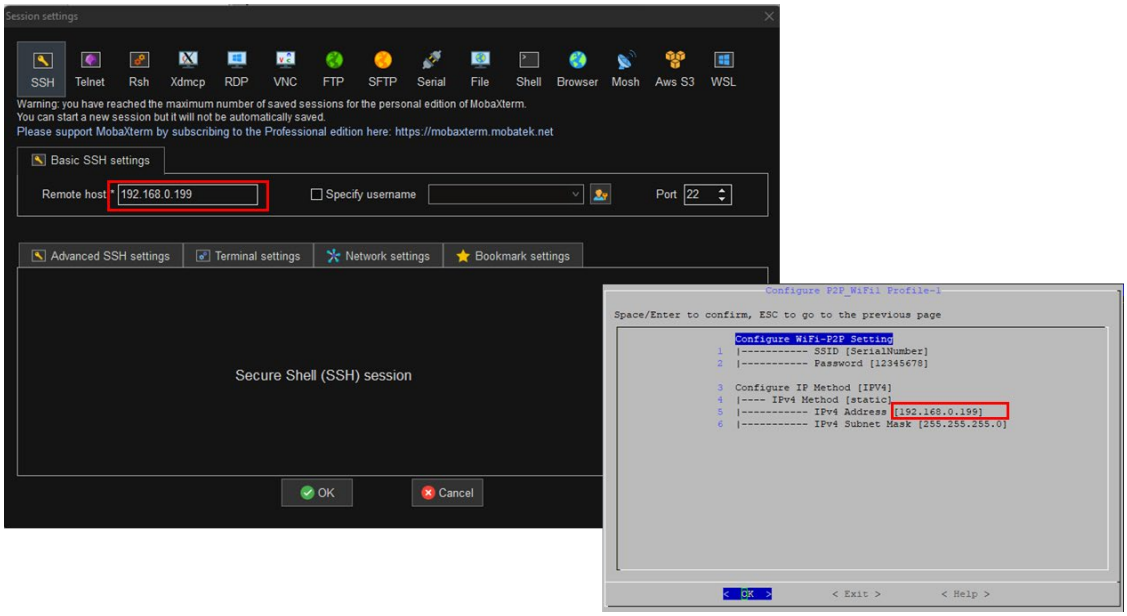
3. Configure the SSID and Password. The default SSID is DIRECT-[Moxa computer's serial number]



4. On another device with Wi-Fi, search for the configured Wi-Fi SSID and enter the password.



5. You can now remotely access the Moxa computer via SSH using the static IPv4 address set in the P2P WiFi profile.



Software Wi-Fi AP for Remote Access

Starting with Moxa Industrial Linux (MIL) 3.4, the Moxa Connectivity Module (MCM) introduces a software Wi-Fi Access Point (AP), enabling local Wi-Fi clients to connect to Moxa computers. This function facilitates seamless connectivity for up to three Wi-Fi clients within the same local network, serving as a lightweight hub for non-critical data exchange.

NOTE

The software Wi-Fi AP does not provide WAN (Internet) connectivity.

It operates only as a local network for secure, isolated communication between Wi-Fi clients and the device.

- **Applicable Products:** UC-2200A, UC-4400A, and UC-8200A Series.
- **Client Limit:** The software Wi-Fi AP supports up to 3 simultaneous clients to ensure reliable performance on resource-constrained IIoT gateways.
- **Non-critical Data Focus:** The feature is optimized for troubleshooting and non-critical data exchange (e.g., logs, configuration data) and is not intended for latency-sensitive or high-bandwidth applications, such as real-time control or critical data transmission.
- **No WAN Bridging:** The software Wi-Fi AP operates in a local network mode and does not provide internet or WAN access, enhancing security by isolating local communications from external networks.
- Either Software Wi-Fi AP or WiFi P2P can be enabled.

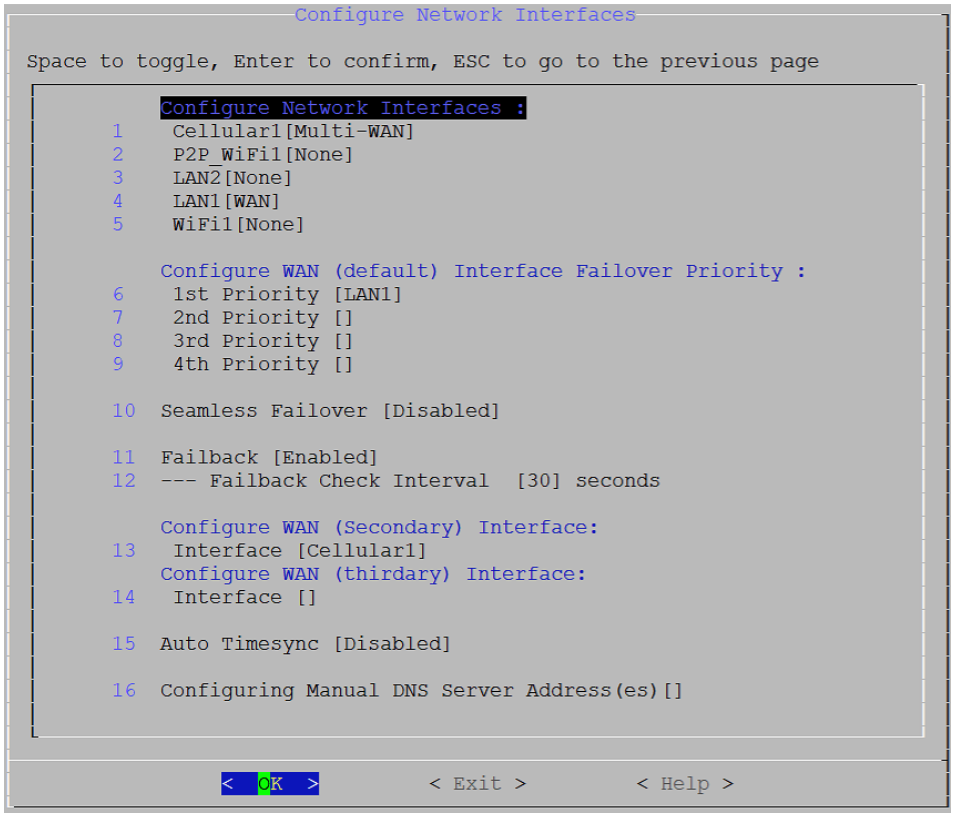
Target Use Case Scenarios

- **Field Diagnostics and Maintenance:** Technicians can use a laptop or mobile device to connect to Moxa computer's Wi-Fi AP for real-time monitoring, log retrieval, or firmware updates in remote or hazardous locations, such as oil rigs or wind turbines, where wired access is impractical.
- **Temporary Device Integration:** During commissioning or testing, the software Wi-Fi AP allows temporary connection of sensors or devices (e.g., temperature or pressure sensors) to Moxa computers for data collection, enabling rapid prototyping or proof-of-concept deployments without modifying existing network infrastructure.
- **Local Data Aggregation:** In small-scale IIoT deployments, such as a factory floor or a smart warehouse, the Wi-Fi AP enables devices like handheld scanners or IoT sensors to share non-critical data (e.g., inventory updates or environmental readings) within the local network, streamlining operations without relying on external networks.
- **Training and Demonstration:** The Wi-Fi AP can be used to showcase Moxa computer's functionality in training sessions or customer demos, allowing multiple devices to connect and interact with the gateway in a controlled, local environment.

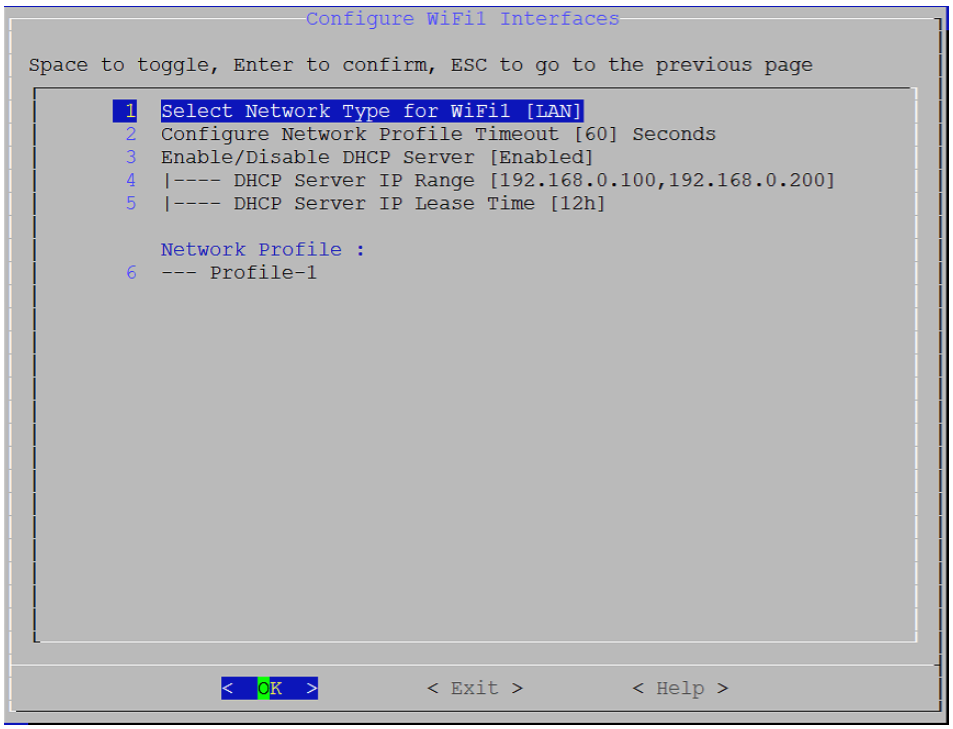
Setting Up a Software Wi-Fi AP

1. Identify the software Wi-FI interface to configure.

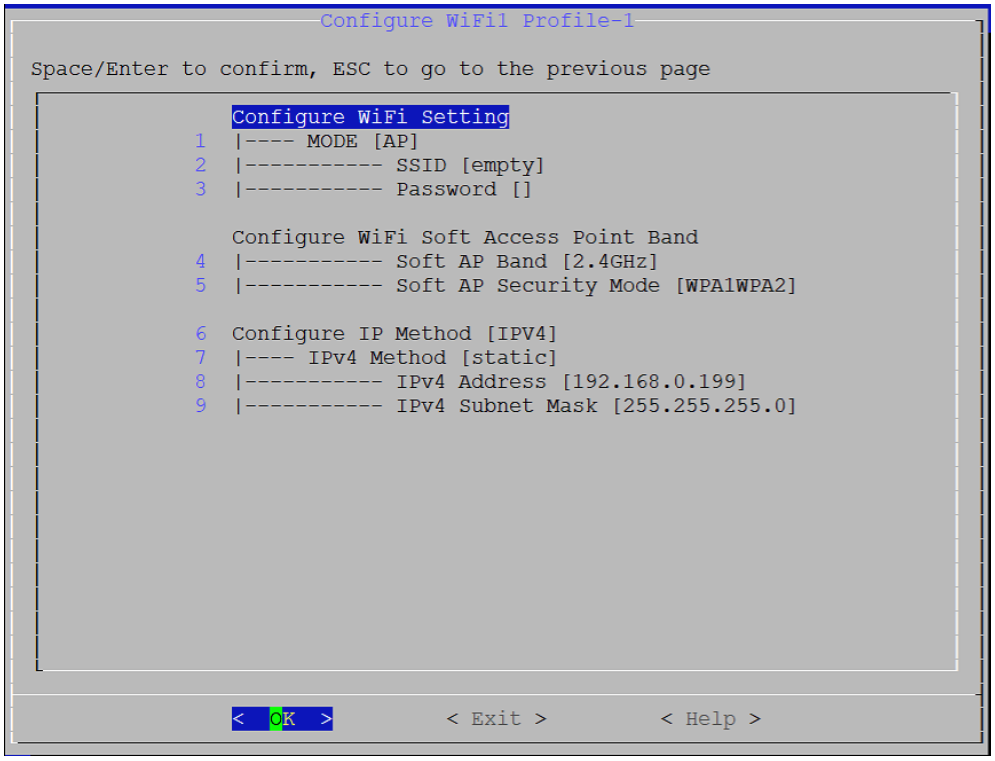
   If your Moxa computer has a supported Wi-Fi module installed, the software Wi-Fi AP will show up as the LAN interface **WiFi [LAN].**



2. Enable **WiFi1 [LAN]** interface and configure the profile



3. Configure the SSID, Password and an IP address for the connected Wi-Fi clients.



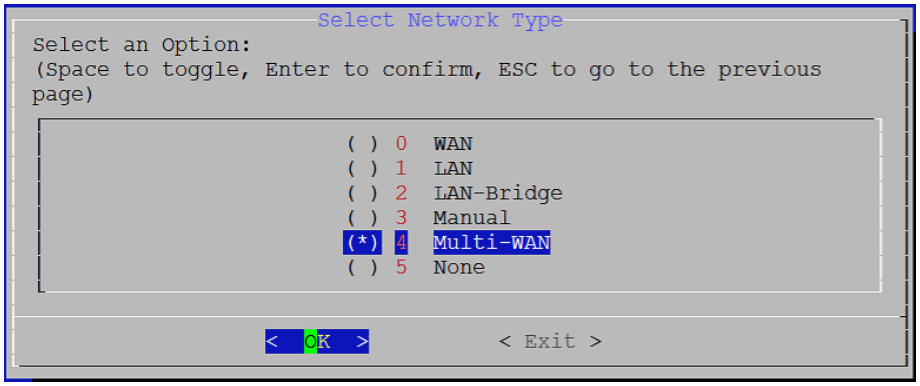
You can now remotely access the Moxa computer via SSH using the static IPv4 address set in the WiFi1 profile or access another Wi-Fi client that also connects to the Moxa computer.

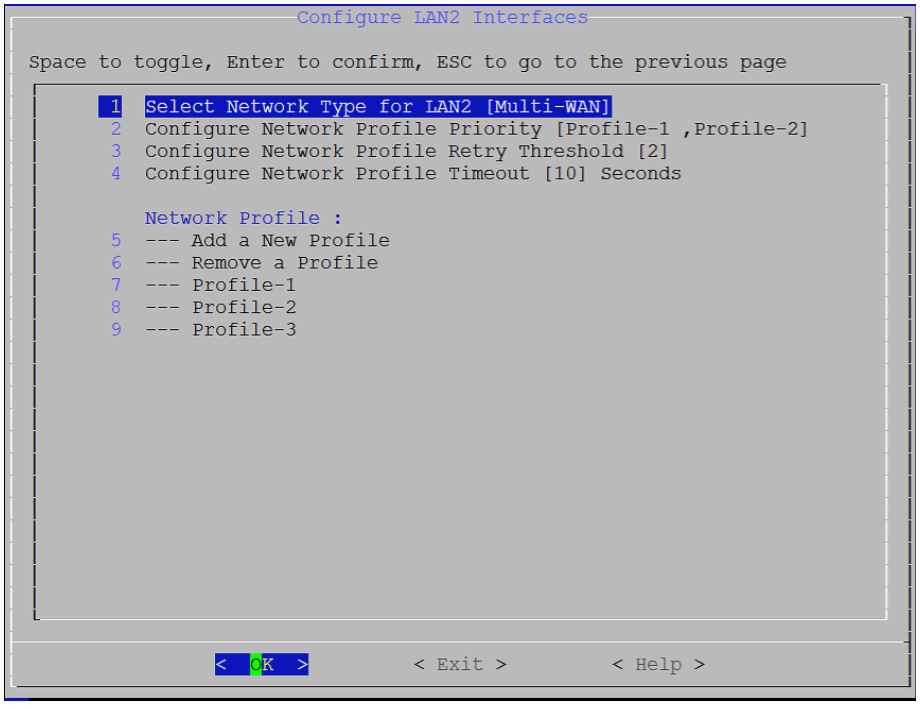
Setting Up a Multi-WAN Interface on LAN

Moxa computers support multiple WAN options, allowing the configuration of several interfaces as WANs. The Multi-WAN function intelligently routes traffic to the interface from which it originates, ensuring efficient and secure data flow in distributed IIoT environments. The function enhances network reliability by leveraging diverse connectivity options (e.g., Ethernet and cellular) to prevent single points of failure, which is critical for applications like remote monitoring and industrial automation. By maintaining traffic origin integrity, the function simplifies policy routing and optimizes bandwidth usage without requiring complex configurations.

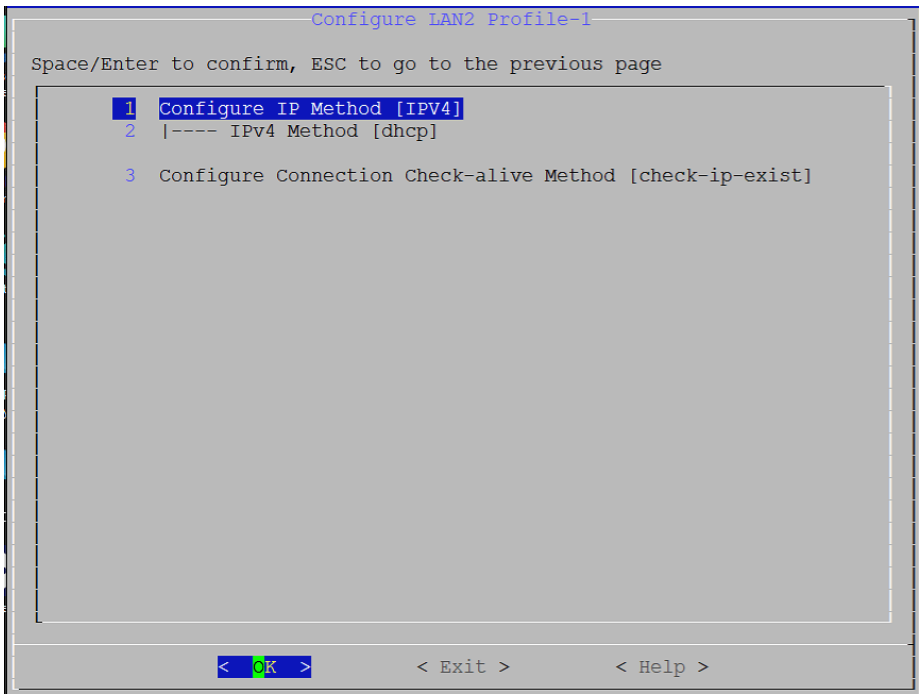
The following configuration example sets up LAN2 as a Multi-WAN interface:

1. Enable **LAN2 [None]** interface and configure the profile.





2. Configure IP Method and Check-alive Method.



Checking the Network Status

Checking the Interface and Connection Status

- Use **# sudo mx-connect-mgmt nwk_info [Interface name]** to check the interface and connection status
- Use **# sudo mx-connect-mgmt nwk_info -a [Interface name]**

```
------------------------------------------------
Interface Name          : Cellular1
Enabled                 : true
WAN Priority            : 1
Device Name             : cdc-wdm0
Device Type             : Modem
Network Ifname          : wwan0
Network Type            : WAN
Mac Address             :
IPv4 Method             : dhcp
IPv6 Method             :
------------------------------------------------
Modem State             : Connected
------------------------------------------------
Radio Access Tech       : LTE
Signal Strength         : Poor
Operator Name           : Chunghwa Telecom
Unlock Retries          : SIM PIN(3)
SIM Slot                : 2
IMSI                    : 466924253357038
APN                     : internet(Auto)
ICCID                   : 89886920042533570383
Cell ID/TAC             : 01C10722/2EE0
LTE RSRP                : -94 dBm
LTE RSSNR               : 0 db
Modem Version           : 25.30.626  1  [Jun 07 2021 06:00:00]
Modem Name              : Telit LE910C4-WWXD 1.00
IMEI                    : 353338974279918
------------------------------------------------
Connection Status       : Connected
Default Route           : true
------------------------------------------------
IPv4 | Address          : 25.8.139.168
     | Netmask          : 255.255.255.240
     | Gateway          : 25.8.139.169
     | Primary DNS      : 168.95.1.1
     | Secondary DNS    : 168.95.192.1
------------------------------------------------
IPv6 | Address          :
     | Netmask          :
     | Gateway          :
     | Primary DNS      :
     | Secondary DNS    :
```

Figure 5.8 –an example of nwk_info result of interface Cellular1

Most of the data fields and values are self-explanatory. Below are additional details to some of the data fields:

Fields	Description	Available Interface
Enabled	• **True:** This interface is managed by MCM • **False:** This interface is not managed by MCM	Wi-Fi, Ethernet, Cellular
WAN priority	The WAN priority set in Figure 5.2	Wi-Fi, Ethernet, Cellular
Network Type	WAN/LAN/Manual/None according to the set value in Figure 5.2	Wi-Fi, Ethernet, Cellular
Modem State	• **Not Ready:** The cellular modem can't be detected, or some configuration is not set correctly in MCM configuration files. • **Initializing:** The cellular is initializing • **SIM PIN Locked:** SIM PIN is locked; you can unlock with unlock_pin command • **SIM PUK Locked:** SIM PUK is locked; you can unlock with unlock_puk command • **Radio Power Off:** The cellular modem is entering flight mode • **Radio Power On:** The cellular modem is exiting flight mode • **Searching Base Station:** The cellular modem has exited flight mode and searching for base-station • **Attached to Base Station:** The cellular modem is registered with a network provider but without data connections. • **Connecting:** The cellular modem is connecting • **Connected:** The cellular modem is connected • **No SIM:** SIM card is missing or malfunctioning	Cellular only
Radio Access Tech	GSM/GSM COMPACT/UMTS/LTE/5G SA/5G NSA, etc.	Cellular only
Signal Strength	• None/Very Poor • Poor • Fair • Good • Excellent *Note: see cellular signal strength for defined criteria*	Cellular only
SIM Slot	The SIM slot number being used	Cellular only
Connection Status	• **Initializing:** Initializing network connection • **Device Ready:** Detected the network interface is ready • **Connecting:** Connecting according to setting in profile • **Configuration Error:** Profile configuration error • **Disabling:** Stopping the connection • **Disabled:** When an interface is not managed by MCM, or MCM service is stopped • **Connected:** Connection is "working". The criteria for "working" are determine by the Keep-alive Check Method in Figure 5.5. For example, if method is set to ping, the connection is consider working if ping is successful • **Unable to connect:** The network profile is set correctly but the connection is not working determined by the Keep-alive Check Method in Figure 5.5 • **Reconnecting:** Connection is being reconnecting	Wi-Fi, Ethernet, Cellular
Default Route	• **True:** This interface is currently being used as default route • **False:** This interface is not the default route	Wi-Fi, Ethernet, Cellular

Cellular Signal Strength

Signal Indicator

1. 3G Signal Indicators:
   - **RSSI** (Received Signal Strength Indicator): Measures the received signal strength in dBm.
   - **EC/IO** (Energy per Chip over Interference): Indicates the signal quality by measuring the ratio of the received energy per chip to the interference level, in dB.
2. 4G Signal Indicators:
   - **RSRP** (Reference Signal Received Power): Represents the power of the reference signal in dBm, used to assess the signal strength in LTE networks.
   - **RSSNR** (Reference Signal Signal-to-Noise Ratio): Measures the quality of the reference signal by evaluating the signal-to-noise ratio in dB.
3. 5G Signal Indicators:
   - **RSRP** (Reference Signal Received Power): Measures the power of the 5G reference signal when connected to a 5G cell, similar to SA.
   - **SINR** (Signal-to-Interference-plus-Noise Ratio): Reflects the quality of the 5G signal, considering the presence of 4G signals in the same environment.

Signal Level Criteria

Below are the criteria that MCM uses to determine the signal strength for 3G(UMTS), 4G(LTE), 5G SA and 5G NSA:

Using 4G(LTE) signal level as an example:

- For the signal level "Excellent", both RSRP and RSSNR need to meet the defined criteria in below table
- If the criteria for RSRP and RSSNR differ, the MCM will display the lower of the two signal levels. For example, if the RSRP value meets the "Excellent" criteria but EC/IO RSSNR meets only the "Good" criteria, then the MCM will show "Good" signal level

5G NSA Signal Level	LTE RSRP (dBm)	LTE RSSNR (db)	5G RSRP (dBm)	5G SINR (dBm)
Good	>= -90	>= 10	>= -80	>= 15
Fair	-105 <= x < -90	5 <= x < 10	-90 <= x < -80	5 <= x < 15
Poor	-125 <= x < -105	-20 <= x < 5	-110 <= x < -90	-20 <= x < 5
No Signal	< -125	<-20	< -110	<-20

5G SA Signal Level	RSRP (dBm)	SINR (db)
Good	>= -80	>= 15
Fair	-90 <= x < -80	5 <= x < 15
Poor	-110 <= x < -90	-20 <= x < 5
No Signal	< -110	<-20

4G(LTE) Signal Level	RSRP (dBm)	RSSNR (db)
Excellent	>=-85	>=13
Good	>=-95	>=5
Fair	>=-105	>=1
Poor	>=-115	>=-3
None/Very Poor	<-115	<-3

3G(UMTS) Signal Level	RSSI (dBm)	EC/IO (db)
Excellent	>=-77	>=-6
Good	>=-87	>=-10
Fair	>=-97	>=-14
Poor	>=-107	>=-20
None/Very Poor	<-107	<-20

Monitoring the Data Usage

Use **# mx-connect-mgmt datausage** to check the data usage of a specified interface between a specified start and end date

```
moxa@moxa-tbbbb1182827:# sudo mx-connect-mgmt datausage -h

mx-connect-mgmt-datausage
Show interface data usage information and related functions

USAGE:
    mx-connect-mgmt datausage [FLAGS] [OPTIONS] [interface]
FLAGS:
    -h, --help     Prints help information
    -r, --reset    data usage database
OPTIONS:
    -s, --since <date>    Sets the begin date of data usage cumulative period,
                            expected date format YYYY-MM-DD HH:MM or YYYY-MM-DD
    -t, --to <date>       Sets the end date of data usage cumulative period,
                            expected date format YYYY-MM-DD HH:MM or YYYY-MM-DD
ARGS:
    <interface>
```

Below is an example of how to check the data usage of Wi-Fi interface between 2022/7/3 and 2022/7/4

```
moxa@moxa-tbbbb1182827:# sudo mx-connect-mgmt datausage --since 2022-07-03 --to
2022-07-04 WiFi1
moxa@moxa-tbbbb1182827:
rx: 21884544 bytes
tx: 116086 bytes
```

Upgrading the Cellular Modem Firmware

Use **# mx-connect-mgmt modem upgrade [Interface name]** will check and install the latest cellular modem firmware tested by Moxa from Moxa APT server.

- Your cellular network will be down temporary during the upgrade and the connection will be reconnected by MCM after the upgrade is complete
- You can also upgrade the firmware locally by specifying a file path following **-F** or **--filepath** option
- By default, firmware downgrade is not allowed and not recommended. If you insist to downgrade the firmware, you can add -f flag to force the downgrade.
- You can use **mx-connect-mgmt nwk_info [interface name] -a** command to check the current cellular modem firmware version
- MCM will perform auto-reinstallation if upgrade fails.

```
moxa@moxa-tbbbb1182827:/# sudo mx-connect-mgmt modem upgrade -h
mx-connect-mgmt-modem-upgrade
Upgrade modem FWR

USAGE:
    mx-connect-mgmt modem upgrade [FLAGS] [OPTIONS] [interface]
FLAGS:
    -f              force upgrade FWR
    -h, --help    Prints help information
OPTIONS:
    -F, --filepath <filename>    Sets the FWR file path
ARGS:
    <interface>
```

An example of automatically updating the cellular modem firmware from Moxa APT server is given below:

```
moxa@moxa-tbbbb1182827:/# sudo mx-connect-mgmt modem upgrade Cellular1
```

An example of manually updating the cellular modem firmware by specifying a firmware file is given below:

```
moxa@moxa-tbbbb1182827:/# sudo mx-connect-mgmt modem upgrade Cellular1 -F /etc/
firmware/Telit-LE910C4-EU-Info-1.1.0
```

An example given below indicates how to manually force the cellular modem firmware update even if the current firmware is newer than the provided firmware:

```
moxa@moxa-tbbbb1182827:/# sudo mx-connect-mgmt modem upgrade Cellular1 -f -F
/etc/ firmware/Telit-LE910C4-EU-Info-1.0.0
```

Cellular Network Diagnosis

Use **# mx-connect-mgmt debug** to perform diagnosis on the cellular network if you have trouble getting it to connect. The diagnosis tool can identify common issues such has missing antenna, weak signal strength, SIM card pin code error, SIM locked, etc.

```
moxa@moxa-tbbbb1182827:# sudo mx-connect-mgmt debug -h
mx-connect-mgmt-debug
Debug and diagnose cellular connection

USAGE:
    mx-connect-mgmt debug [SUBCOMMAND]

FLAGS:
    -h, --help    Prints help information

SUBCOMMANDS:
    diag      Perform diagnosis on the cellular interface
    help      Prints this message or the help of the given subcommand(s)
    listen    Listen to properties changed
```

NOTE

Cellular network diagnosis is not available for 5G yet.

Using API to Retrieve the MCM Status

MCM provides C application programming interfaces (APIs) for developer to retrieve various network and interface status from MCM

Please refers to following link for the C API document

https://moxa.gitlab.io/open-source/linux/gitbook/moxa-connection-manager-reference-manual/MCM/Libmcm

To integrating your applications securely with the MC C API, you should follow the below guideline:

1. Confirm that the return value of the API is 0 and the returned struct pointer is not NULL to avoid using the wrong memory address.
2. Always free the structure pointer returned by the API to avoid memory leak.

How to Migrate From cell_mgmt to MCM

For instructions on migrating from **cell_mgmt** in MIL1 to **Moxa Connection Manager (MCM)** for cellular connection management, see cell_mgmt to MCM migration.

6. System Installation and Update

In this chapter, we will explain how to install and update **Moxa Industrial Linux (MIL)** and the **bootloader**. The MIL image file can be downloaded from the official product page for the Moxa computer series.

For example, the image for the **UC-3400A** can be found at:
https://www.moxa.com/en/products/industrial-computing/arm-based-computers/uc-3400a-series#resources

Full System Installation Using .img File

Using a TFTP Server From Bootloader Menu

Refers to instruction in Accessing Bootloader Menu section

> **NOTE**
>
> TFTP update is disabled in Secure model by default due to TFTP is not a secure transmission protocol.

Using a USB/SD From Bootloader Menu

Refers to instruction in Accessing Bootloader Menu section

Automatic Installation From a USB or SD

Beside manually installing the system image from bootloader menu, you can also trigger the image installation process within the operating system using **`sudo mx-bootloader-mgmt image_auto_install`** command. Once this process is triggered, the Arm-based computer will automatically install the specified system image in the USD or SD attached to the system. The new image will be available upon the next system boot-up.

> **NOTE**
>
> The format supported for USB and SD are FAT32 and ext4, respectively.

Command	Description
-d, --disk	Display the name of the external storage (e.g., USB, SD) where the image file is located. You can use the **`mx-interface-mgmt disk`** command to query the external storage name.
-f, --file	Specify the name of the image file in the external storage • You can also use an absolute path starting from MIL 3.2
-i, --info	Display the names of the image file and external storage configured for auto-install upon next boot-up
-r, --remove	Remove the auto-installation configuration
-h, --help	Display the available commands with a brief description
-v, --version	Display the version of **`mx-image-auto-install-tool`**

Following is an example of the automatic installation of the system image from a USB device:

1. Use **`mx-interface-mgmt disk`** command to check the name of available storage device name.

```
moxa@moxa-tbzkb1090918:~# sudo mx-interface-mgmt disk
NAME    DEVICE          SYSTEM_DISK  NUMBER_OF_PARTITIONS  AUTOMOUNT_SETTING
USB     /dev/sda        N              1                       false
eMMC    /dev/mmcblk0    Y              4                       false
```

2. Mount the USB if it is not already mounted. Refer to Storage and Partition section for detail.

```
moxa@moxa-tbzkb1090923:~$ sudo mx-interface-mgmt partition
NAME        EVICE           IS_MOUNTED  FS_TYPE  MOUNTPOINT
eMMC_p1   /dev/mmcblk0p1    Y           ext4     /boot_device/p1
eMMC_p2   /dev/mmcblk0p2    Y           ext4     /boot_device/p2
eMMC_p3   /dev/mmcblk0p3    Y           ext4     /boot_device/p3
eMMC_p4   /dev/mmcblk0p4    Y           ext4     /boot_device/p4
USB_p1    /dev/sdb1         N           N/A      N/A

moxa@moxa-tbzkb1090923:~$ sudo mx-interface-mgmt partition USB_p1 mount
```

3. Configure an auto-installation event in partition 1 of the USB device with the image file **IMG_UC-8200_MIL3_V1.0_Build_22053011_ImageBuild_220530_133813.img**:

```
moxa@moxa-tbzkb1090918:~# sudo mx-bootloader-mgmt image_auto_install -d USB
-f
IMG_UC-8200_MIL3_V1.0_Build_22053011_ImageBuild_220530_133813.img
```

> **NOTE**
>
> - Ensure that the image file and sha256/512 hash files are available in partition 1 of USB or SD before configuring the event.
> - For Secure models, the digital signature file (.sha256.bin.signed or .sha512.bin.signed) must also be placed alongside the image file.

4. Reboot the system to trigger the auto installation of the system image from the USB device.

```
moxa@moxa-tbzkb1090918:~# sudo reboot
```

Offline or Online Upgrade Using MSU

Moxa Software Updater (MSU) is a Moxa utility for performing both offline and online software upgrades to update the MIL version on Moxa computers. For offline upgrades, two types of upgrade packages are available: the **Upgrade Pack** and the **Refresh Upgrade Pack**.

- The **Upgrade Pack upgrades** the system while preserving user data and configurations. It contains only the differences between the current and target versions, making it significantly smaller in size.
- The **Refresh Upgrade Pack** performs a full system upgrade by wiping all user data and restoring the system to its factory default environment. This pack contains all the files from the target version and is, therefore, larger in size.

MSU is available starting from MIL 3.2. For all MIL versions after MIL 3.2 (e.g., MIL 3.3), Moxa will release software update packs that can be downloaded from Moxa Software Release Service (SRS).

Moxa Software Updater (MSU) includes built-in integrity and authenticity checks. Each upgrade pack is accompanied by a **SHA-512 hash file** and a **digitally signed version** of that hash (.sha512 and .sha512.bin.signed). During the upgrade process, the system automatically verifies the **SHA-512 hash** of the upgrade package to ensure file integrity. It then validates the **digital signature** using a trusted public key to confirm the authenticity of the package.

> **NOTE**
>
> If either the hash or signature verification fails, the upgrade process is immediately aborted to prevent tampering or unauthorized updates.

NOTE

The Moxa Industrial Linux (MIL) kernel and Moxa-developed packages are available pre-configured on the Moxa APT repository when you enable updates via APT. However, we first recommend using **`mx-sw-updater`** command to resolve package dependency issues and update the MIL3 kernel before the APT package update. If you want to exclusively upgrade from the APT repository, run the **`apt full-upgrade`** command to minimize dependency issues.

To use Moxa Software Updater (MSU), run **`# sudo mx-sw-updater [command]`**.

Command and Usage	Description
`configure [flags]`	The **`mx-sw-updater configure`** command sets up an offline upgrade pack (root required). It prepares the upgrade or refresh pack and verifies it with a signature file. This step ensures that the package and metadata are copied and configured to the device's local cache before upgrading. **Key Flags** • **-p, --path:** Specifies the path to the upgrade pack or refresh upgrade pack. • **-s, --signature:** Provides the path to the signature file for verification. If not specified, it will attempt to find a matching signature file in the same directory.
`clear`	Clears the package file and metadata copied to the device's local cache after the completing the upgrade.
`update`	The **`mx-sw-updater configure`** command fetches the latest metadata from the Moxa Apt Repository to the device's local cache. This command ensures that the system's package information is up-to-date and ready for installing or upgrading packages.
`upgrade[flags]`	The **`mx-sw-updater configure`** command updates the system to a target official version while preserving user data and configurations, with options to perform the upgrade using a local package or remote APT server with automatic recovery. This command requires root privileges. **Key Flags** • **-l, --latest:** Upgrade to the newest version in the local cache • **--remote:** Upgrade remotely via the APT server (default option). • **--local:** Upgrade using the local upgrade pack • **--system-failback:** Performs the upgrade with system failback enabled to ensure auto system recovery if the upgrade fails. • **-r, --release <string>:** Upgrades to a specified target version (e.g., -r V1.1).
`Refresh-upgrade [flags]`	The **`mx-sw-updater refresh-upgrade`** command upgrades the system by wiping all user data and restoring it to the factory default environment. Unlike the **`mx-sw-updater upgrade`** command, which preserves user data, the refresh-upgrade command resets the system to its original state. This command supports only local upgrades and requires root privileges. **Key Flags** • **-l, --latest:** Upgrade to the newest version in the local cache. • **--system-failback:** Performs the upgrade with failback enabled, ensuring automatic system recovery if the upgrade fails. • **-r, --release <string>:** Upgrade to a specified target version (e.g., -r V1.1).

Command and Usage	Description
`show [flags]`	The **`mx-sw-updater show`** command displays details about the upgrades that have been added to the device's local cache via the **`mx-sw-updater configure`** and **`mx-sw-updater update`** commands. The details include the version number, supported Moxa computer models, and the changelog. **Key Flags** • **-a, --all:** Shows information of all available upgrades. • **-l, --latest:** Displays information of the newest upgradable version. • **-r, --release <string>:** Shows details of a specified upgradable version (e.g., -r V1.1). • **--from <string>:** Specifies the starting version for a range. • **--to <string>:** Spec**ifies the ending version for a range.** *Note: The --from and --to flags can be used together to display information for a range of versions*
`status [flags]`	The **`mx-sw-updater status`** command provides information about the current status of upgrade packages that have been added to the device's local cache via the **`mx-sw-updater configure`** and **`mx-sw-updater update`** commands, allowing users to check the availability and progress of various software updates. **Key Flags** • **-a, --all:** Shows the status of all available upgrades. • **-l, --latest:** Displays the status of the newest upgradable version. • **-r, --release <string>:** Shows the status of a specified upgradable version (e.g., -r V1.1). • **--from <string>:** Specifies the starting version for a range. • **--to <string>:** Specifies the ending version for a range. *Note: The --from and --to flags can be used together to display the status for a range of versions*
`list [flags]`	The **`mx-sw-updater list`** command is used to display information about software packages, including installed packages, upgradable packages in the local cache, and differences between versions. **Key Flags** • **-l, --latest:** Lists all packages from the newest upgradable version in the local cache • **-s, --system:** Lists all packages currently installed on the system. • **-r, --release <string>:** Lists all packages from a specified upgradable version (e.g., -r V1.1). • **-c, --compare <stringArray>:** Show the changed packages between two specified versions (e.g., -c V1.0 -c V1.1). If the second version is not specified, it compares the specified version with the installed system packages. • --detailed: Shows both changed and unchanged packages. This flag is to be used with -c, --compare. • --no-fixed: Display output without fixed-length formatting.
`Verify [flags]`	The **`mx-sw-updater verify`** command is used to verify the integrity and authenticity of an upgrade pack or refresh-upgrade pack by checking its digital signature. This ensures that the upgrade package has not been tampered with and is valid before performing any system upgrades. **Key Flags** • **-p, --path <string>:** Specifies the path to the upgrade pack or the refresh upgrade pack. • **-s, --signature <string>:** Specifies the path to the signature file for verification. If not specified, the command will try to find the .sha512.bin.signed file in the same directory as the upgrade pack.

Offline Upgrade

An example of using **mx-sw-updater** to upgrade a UC-4434A-I-T computer from OS image v1.0/1.1 to v1.3 (MIL 3.4.1):

1. Download the v1.3 firmware package (ZIP format) for UC-4400A from the UC-4400A Product Site
2. Extract the ZIP file and locate the upgrade pack in the **Offline Upgrade Pack** folder.
3. Transfer both the upgrade pack and its digital signature file to the target device.
   - moxa-UC-4400A_MIL3_V1.1-upgrade-pack
   - moxa-UC-4400A_MIL3_V1.1-upgrade-pack.sha512.bin.signed

```
root@moxa-imoxa0920070:/home/moxa# ls -1
moxa-UC-4400A_MIL3_V1.3-upgrade-pack
moxa-UC-4400A_MIL3_V1.3-upgrade-pack.sha512.bin.signed
```

   If you intend to perform a full system upgrade that wipes all user data and restores the system to its factory default settings, use the **Refresh Upgrade Pack** instead:
   - moxa-UC-4400A_MIL3_V1.3-refresh-upgrade-pack
   - moxa-UC-4400A_MIL3_V1.3-refresh-upgrade-pack.sha512.bin.signed
4. Copy and configure the upgrade pack and its metadata to the UC-4434A-I-T's local cache using the **mx-sw-updater configure -p moxa-UC-4400A_MIL3_V1.3-upgrade-pack** command.

```
root@moxa-imoxa0920070:/home/moxa# sudo mx-sw-updater configure -p moxa-UC-
4400A_MIL3_V1.3-upgrade-pack
INFO[2024-10-13T04:24:44Z] configure successfully
```

NOTE

Replace step 1 and 2 with the **mx-sw-updater update** command if you intend to perform the upgrade remotely via the Moxa APT repository.

You can use the **mx-sw-updater list --compare V1.3** command to check the packages that will be upgraded when you apply the v1.3 upgrade pack.

```
root@moxa-imoxa0920070:/home/moxa# mx-sw-updater list --compare V1.3
INFO[2024-10-13T11:08:01Z] compare two packages: system and 1.3
 Name                 Arch     Version              NewVersion           Status
-------------------------------------------------------------------------------
emwicon-wmx7205-d... arm64    5.10.194-cip39-rt... 5.10.214-cip46-rt... upgraded
libmcm0              arm64    1.4.26-1+deb11       1.5.14-1+deb11       upgraded
libmm-glib0          arm64    1.20.4+moxa1-1+deb11 1.20.4+moxa2-1+deb11 upgraded
linux-headers-5.1... arm64    5.10.194-cip39-rt... 5.10.214-cip46-rt... upgraded
linux-image-5.10.... arm64    5.10.194-cip39-rt... 5.10.214-cip46-rt... upgraded
linux-kbuild-5.10... arm64    5.10.194-cip39-rt... 5.10.214-cip46-rt... upgraded
modemmanager         arm64    1.20.4+moxa1-1+deb11 1.20.4+moxa2-1+deb11 upgraded
moxa-bootloader-m... all      2.3.0-1+deb11        2.5.0-1+deb11        upgraded
moxa-computer-int... arm64    1.34.2-1+deb11       1.37.0-1+deb11       upgraded
moxa-connection-m... arm64    1.4.26-1+deb11       1.5.14-1+deb11       upgraded
moxa-image-archiv... all      1.5.0+deb11          1.7.0-1+deb11        upgraded
moxa-mil-base-sys... all      3.2.0-2-1+deb11      3.3.0-1+deb11u2      upgraded
moxa-mxview-one-m... all      1.5.0-1+deb11        1.6.1-1+deb11        upgraded
moxa-system-manager  all      2.22.3-1+deb11       2.23.1-1+deb11       upgraded
moxa-uc-4400a-bas... arm64    3.2.0+deb11u4        3.3.0+deb11u1        upgraded
```

5. Upgrade the system to v1.3.
   Enable auto-recovery and run the **`mx-sw-updater upgrade --local --system-failback`** command.

```
root@moxa-imoxa0920070:/home/moxa# sudo mx-sw-updater upgrade --local --
system-failback
INFO[2024-10-13T11:32:22Z] current version: V1.0, target version: 1.3

Would you like to continue? (y/N)y
Synchronize boot files...
             0   0%    0.00kB/s    0:00:00 (xfr#0, to-chk=0/2)
             0   0%    0.00kB/s    0:00:00 (xfr#0, to-chk=0/2)
Start creating replica...
    150,208,843  99%   97.63MB/s    0:00:01 (xfr#133, to-chk=0/269)
Type: replica
Create Time: 2024.10.13-11:32:35
Size: 145MB
The system failback has been enabled and the replica has been created
successfully.
```

6. Reboot the computer after the upgrade is complete.
7. Verify the system has been upgraded to v1.3 by using the **`mx-ver`** command.

```
root@moxa-imoxa0920070:/home/moxa# mx-ver
UC-4434A-I-T MIL3 version 1.3 Build 25101605
```

Online Upgrade

An example for using the **`mx-sw-updater`** command to perform an online OTA upgrade from OS image v1.0/1.1 to v1.3 (MIL 3.4.1).on a UC-4434A-I-T computer is as follows:

1. Sync the latest metadata from APT repo to local

```
root@moxa-imoxa1000030:/# sudo mx-sw-updater update
```

2. Perform OTA upgrade to the latest available MIL3 version

```
root@moxa-imoxa1000030:/# sudo mx-sw-updater upgrade
```

Online Update via Secure APT

Moxa Arm-based computers support SecureApt, which uses a GPG public key system to ensure the integrity and authenticity of patches are validated before download, and x.509 certification authentication for secure transmission via HTTPS. The private key pair of the GPG key for the Moxa APT repository is stored in an on-premises Sign Server, accessible only by authorized Moxa personnel.

NOTE

Click the following link for more information on how SecureAPT works: https://wiki.debian.org/SecureApt

Querying the System Image Version

Use the mx-ver command to check the system image version on your Arm-based computers.

```
moxa@moxa-tbzkb1090923:# mx-ver
UC-8220-T-LX-US-S MIL3 version 1.0 Build 22052300
```

Failback Update

We strongly recommend enabling the failback function before performing an update. Refer to failback feature in the Moxa System Manager (MSM) for details.

Managing the APT Repository

The APT Repository is the network server from which APT downloads packages that are installed on your Moxa Arm-based computer. By default, Moxa Arm-based computers include the following repositories that contain stable and well-tested packages best suited for ensuring the stability of your project.

Source list	Repository URL	Description
/etc/apt/sources.list	https://deb.debian.org/debian bullseye	Debian official repository containing the latest stable Debian 11 release (released about every 2 months)
	https://deb.debian.org/debian bullseye-updates	Debian official repository containing bug fixes that will be included in the upcoming Debian 11 release
	https://deb.debian.org/debian -security/bullseye-security	Debian official repository containing security hotfixes that will be included in the upcoming Debian 11 release
/etc/apt/sources.list.d/ moxa.list	https://debian.moxa.com/mil3 bullseye	Moxa repository containing Moxa's proprietary library, tools, utilities, and kernel. Moxa will maintain security and bug fixes even after Debian 11 has reached its end of life (EOL).

To add a new repository, you must add the repository URL and official GPG key to the source list and keyring in your Moxa Arm-based computer.

Here is an example for adding the Docker repository https://docs.docker.com/engine/install/debian/.

1. Add the repository URL to the source list on your Arm-based computer.

```
moxa@moxa-tbzkb1090923:# echo "deb https://download.docker.com/linux/debian
bullseye stable" > /etc/apt/sources.list.d/docker.list
```

2. Add the official GPG public key of the Docker repository to the keyring in your computer for SecureAPT.

```
moxa@moxa-tbzkb1090923:# curl -fsSL
https://download.docker.com/linux/debian/gpg | gpg --dearmor -o
/etc/apt/trusted.gpg.d/docker.gpg
```

3. Verify the newly added Docker repository by running an update.

```
moxa@moxa-tbzkb1090923:# apt update
Get:1 https://download.docker.com/linux/debian bullseye InRelease [43.3 kB]
Hit:2 http://deb.debian.org/debian bullseye InRelease Get:3
http://deb.debian.org/debian-security bullseye-security InRelease [48.4 kB]
Get:4 https://download.docker.com/linux/debian bullseye/stable amd64
Packages [13.8 kB] Get:5 http://deb.debian.org/debian bullseye-updates
InRelease [44.1 kB] Get:6 http://deb.debian.org/debian-security bullseye-
security/main amd64 Packages [191 kB] Fetched 341 kB in 1s (356 kB/s)
Reading package lists... Done Building dependency tree... Done Reading state
information... Done 30 packages can be upgraded. Run 'apt list --upgradable'
to see them.
```

Updating Your System

Preparing a Staging Environment

Since Moxa Arm-based computers are open platforms, you are free to install any software that you would like to use. However, we highly recommend that you test all new software on a staging platform before installing them on your production gateways.

Synchronizing the Repository Information

The first and most important step is to synchronize the package index files in your Arm-based computer with the source repositories specified in the file **/etc/apt/sources.list**. When you perform the synchronization, information related to the packages, including versions and dependencies, will also be downloaded from the repositories.

To perform the synchronization, make sure that your network environment can connect to the APT repositories, and then run the **apt update** command with root permission to synchronize the package index.

```
moxa@moxa-tbbbb1182827:# sudo apt update
```

Updating the Entire System

Use the **apt full-upgrade** command to upgrade all packages used by your Moxa Arm-based computer to latest versions.

```
moxa@moxa-tbbbb1182827:# sudo apt full-upgrade
```

Updating the Bootloader

When a updated Bootloader firmware is available, Moxa will publish a notification on the Moxa Arm-based computer product page and upload the new firmware to the Moxa APT repository. You can download the firmware (**.bin** format) via SecureAPT so that the authenticity and integrity of the firmware is verified.

NOTE

Click the following link for more information on how SecureAPT
https://wiki.debian.org/SecureApt

Querying the Current Bootloader Version

Use the **mx-bootloader-mgmt upgrade -i** command to check the current bootloader version of your Arm-based computer.

```
root@moxa-imoxa1000030:/# mx-bootloader-mgmt upgrade -i

Current bootloader information:
compatible model: UC-4400A
bootloader version: 1.0.0S08
sha256sum: 8be98fd0804234099111cdc98a08ed69c075a8ff6433d2995e00726284ef5c31
md5sum: cfdb7b3e89e1eef3d291731f0e784d27
```

Downloading the latest Bootloader

Use the **sudo apt update && sudo apt install -y moxa-[computer series name]-uboot** to download the bootloader .bin file

After installation, the .bin file will be located at:
/lib/firmware/moxa/bootloader/[computer-series-name]

Starting with MIL 3.4, you can use the command **mx-bootloader-mgmt upgrade -l** or –list to display detailed information about the downloaded bootloader .bin file, including the compatible model, version, file path, and hash.

Moxa Computer Series	How to Download Bootloader
UC-8200 Series	apt update && apt install -y moxa-uc-8200-uboot
UC-1222A Series UC-2222A Series	apt update && apt install -y moxa-uc-1200a-uc-2200a-uboot
UC-3400A Series	apt update && apt install -y moxa-uc-3400a-uboot
UC-4400A Series	apt update && apt install -y moxa-uc-4400a-uboot
V1200 Series	apt update && apt install -y moxa-v1200-uboot

Example: Downloading the bootloader .bin file for your UC-8200 Series computer

```
root@moxa-tb11827:/# sudo apt update && sudo apt install -y moxa-uc-8200-uboot
root@moxa-tb11827:/# cd /lib/firmware/moxa/bootloader/uc-8200
root@moxa-tb11827:/lib/firmware/moxa/bootloader/uc-8200# ls
u-boot.bin
```

Updating Bootloader

Use the **mx-bootloader-mgmt upgrade -f [file path]** command to update the Bootloader

```
root@moxa-tbbbb1182827:# sudo mx-bootloader-mgmt upgrade -f
/lib/firmware/moxa/bootloader/uc-8200/bootloader.bin

The version of bootloader being updated: 3.0.0S07
The version of current bootloader: 3.0.0S07
Your bootloader version is the same as the version of bootloader being updated.
Do you want to continue? (y/N)y
Start to upgrade bootloader…
Upgrade /dev/mtd1 bootloader to version 3.0.0S07 successfully
```

Enable the Failback Function Before Update

We highly recommend enabling the failback function before performing a bootloader update because a power outage may cause the device to be unable to boot. For details, see failback feature in the Moxa System Manager (MSM) tool.

7. Backup, Decommission, and Recovery

In this chapter, we will introduce how to use Moxa System Management (MSM) utility to perform snapshot, backup, decommission, and recovery of your system. MSM provides an automatic failback mechanism to ensure that the device can recover to the last known working and secure state when the device fails after a critical event such as a system update.

Function	Description
Snapshot	• The snapshot has a smaller footprint as it saves just the differences (partition 3 in Figure 7.1) compared to the out-of-factory rootfs (partition 2 in Figure 7.1). • The snapshot is saved in the Moxa Arm-based computer and cannot be exported. Hence, a snapshot can only be used to restore the computer that the snapshot was taken from.
Backup	• The backup has a larger footprint as it saves the entire system including the out-of-factory rootfs. • The backup can be exported to an external storage. • The backup can be used to restore the Moxa Arm-based computer that the backup is taken from or another computer of the same model.
Automatic Failback Recovery	• When failback recovery is enabled, a replica of the system including the snapshot and bootloader is created. • If a boot failure event occurs after failback recovery is enabled, the system will automatically use the replica to recover the system • Failback recovery should be enabled before performing any critical actions that may potentially result in a device failure (e.g., power loss during a bootloader update could brick a computer).

Below diagram illustrate an overview of MIL3 system layout:

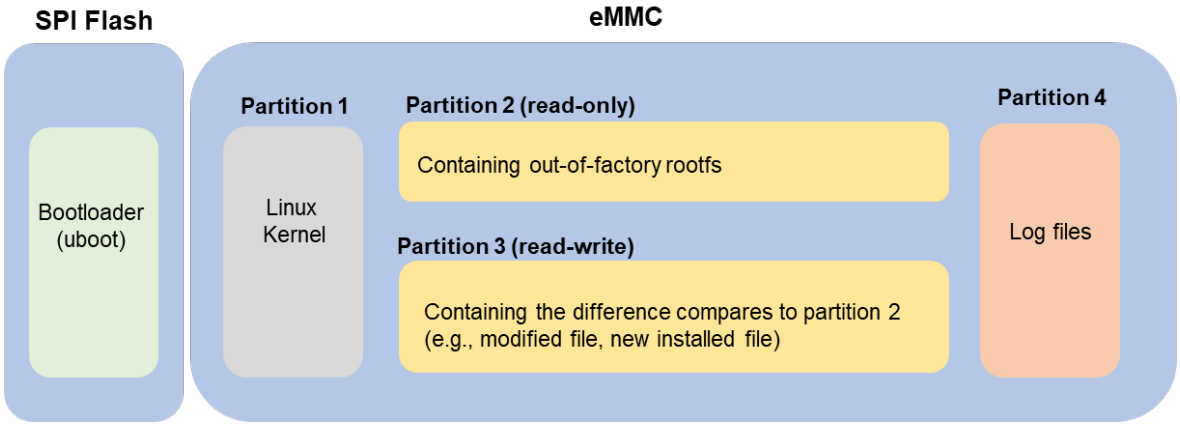


Figure 7.1 - Layout Overview of Arm-based Computer with MIL3

Creating a System Snapshot

A snapshot preserves the state and data of the Moxa Arm-based computer as a restoration point at a specific point in time so that you can restore it to that point if something goes wrong. Snapshots only save the Linux kernel and new and modified files to the out-of-factory rootfs (partition 2). Therefore, the size of a snapshot is much smaller than a backup.

Use the **`# sudo mx-system-mgmt snapshot <sub-command> <options> <flag>`** to create restore a system. You must use **`sudo`** or run the command with root permission.

Sub-commands	Description
create	Creates a snapshot of system • A snapshot includes **kernel (partition 1)** and **rootfs (partition 3)** • Only one snapshot is saved. A new snapshot will overwrite the previous snapshot • Snapshot is stored in **rootfs (partition 3)**
restore	Restores the system with the snapshot. System failback will be disabled after a system is restored from the snapshot.
delete	Deletes the existing snapshot
info	Displays the creation time and size of the existing snapshot

Options	Description
--cold	Creates a snapshot after restarting the system in a minimal environment, such as initrd. This ensures data consistency but requires system downtime during the snapshot creation process.
--hot	• This is the default mode if neither the **--cold** nor **--hot** options are specified. • Using **--hot** creates a snapshot of the system while it remains fully operational, without requiring system downtime. **Caution:** While the hot snapshot method minimizes disruption, there is a potential risk of data inconsistencies due to changes that may occur during the snapshot process.
--size	Estimates the additional disk space required to create the snapshot.

Flag	Description
-y or --yes	Automatically consent to the prompts during create, restore, and delete processes

WARNING

Before initiating the backup or snapshot process with **--hot** option, it is crucial to ensure that all active services involving customer-developed software, are temporarily disabled. Services that are running during the backup may lock certain files, preventing them from being copied and resulting in an incomplete backup. This can compromise the integrity of your backup and the ability to fully restore your system later.

Creating a System Backup

Compares to snapshot, a backup saves Linux kernel and the rootfs on your Moxa Arm-based Computer. Therefore, a backup can be exported and use to restore a Moxa Arm-based computer of the same model with MIL 3.0. For example, if you create a backup on UC-8200 Secure model with MIL3, you can use the backup to restore another UC-8200 Secure model with MIL3

Use **`# sudo mx-system-mgmt backup <sub-command> <options> <flag>`** command to create, delete, and restore a backup. You must use **`sudo`** or run the command with the root permission.

Sub-commands	Description
create	Creates a backup of the system • The backup includes **kernel (partition 1), rootfs (partition 2)**, and **rootfs (partition 3)** • By default, the backup is created in the **/boot_device/p3/backup/** directory with the name **backup.tar**, together with an info file that contains the backup information and cryptographic hash of the backup. • The backup includes system snapshot. If you would like to reduce the size of backup, you can delete the snapshot in the system before performing the backup if the snapshot is not needed.
delete	Deletes the backup from the default directory
restore	Restores the system using the backup from default directory. • System failback will be disabled after restoration. • Existing snapshot on system will be deleted after restoring the system from a backup. • The cryptographic hash in the **info** file will be used to validate the integrity of the backup file before the restore process begins. • A system reboot is required after restoration.
info	Displays the creation time and size of the backup in the default directory

Options	Description
--cold	Creates a backup after restarting the system in a minimal environment, such as initrd. This ensures data consistency but requires system downtime during the backup creation process. *Note: This feature is available in MIL v3.2 and later versions.*
--hot	• This is the default mode if neither the **--cold** nor **--hot** options are specified. • Using **--hot** creates a snapshot of the system while it remains fully operational, without requiring system downtime. **Caution:** While the hot backup method minimizes disruption, there is a potential risk of data inconsistencies due to changes that may occur during the backup process. Ensure that all active services involving customer-developed software, are temporarily disabled.
--compress	Create a backup with compression. Please note that this might result in a significantly longer backup time *Note: This feature is available in MIL v3.3 and later versions.*
-D or --directory	Specifies the directory (e.g., /media/USB_p1) where the backup will be created
--size	Estimates the additional disk space required to create the backup.

Flag	Description
-y or --yes	Automatically consent to the prompt during create, delete and restore

ATTENTION

When restoring a backup from one Moxa computer to multiple other Moxa computers, the SSH host key will be identical across all devices. If you need each computer to have a unique SSH host key, ensure you regenerate the host key after restoring the backup.

The following example demonstrates how to perform a system backup using the hot method to a USB storage drive mounted at **/media/USB_p1**:

```
moxa@moxa-tbzkb1090923:# sudo mx-system-mgmt backup create --hot -D
/media/USB_p1
Set /media/USB_p1 as backup directory.
Check the backup information...
There is no backup information
Start evaluating space, please wait...
Estimation of Required Space: 628MB
Available Space: 32756MB
Would you like to continue? (y/N)y
Synchronize boot files...
             0    0%     0.00kB/s     0:00:00 (xfr#0, to-chk=0/2)
Start creating backup file...
 628MiB 0:00:57 [11.0MiB/s] [ <=> ]
Type: backup
Create Time: 2021.11.06-17:32:29
Size: 628MB
The backup has been created successfully under: /media/USB_p1
```

The following example shows how to restore a backup from the USB storage drive with the mounting point **/media/USB_p1**:

```
moxa@moxa-tbzkb1090923:# sudo mx-system-mgmt backup restore -D /media/USB_p1
Set /media/USB_p1 as backup directory.
Check the backup information...
Type: backup
Create Time: 2021.11.06-17:44:43
Size: 628MB
Start verifying backup file, please wait...
Verified OK!
Start evaluating space, please wait...
Estimation of Required Space: 628MB
Available Space: 5125MB
Would you like to continue? (y/N)y
Check the snapshot information...
Type: snapshot
Create Time: 2021.11.06-15:42:47
Size: 235MB
This will delete the existing snapshot.
Do you want to continue? (y/N)y
Check the snapshot information...
Type: snapshot
Create Time: 2021.11.06-15:42:47
Size: 235MB
The snapshot has been deleted successfully.
To restore the backup file will overwrite current system and factory default
system.
Do you want to continue? (y/N)y
Start using the backup file to restore the system...
 628MiB 0:01:00 [10.4MiB/s] [==============================================>]
100%
Synchronize boot files...
              0   0%    0.00kB/s    0:00:00 (xfr#0, to-chk=0/2)
System has been restored successfully. Reboot is required to take effect.
moxa@moxa-tbzkb1090923:# sudo reboot
```

> **WARNING**
>
> Before initiating the backup or snapshot process with **--hot** option, it is crucial to ensure that all active services involving customer-developed software, are temporarily disabled. Services that are running during the backup may lock certain files, preventing them from being copied and resulting in an incomplete backup. This can compromise the integrity of your backup and the ability to fully restore your system later.

Setting the System to the Default

Press and hold the **FN** button for 7 to 9 seconds to reset the computer to the factory default settings. When the reset button is held down, the LED will blink once every second. The LED will become steady when you hold the button continuously for 7 to 9 seconds. Release the button immediately when the LED become steady to load the factory default settings. For additional details on the LEDs, refer to the quick installation guide or the user's manual for your Arm-based computer

> **ATTENTION**
>
> **Reset-to-default will erase all data stored on the boot-up storage**
>
> Back up your files before resetting the system to factory defaults. All the data stored in the Arm-based computer's boot-up storage will be destroyed after resetting to factory defaults, except for snapshots and backups located under /boot_device/p3.

You can also use the **sudo mx-system-mgmt default restore** command to restore the computer to factory default settings. You must use sudo or run the command with the root permission.

```
moxa@moxa-tbzkb1090923:/# sudo mx-system-mgmt default restore
```

If you would like to configure the **FN** button for a different action (e.g., restore to a snapshot), refer to Customize the Button Action section.

Decommissioning the System

Compared with the set-to-default function, decommissioning will further erase all data stored in the log partition and /boot_device/p3 to help erase security-sensitive information.

> **ATTENTION**
>
> **Decommission will erase all the data including event and audit logs**
>
> Please back up your files before resetting the system to factory defaults. All user data including logs in your Arm-based computer will be destroyed after performing decommissioning. Bootloader configuration, including administrator password, will also be set to factory default.

You can also use the **sudo mx-system-mgmt default decommission** command to restore the computer to factory default. You must use sudo or run the command with the root permission.

```
moxa@moxa-tbzkb1090923:/# sudo mx-system-mgmt default decommission
```

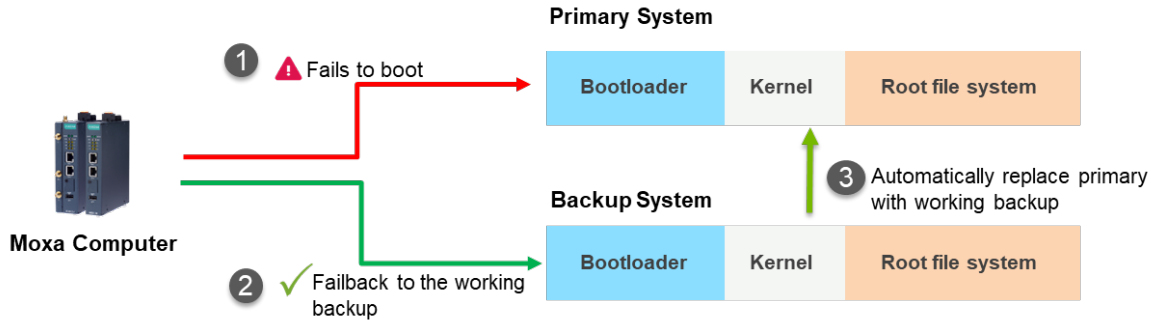
The decommissioning process will do the following:

1. Overwrite the system partition 4 times with shred so that all user files will be deleted and cannot be recovered.
2. Overwrite the log partition 4 times with shred so that all log files will be deleted and cannot be recovered.
3. Trigger the bootloader decommissioning function, so all configurations and log messages in the bootloader are also deleted and cannot be recovered.

System Failback Recovery

A system bootup failure may occur when critical files are lost or corrupted. A typical and common cause of boot up failure is power lost during system update. Moxa System Management (MSM) provides system failback capability which can automatically recovers your system to the last known working state if boot up failure is detected after critical change(s) are made to the primary system. The boot failure criteria are customizable by user.

Before applying critical update or changes to the device, it is recommended to enable system failback first.



Use **`# sudo mx-system-mgmt system-failback <sub-command> <flag>`** to enable or disable system failback. You must use sudo or run the command with the root permission.

Sub-commands	Description
enable	Enables system failback and create a replica of the system • The replica includes **Bootloader**, **kernel (partition 1)** and **rootfs (partition 3)** • The replica is stored in **rootfs (partition 3)** • When the Moxa Arm-based computer fails to boot up, the device will automatically reboot and replace the broken system with the working replica. • The replica includes a system snapshot. If you would like to reduce the size of the replica, you can delete the snapshot if you no longer need it.
disable	Disables the system failback and delete the existing system replica
Info	Displays the create time and size of replica
State	Displays the status of system failback (enabled/disabled)

Options	Description
--cold	Creates a replica after restarting the system in a minimal environment, such as initrd. This ensures data consistency but requires system downtime during the replica creation process.
--hot	• This is the default mode if neither the **--cold** nor **--hot** options are specified. • Using **--hot** creates a replica of the system while it remains fully operational, without requiring system downtime. **Caution:** While the hot replica method minimizes disruption, there is a potential risk of data inconsistencies due to changes that may occur during the replica creation process.
--size	Estimates the additional disk space required to create the replica.
-V or --value	Displays only the binary value of the system failback state: • Enabled : 1 • Disabled: 0 **Example:** **mx-system-mgmt system-failback state -V**

Flag	Description
-y or --yes	Automatically consent to the prompts during the enable and disable processes

WARNING

Before initiating the replica creation process with **--hot** option, it is crucial to ensure that all active services involving customer-developed software, are temporarily disabled. Services that are running during the creation process may lock certain files, preventing them from being copied and resulting in an incomplete replica. This can compromise the integrity of your replica and the ability to fully recover your system later.

Below is an example of how to enable system failback using the hotession s method and display the information of the system replica:

```
moxa@moxa-tbzkb1090923:/# sudo mx-system-mgmt system-failback enable
Start evaluating space, please wait...
Estimation of Required Space: 233MB
Available Space: 5333MB
Would you like to continue? (y/N) y
Start processing...
Synchronize boot files...
             0   0%    0.00kB/s    0:00:00 (xfr#0, to-chk=0/2)
             0   0%    0.00kB/s    0:00:00 (xfr#0, to-chk=0/2)
Start creating replica...
   244,670,045  99%   11.94MB/s   0:00:19 (xfr#170, to-chk=0/294)
Type: replica
Create Time: 2021.11.06-14:35:14
Size: 235MB
The system failback has been enabled and the replica has been created
successfully.
moxa@moxa-tbzkb1090923:/# sudo mx-system-mgmt system-failback info
Check the replica information...
Type: replica
Create Time: 2021.11.06-14:35:14
Size: 235MB
```

Customize the Boot Up Failure Criteria

If you would like to customize the boot failure criteria, you can edit below script to add criteria you like Moxa System Manager to check.

/etc/moxa-system-manager/check-hooks.d/99-example.sh

In below example in **99-example.sh**, Moxa System Manager will consider the boot up is successful if "moxa-connection-manager.service" start successfully by returning a zero value. If the program returns a non-zero value, the moxa-system-manager service will not mark this startup as successful, and it will enter the system-failback process to restore the system.

```
#systemctl is-active moxa-connection-manager.service && exit 0 || exit 1
```

8. Security Capability

In this chapter, we will introduce Moxa Arm-based computers key security functions and a security hardening guide to deploy and operate Moxa computer in a secure manner

Communication Integrity and Authentication

Below is a list of network communication services and protocols available in the Moxa Arm-based computer and their data integrity and authentication protection mechanisms.

Service	Protocol	Data Integrity	Data Authentication
SSH server and client	SSH	HMAC algorithm is used to guarantee data integrity	Uses key signature algorithms such as ED25519, ECDSA, or RSA to verify authenticity.
SFTP server	SSH		
SCP server	SSH		
APT client	HTTPS	SecureAPT uses checksum to guarantee data integrity	SecureAPT uses GPG public key system to validate data authenticity
NTP client (NTS support)	TLS/SSL, NTP	NTS guarantees data integrity via NTS Authenticator and Encrypted EF	NTS provides TLS layer to guarantee authenticity
Device Discovery	mDNS	The mDNS protocol doesn't implement data integrity and authentication protection.	

ATTENTION

For post-installed communication services and protocols, you must ensure data integrity and authentication are implemented. If integrity and authentication are not available, you must use additional compensating countermeasures in system to compensate the risk. For example, physical cable protection for serial Modbus RTU.

User Account Permissions and Privileges

Switching to the Root Privilege

In Moxa Arm-based computers, the root account is disabled in favor of better security. The default user account **moxa** belongs to the sudo group. Sudo is a program designed to let system administrators allow permitted users to execute some commands as the root user or another user. The basic philosophy is to give as few privileges as possible but still allow people to get their work done. Using sudo is better (safer) than opening a session as root for a number of reasons, including:

- Nobody needs to know the root password (sudo prompts for the current user's password). Extra privileges can be granted to individual users temporarily, and then taken away without the need for a password change.
- It is easy to run only the commands that require special privileges via sudo; the rest of the time, you work as an unprivileged user, which reduces the damage caused by mistakes.
- Some system-level commands are not available to the user moxa directly, as shown in the sample output below:

```
moxa@Moxa-tbzkb1090923:~$ sudo ifconfig
eth0      Link encap:Ethernet  HWaddr 00:90:e8:00:00:07
          inet addr:192.168.3.127  Bcast:192.168.3.255  Mask:255.255.255.0
          UP BROADCAST ALLMULTI MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
```

```
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)

eth1      Link encap:Ethernet  HWaddr 00:90:e8:00:00:08
          inet addr:192.168.4.127  Bcast:192.168.4.255  Mask:255.255.255.0
          UP BROADCAST ALLMULTI MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)

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

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

NOTE

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

ATTENTION

You might get the permission denied message when using pipe or redirect behavior with a non-root account.

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

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

Controlling Permissions and Privileges

Moxa Industrial Linux uses Discretionary Access Control (DAC) based on Access Control Lists (ACLs) to manage permissions and privileges, which an object has an owner that controls the permissions to access the object. Subjects can transfer their access to other subjects. In other words, the owner of the resource has full access and can determine the access type (rwx: read, write, execute) of other users.

You can use **`chmod`** command to configure who (user, group, other) can do what (read, write, execute) to a file or directory. The access permission is extended by Access Control Lists (ACLs) authorization. ACL provides a more flexible mechanism that allows multiple users and groups to own an object. You can check and configure access control lists of a specific file or directory using **`getfacl`** and **`setfacl`** commands.

NOTE

Click the following link for more information on usages of chmod and Access Control Lists (ACLs)
https://wiki.debian.org/Permissions

Moxa Arm-based computers only provide one account in sudo group by default because it is intended for the system integrator to customize and build their applications on top.

The system integrator shall be responsible for setting the appropriate permissions to roles and user accounts to enforce the concept of least privilege.

Linux Login Policy

Invalid Login Attempts

Moxa Industrial Linux provides the capability to configure allowed invalid login attempts to mitigate against Denial-of-Service (DoS) and Brute-force attack.

Model Type	Default Rule
Secure model	**[5]** consecutive invalid login within **[60]** seconds will deny access for **[300]** seconds.
Standard model	**[5]** consecutive invalid login within **[60]** seconds will deny access for **[300]** seconds.

Following is the configuration file and variable to configure the setting:

Configuration Option	Configuration file	Variable to Set
Consecutive invalid login	/etc/security/faillock.conf	deny
Within how many seconds	/etc/security/faillock.conf	fail_interval
Deny access for how long (in seconds)	/etc/security/faillock.conf	unlock_time

More configurable options can be found in following reference:

- login.defs(5) > login > Debian bullseye > Debian Manpages
- faillock.conf(5) > libpam-modules > Debian bullseye > Debian Manpages

Session Termination After Inactivity

This setting automatically terminates the login sessions after a standard period of inactivity. Below is the default configuration set in Moxa Arm-based computer.

Security Model	Default Value
Secure model	• Automatically logout standard user after 900 second of inactivity • Automatically terminate root privilege of sudo user after 900 second of inactivity
Standard model	Not applicable; requires manual configuration. For more information about session termination, visit: https://manpages.debian.org/bullseye/bash/bash.1.en.html

Follow below instructions to configure the inactivity time:

Login Method	Configuration
Serial Console and SSH (Secure Shell)	• Set the value (in seconds) of variable **TMOUT** in **/etc/profile.d/99-moxa-profile.conf** • Apply the same value to variable **ClientAliveInterval** in **/etc/ssh/sshd_config.d/00-moxa-sshd.conf** • To apply the rule to sudo user, make sure variable **env_keep+="TMOUT"** exist in **/etc/sudoers.d/00-moxa-sudoers-conf**

NOTE

The SSH session termination takes effect upon the next login and does not apply to currently active sessions.

Login Banner Message

You can set a message banner message to displaying welcome or informational messages or warming message to un-authorized users. Follow below instructions to add a banner Moxa Industrial Linux 3.0 UM for Arm-based Computers Moxa Industrial Linux 3.0 UM for Arm-based Computers.

Login Method	Banner Content	Additional Configuration Required
Serial Console	/etc/issue	n/a
SSH (Secure Shell)	/etc/issue.net	Add variable **Banner /etc/issue.net** is added in **/etc/ssh/sshd_config.d/00-moxa-sshd.conf**

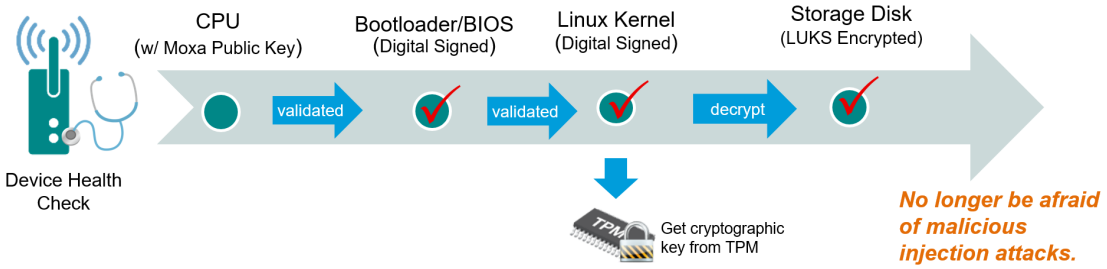
Bootloader Login Policy

For bootloader login policy management, refers to the bootloader configuration section.

Secure Boot and Disk Encryption

Secure boot and disk encryption are available in Secure model, designed to make platform integration more secure. Moxa's secure boot process begins from CPU as hardware root-of-trust to ensure integrity and authenticity of bootloaders and Linux kernels are validated with Moxa digital signature before execution, preventing malicious or un-authenticated bootloader and kernels to run on Moxa Arm-based computer.

Next, only after bootloader and kernel have been validated, the LUKS (Linux Unified Key Setup) encrypted root file system (rtfs) will be decrypted by a key provisioned in TPM during factory production. The disk encryption prevent confidential data could be read without authorization when the device is stolen or lost.



Public key infrastructure (PKI)

Moxa secure boot use X.509 public key infrastructure (PKI) to validate authenticity and integrity of bootloader and Linux kernel.

How are private keys protected?

Private keys used to digital sign Moxa software are stored in on-premises tamper and intrusion-resistant hardware security module (HSM), where strict access authorization and 24-hour video surveillance are applied.

Key lifecycle and revocation

In an unlikely scenario where the private key stored in HSM is compromised, Moxa will announce the news on Moxa Security Advisory, including instructions to revoke the compromised public key burned in the CPU via a utility downloadable from Moxa APT repository. Then update the bootloader and system image signed by a new private key.

ATTENTION

DO NOT arbitrarily replace the kernel or bootloader on Secure models, or the computer will not be able to boot up.

Trusted Platform Module (TPM 2.0)

The Moxa Arm-based computer includes a TPM 2.0 hardware module. TPM provides a hardware-based approach to manage user authentication, network access, data protection and more that takes security to higher level than software-based security. It is strongly recommended to manage keys with TPM and also store digital credentials such as passwords

The TPM can be managed via the tpm2_tools pre-installed in Moxa Industrial Linux (https://github.com/tpm2-software/tpm2-tools).

TPM software stack & tool is maintained by tpm2-software community
https://tpm2-software.github.io/

A good reference of TPM 2.0 introduction
https://link.springer.com/chapter/10.1007/978-1-4302-6584-9_3

Host Intrusion Detection

Secure model of Moxa Arm-based computer comes with **AIDE** (Advanced Intrusion Detection Environment) preconfigured. AIDE is a lightweight but powerful host intrusion detection utility for checking the integrity of files.

The out-of-factory Moxa Arm-based computer comes with a database created by AIDE at the first time bootup containing all security configurations set by Moxa. You can compare the system's status against this database to find out if there is any integrity breach. You can also update the database after making changes to the configuration or adding additional software.

Default Monitored Files

Below are the security configuration files and directories included in the default database created by Moxa.

- The database is **`aide-moxa.db`** and put under **`/var/lib/aide/aide-moxa.db.`**
- The configuration file of AIDE is **`/etc/aide/aide-moxa.conf`**; you can add additional files and directories to the database.

Configuration Type	Path
File	/etc/adduser.conf
	/etc/login.defs
	/etc/logrotate.conf
	/etc/nftables.conf
	/etc/profile
	/etc/rsyslog.conf
	/etc/sudoers
	/etc/security/pwquality.conf
	/etc/sysctl.conf
	/etc/moxa/moxa-guardian/
Directory	/etc/aide/
	/etc/audit/
	/etc/logrotate.d/
	/etc/moxa/MoxaComputerInterfaceManager/
	/etc/moxa/MoxaConnectionManager/
	/etc/moxa/moxa-guardian/
	/etc/pam.d
	/etc/security/
	/etc/profile.d/
	/etc/rsyslog.d/
	/etc/ssh/
	/etc/sudoers.d
	/var/lib/moxa-guardian/
	/etc/chrony/
	/etc/fail2ban/
	/etc/fstab
	/etc/security/pwquality.conf.d/
	/etc/sysctl.d/

To run a comparison between current system against the Moxa AIDE database, run **aide --check -c /etc/aide/aide-moxa.conf**. You can also check the result from **/var/log/aide/aide.log**.

```
moxa@moxa-tbbbb1182827:/# sudo aide --check -c /etc/aide/aide-moxa.conf
Start timestamp: 2022-06-12 13:47:38 +0000 (AIDE 0.17.3)
AIDE found NO differences between database and filesystem. Looks okay!!

Number of entries:      254
---------------------------------------------------
The attributes of the (uncompressed) database(s):
---------------------------------------------------
/var/lib/aide/aide-moxa.db
 MD5       : A8wKxphrNVlWz31AVf3esA==
 SHA256    : trGvVioXdZf/RISmj3v60mQsmcrqK4kV
             sUFm068cLOs=

End timestamp: 2022-06-12 13:47:39 +0000 (run time: 0m 1s)
```

To update the database after you have make configuration changes, run **aide --init -c /etc/aide/aide-moxa.conf**.

You should see following output which created a new AIDE database **aide-moxa.db.new** under **/var/lib/aide.**

```
moxa@moxa-tbbbb1182827:/# sudo aide --init -c /etc/aide/aide-moxa.conf

Start timestamp: 2022-06-12 14:39:30 +0000 (AIDE 0.17.3)
AIDE initialized database at /var/lib/aide/aide-moxa.db.new

Number of entries:      254
---------------------------------------------------
The attributes of the (uncompressed) database(s):
---------------------------------------------------

/var/lib/aide/aide-moxa.db.new
 MD5       : Mb74vEG93jjVfJMGSZa+DA==
 SHA256    : ENl5QGVgYXuKEwE3FSXRfzxl3vJg0TxU
             WsQnHN16E74=

End timestamp: 2022-06-12 14:39:30 +0000 (run time: 0m 0s)
```

For AIDE to use the new database, you need to rename it to **aide-moxa.db**.

```
moxa@moxa-tbbbb1182827:/# sudo mv /var/lib/aide/aide-moxa.db.new
/var/lib/aide/aide-moxa.db
```

At this point, you can run **aide --check -c /etc/aide/aide-moxa.conf** to compare current system against the updated AIDE database.

How to Perform Authenticity and Integrity Check on All Files

If you would like to ensure authenticity and integrity of all files in the Moxa Arm-based computer, you can create a openSSL signed database containing every single file under the filesystems, then validate the authenticity of the database before using AIDE to check the integrity of all files in the filesystem. Following below steps to create such AIDE database.

1. Create a database using **/etc/aide/aide-fs-moxa.conf**; this configuration file monitors every single file in the filesystem.

```
moxa@moxa-tbbbb1182827:/# sudo aide --init -c /etc/aide/aide-fs-moxa.conf
```

2. Rename the created database to **/var/lib/aide/aide-fs-moxa.db.**
3. Generate a 4096-bit RSA private key.

```
moxa@moxa-tbbbb1182827:/# sudo openssl genrsa -out aide-key.pem 4096
Generating RSA private key, 4096 bit long modulus (2 primes)
.....................................................................+
+++
.................................+++
e is 65537 (0x010001)
Enter pass phrase for aide-key.pem:
```

ATTENTION

You MUST keep the private key and pass phrase in a secure location.

4. Generate a public key from the private key:

```
moxa@moxa-tbbbb1182827:~$ sudo openssl rsa -in aide-key.pem -pubout -out
aide-
key.pub
Enter pass phrase for aide-key.pem:
writing RSA key
moxa@moxa-tbbbb1182827:~$
```

5. Generate a digital signature of **aide-filesystem-moxa.db** by the private key.

```
moxa@moxa-tbbbb1182827:~$ sudo openssl dgst -sha256 -sign aide-key.pem -out
aide-filesystem-moxa.db.sha256 /var/lib/aide/aide-fs-moxa.db
Enter pass phrase for aide-key.pem:
```

6. Now, you can distribute the database, public key and signed signature to other location, such as a centralized remote system.
7. Verify the database has been tampered or not.

```
moxa@moxa-tbbbb1182827:~$ sudo openssl dgst -sha256 -verify aide-key.pub -
signature aide-filesystem-moxa.db.sha256 /var/lib/aide/aide-fs-moxa.db
Verified OK
```

8. After the AIDE database' authenticity has been validated, you can run a comparison between current system against the AIDE database using **aide --check -c /etc/aide/aide-fs-moxa.conf**

NOTE

Click the following link for more information on usages of AIDE
https://manpages.debian.org/bullseye/aide-dynamic/aide.1.en.html

Intrusion Prevention

Fail2ban is pre-installed in Moxa Industrial Linux as an intrusion prevention software framework designed to prevent against brute-force attacks

> **NOTE**
>
> Click the following link for detail instructions of Fail2ban usage
> https://www.fail2ban.org/wiki/index.php/Main_Page

Network Security

Zeek for Network Security Monitoring

Zeek is pre-installed in Moxa Industrial Linux for network security monitoring. Zeek is a passive network traffic analyzer. Many operators use Zeek as a network security monitor (NSM) to support investigations of suspicious or malicious activity. Zeek also supports a wide range of traffic analysis tasks beyond the security domain, including performance measurement and troubleshooting. Zeek provides an extensive set of logs describing network activity. These logs include not only a comprehensive record of every connection seen on the wire, but also application-layer transcripts

If you have configured **cellular(4G/LTE)** and **ethernet** networks in Moxa Connection Manager (MCM). You can also enable Zeek to monitor the network traffic of these interfaces. Following the simple instruction below:

1. Export the Zeek environment.

```
export PATH=$PATH:/opt/zeek/bin
export ZEEK_PREFIX=/opt/zeek
```

2. [Required] Configure the interface to monitor by running # vim $ZEEK_PREFIX/etc/node.cfg.
3. [Required] Modify the interface list according to the interface you like to monitor. For example, add LAN1, LAN2, and cellular (4G/LTE) in the list.

```
# This example has a standalone node ready to go except for possibly
changing
# the sniffing interface.

# This is a complete standalone configuration.  Most likely you will
# only need to change the interface.
[zeek]
type=standalone
host=localhost
interface=eth0,eth1,wwan0
```

4. [Optional] change the **MailTo** email address to a desired recipient and the **LogRotationInterval** to a desired log archival frequency

   vim $ZEEK_PREFIX/etc/zeekctl.cfg

```
# Recipient address for all emails sent out by Zeek and ZeekControl.
MailTo = root@localhost

# Rotation interval in seconds for log files on manager (or standalone)
node.
# A value of 0 disables log rotation.
LogRotationInterval = 3600
```

5. [Required] Run **$ZEEK_PREFIX/bin/zeekctl** to start **Zeek**

```
root@moxa-tbbbb1182827:/home/moxa# $ZEEK_PREFIX/bin/zeekctl

Hint: Run the zeekctl "deploy" command to get started.
Welcome to ZeekControl 2.4.0

Type "help" for help.
[ZeekControl] >
```

6. [Required] For the first-time use of the shell, use **install** command to perform initial installation of the ZeekControl configuration.

```
[ZeekControl] > install

creating policy directories ...
installing site policies ...
generating standalone-layout.zeek ...
generating local-networks.zeek ...
generating zeekctl-config.zeek ...
generating zeekctl-config.sh ...
[ZeekControl] >
```

7. [Required] Start Zeek instance by **Start** command (Use CTRL+D to exit if initializing successfully).

```
[ZeekControl] > start

starting zeek ...
(zeek still initializing)
```

8. View the Zeek logs under **$ZEEK_PREFIX/logs**.

```
root@moxa-tbbbb1182816:/# ls -alh /opt/zeek/logs/current/
total 96K
drwxr-sr-x 2 root zeek 4.0K Jun 19 04:18 .
drwxrws--- 1 root zeek 4.0K Jun 19 04:17 ..
-rw-r--r-- 1 root zeek  250 Jun 19 04:18 capture_loss.log
-rw-r--r-- 1 root zeek  128 Jun 19 04:17 .cmdline
-rw-r--r-- 1 root zeek  583 Jun 19 04:18 conn.log
-rw-r--r-- 1 root zeek  352 Jun 19 04:17 .env_vars
-rw-r--r-- 1 root zeek  30K Jun 19 04:17 loaded_scripts.log
-rw-r--r-- 1 root zeek  753 Jun 19 04:18 notice.log
-rw-r--r-- 1 root zeek  227 Jun 19 04:17 packet_filter.log
-rw-r--r-- 1 root zeek    5 Jun 19 04:17 .pid
-rw-r--r-- 1 root zeek   61 Jun 19 04:17 .startup
-rw-r--r-- 1 root zeek  686 Jun 19 04:17 stats.log
-rwxr-xr-x 1 root zeek   19 Jun 19 04:17 .status
-rw-r--r-- 1 root zeek   19 Jun 19 04:17 stderr.log
-rw-r--r-- 1 root zeek  204 Jun 19 04:17 stdout.log
-rw-r--r-- 1 root zeek  367 Jun 19 04:18 weird.log
```

NOTE

Click the following link for Zeek's detail instruction and also the explanation on log types
https://docs.zeek.org/en/master/quickstart.html

If you prefer not to use ZeekControl (e.g., you don't need its automation and management features), you can refer to https://docs.zeek.org/en/master/quickstart.html#zeek-as-a-command-line-utility on how to directly control Zeek for your analysis activities from the command line for both live traffic and offline working from traces.

Enhance DNS Security

Set up DNSSEC and DNS-over-TLS (DoT) with the systemd-resolved service and enable it. The detailed steps are:

1. Make sure NetworkManager is configured to use systemd-resolved for DNS management

```
root@moxa-imoxa1234567:/home/moxa# cat /etc/NetworkManager/conf.d/00-nm-
mcm.conf
[main]
dns=systemd-resolved
root@moxa-imoxa1234567:/home/moxa#
```

2. Configure DNSSEC and DNS over TLS

```
root@moxa-imoxa1234567:/home/moxa# cat /etc/systemd/resolved.conf

[Resolve]
# Some examples of DNS servers which may be used for DNS= and FallbackDNS=:
# Cloudflare: 1.1.1.1 1.0.0.1 2606:4700:4700::1111 2606:4700:4700::1001
# Google:     8.8.8.8 8.8.4.4 2001:4860:4860::8888 2001:4860:4860::8844
# Quad9:      9.9.9.9 2620:fe::fe

DNS=1.1.1.1 8.8.8.8 9.9.9.9
DNSSEC=yes
DNSOverTLS=yes
```

3. Restart, then enable the system-resolved service.

```
sudo systemctl restart systemd-resolved
sudo systemctl enable systemd-resolved
```

4. The **systemctl status** output indicates the service is running successfully

```
root@moxa-imoxa1234567:/home/moxa# systemctl status systemd-resolved

• systemd-resolved.service - Network Name Resolution

     Loaded: loaded (/lib/systemd/system/systemd-resolved.service; disabled;
vendor preset: enabled)
     Active: active (running) since Wed 2024-08-14 22:59:39 GMT; 1min 45s
ago
       Docs: man:systemd-resolved.service(8)
             man:org.freedesktop.resolve1(5)
             https://www.freedesktop.org/wiki/Software/systemd/writing-
network-configuration-managers
             https://www.freedesktop.org/wiki/Software/systemd/writing-
resolver-clients
   Main PID: 1504 (systemd-resolve)
     Status: "Processing requests..."
      Tasks: 1 (limit: 4011)
     Memory: 5.1M
        CPU: 287ms
     CGroup: /system.slice/systemd-resolved.service
             mq1504 /lib/systemd/systemd-resolved
```

5. Verify the configuration using the resolvectl

```
root@moxa-imoxa1234567:/home/moxa# resolvectl status
Global
         Protocols: +LLMNR +mDNS +DNSOverTLS DNSSEC=yes/supported
  resolv.conf mode: foreign
Current DNS Server: 1.1.1.1
       DNS Servers: 1.1.1.1 8.8.8.8 9.9.9.9

Link 2 (eth0)
    Current Scopes: DNS LLMNR/IPv4 LLMNR/IPv6
         Protocols: +DefaultRoute +LLMNR -mDNS +DNSOverTLS
DNSSEC=yes/supported
Current DNS Server: 1.1.1.1
       DNS Servers: 1.1.1.1 8.8.8.8 9.9.9.9

Link 3 (eth1)
Current Scopes: none
     Protocols: -DefaultRoute +LLMNR -mDNS +DNSOverTLS DNSSEC=yes/supported

Link 4 (can0)
Current Scopes: none
     Protocols: -DefaultRoute +LLMNR -mDNS +DNSOverTLS DNSSEC=yes/supported

Link 5 (can1)
Current Scopes: none
     Protocols: -DefaultRoute +LLMNR -mDNS +DNSOverTLS DNSSEC=yes/supported

Link 7 (wwan0)
Current Scopes: none
     Protocols: -DefaultRoute +LLMNR -mDNS +DNSOverTLS DNSSEC=yes/supported

Link 8 (dummy0)
Current Scopes: none
     Protocols: -DefaultRoute +LLMNR -mDNS +DNSOverTLS DNSSEC=yes/supported
```

Firewall

nftable is the built-in firewall in Moxa Industrial Linux. Secure model of Moxa Arm-based computer has pre-configured rules to further protect your device from network attacks.

NOTE

Click the following link for detail instructions of nftable usages
https://wiki.nftables.org/wiki-nftables/index.php/Main_Page
https://wiki.nftables.org/wiki-nftables/index.php/Quick_reference-nftables_in_10_minutes

Pre-configured Rule

Below is a summary of nftable rules in **`/etc/nftables.conf`** set by Moxa in Secure model of Moxa Arm-based computer. For Standard model, nftable is not enabled by default.

Rules Set	Location/Parameters
Allowed only ports following port • TCP: SSH (22), HTTPS (443), SNMP TRAP (162) • UDP: NTP (123), DNS (53), SNMP (161), SNMP TRAP (162)	define tcp_port_allow = { ssh, https, 161, 162}; define udp_port_allow = { 53, ntp, 161, 162};
Allow all traffic from loopback interface	iifname "lo" accept
Drop all input traffic except for traffic from allowed ports and icmp (ping)	chain input { ……}
Allow related and established traffic by using conntrack	ct state invalid drop ct state established,related accept

Rules Set	Location/Parameters
Drop all forward traffic	chain forward { ……}
Accept all output traffic	chain output { ……}

```
flush ruleset

define tcp_port_allow = { 22, 443, 161, 162 };
define udp_port_allow = { 53, 123, 161, 162 };

table inet filter {
        # input: drop all traffic
        chain input {
                type filter hook input priority 0; policy drop;

                ct state invalid drop
                ct state established,related accept

                # allow icmp
                icmp type {
                        echo-request,
                        echo-reply,
                        time-exceeded,
                        parameter-problem,
                        destination-unreachable
                } accept

                # allow icmp6
                icmpv6 type {
                        echo-request,
                        echo-reply,
                        time-exceeded,
                        parameter-problem,
                        destination-unreachable,
                        nd-neighbor-solicit,
                        nd-router-advert,
                        nd-neighbor-advert
                } accept

                # accept lo
                iifname "lo" accept

                tcp dport $tcp_port_allow accept
                udp dport $udp_port_allow accept
        }

        # foward: drop all traffic
        chain forward {
                type filter hook forward priority 0; policy drop;
        }

        # output: accept all traffic
        chain output {
                type filter hook output priority 0; policy accept;
        }
}
```

Common nftable Usage

1. List the currently loaded nftable rules **`# nft list ruleset`**
2. Debug and tracing if traffic drops or accepts unexpectedly **`# nft monitor trace`**
   a. Add trace_chain before the existing input chain

```
nft add chain inet filter trace_chain { type filter hook prerouting
priority -1\; }
```

   b. Add nftrace flag

```
nft add rule inet filter trace_chain meta nftrace set 1
```

   c. Monitor trace (you can use another device with ncat tool to test it)dd nftrace flag

```
moxa@moxa-tbbbb1182816:/# sudo nft monitor trace

trace id d51bda11 inet filter trace_chain packet: iif "eth0" ether saddr
d8:5e:d3:a5:7b:29 ether daddr 00:90:e8:a6:37:cb ip saddr 192.168.1.102 ip
daddr 192.168.1.107 ip dscp cs0 ip ecn not-ect ip ttl 128 ip id 36481 ip
protocol tcp ip length 52 tcp sport 1142 tcp dport 53 tcp flags == syn
tcp window 64240 trace id d51bda11 inet filter trace_chain rule meta
nftrace set 1 (verdict continue) trace id d51bda11 inet filter
trace_chain verdict continue trace id d51bda11 inet filter trace_chain
policy accept trace id d51bda11 inet filter input packet: iif "eth0"
ether saddr d8:5e:d3:a5:7b:29 ether daddr 00:90:e8:a6:37:cb ip saddr
192.168.1.102 ip daddr 192.168.1.107 ip dscp cs0 ip ecn not-ect ip ttl
128 ip id 36481 ip protocol tcp ip length 52 tcp sport 1142 tcp dport 53
tcp flags == syn tcp window 64240 trace id d51bda11 inet filter input
verdict continue trace id d51bda11 inet filter input policy drop
```

   d. Once debugging is completed, make sure to remove the debug flag by either method below:
      - ❒ Restart nftable **`# systemctl restart nftables`** or
      - ❒ Reload the configuration again **`# nft -f /etc/nftables.conf`**

Rate Limiting

Rate limiting is a common strategy to prevent network attacks such as DOS, DDOS, and brute force by limiting the network traffic within a specified time. As the suitable rate limit configuration depends heavily on the asset owner's applications, rate limiting is not configured by default in Moxa Industrial Linux.

nftable Rate Limit Usage	Example of Rate Limit Configuration
rate [over] <value> <unit> [burst <value> <unit>]	limit rate 400/minute limit rate 400/hour limit rate over 40/day limit rate over 400/week limit rate over 1023/second burst 10 packets limit rate 1025 kbytes/second limit rate 1023000 mbytes/second limit rate 1025 bytes/second burst 512 bytes limit rate 1025 kbytes/second burst 1023 kbytes limit rate 1025 mbytes/second burst 1025 kbytes limit rate 1025000 mbytes/second burst 1023 mbytes

You can directly add rate limit to existing rule in **`/etc/nftables.conf`**:

Below is an example of limiting TCP and UDP network traffic to 4 packets per second

```
#!/usr/sbin/nft -f

flush ruleset

define tcp_port_allow = { ssh, https };
define udp_port_allow = { 53, ntp };

table inet filter {
        # input: drop all traffic
```

```
        chain input {
                type filter hook input priority 0; policy drop;

                ct state invalid drop
                ct state established,related accept

                # allow icmp
                ip protocol icmp icmp type {
                        echo-request,
                        echo-reply,
                        time-exceeded,
                        parameter-problem,
                        destination-unreachable
                } accept

                # allow icmp6
                ip6 nexthdr icmpv6 icmpv6 type {
                        echo-request,
                        echo-reply,
                        time-exceeded,
                        parameter-problem,
                        destination-unreachable
                } accept

                # accept lo
iifname "lo" accept

                tcp dport $tcp_port_allow limit rate 4/second accept
                udp dport $udp_port_allow limit rate 4/second accept        }
```

Mitigating a NTP Amplification Attack

The default configured NTP servers in Moxa Industrial Linux(MIL) are with NTS support. If you use public NTP servers without NTS support, it is vulnerable to the **NTP amplification attack**, in which the attacker could exploit the public NTP servers to overwhelm Moxa Arm-based computer with UDP traffic. Under such an incident, you can follow the steps to stop the attack:

1. Stop NTP service temporarily with the # systemctl stop systemd-timesyncd command.
2. Block the tainted NTP server by nftables command
   a. Create new firewall table
      ```
      nft add table inet firewall-filter
      ```
   b. Create new chain input in firewall table
      ```
      nft add chain inet firewall-filter input
      ```
   c. Create new chain input in firewall table
      ```
      nft add rule inet firewall-filter input tcp dport { ntp } ip saddr <your
      ip> reject
      ```
   d. Block NTP server IP
      ```
      nft add rule inet firewall-filter input tcp dport { ntp } ip saddr <your
      ip> reject
      ```
   e. Check the rule set
      ```
      nft list ruleset

      ...
      table inet firewall-filter {
              chain input {
                      tcp dport { 123 } ip saddr 10.213.123.55 reject
              }
      }
      ```
3. You can choose to specify another NTP server (modify **/etc/systemd/timesyncd.conf**) or wait for this server to finish troubleshooting

4. Remember to flush the rule after recovery

```
nft delete chain inet firewall-filter input # delete chain
# or
nft delete table inet firewall-filter # delete table
```

Service and Ports

Only activate protocols that you require to use the system. Below is the list for the protocol and port numbers used for all external interfaces. Please refer to Firewall section to modify the list of allowed port if additional port is required.

Protocol	Protocol Type	Port Number
SSH	TCP	22
HTTPS	TCP	443
NTS	UDP	123/4460
DNS	UDP	53
mDNS	UDP	5353

Disable Unused Interface

To enhance cybersecurity by reducing the attack surface, disable any unused interfaces using the Moxa Computer Interface Manager (MCIM)

- Serial console port
- Serial port
- CAN port
- Ethernet port
- External storage (e.g., USB, SD)

Disable Unnecessary Protocols, Services, and Ports

You can use **#ss** to list all the current running processes using with the associated service, protocol, and network port.

```
moxa@moxa-tbbbb1182827:~$ sudo ss -tulpn
Netid       State         Recv-Q       Send-Q              Local Address:Port
Peer Address:Port       Process
tcp         LISTEN        0            128                       0.0.0.0:22
0.0.0.0:*           users:(("sshd",pid=974,fd=3))
tcp         LISTEN        0            128                          [::]:22
[::]:*            users:(("sshd",pid=974,fd=4))
```

You can disable a daemon or service by killing process ID (PID) directly. For example:

```
moxa@moxa-tbbbb1182827:~$ sudo kill 974
```

Or you can just stop and disable the service using **#systemctl**. For example:

```
moxa@moxa-tbbbb1182827:~$ sudo systemctl stop sshd
moxa@moxa-tbbbb1182827:~$ sudo systemctl disable sshd
```

Restrict Unnecessary Protocols, Services, and Ports

1. Protocols:

   Use **nfables** meta to match kind of TCP traffic Matching packet metainformation. Refers to nftables wiki.

2. Services:

   Use **# systemctl list-unit-files** to find unused services and disable them by **systemctl disable <service>**.

3. Ports:

   Use nftables to add accepted ports in whitelist. Refers to the Firewall section for detail instructions.

Services Enabled by Default

Below is the list for the services enabled by default in the secure model of the Moxa Arm-based computers.

Service Name	Description
auditd.service	Security Audit log service
dbus.service	System Message Bus
fail2ban.service	Fail2ban IPS (intrusion prevention software)
getty@tty1.service	Getty on tty1
ifupdown-pre.service	Helper to synchronize boot up for ifupdown
kmod-static-nodes.service	Create list of static device nodes for the current kernel
ModemManager.service	DBus-activated daemon which controls mobile broadband (2G/3G/4G) devices and connections
moxa-connection-manager.service	Moxa Connection Manager (MCM)
moxa-guardian.service	Initializing security configuration for Moxa Industrial Linux
moxa-sys-rdy.service	A service the light up the “READY” or “RDY” when the computer successfully boots up
moxa-system-manager-init.service	Moxa System Manager initialization service
moxa-system-manager.service	Moxa System Manager
MoxaComputerInterfaceManager.service	Moxa Computer Interface Manager
moxa-hostname.service	This service is designed to execute automatically during system startup, setting the hostname to a default unique value in the format moxa-[serial number]. If you prefer to define a custom hostname, you can disable this service by utilizing the 'systemctl disable moxa-hostname.service' command.
Moxa-mdns.service	A device discovery service.
networking.service	Raises or downs the network interfaces
NetworkManager.service	Network Manager
nftables.service	nftable
polkit.service	For controlling system-wide privileges is Moxa Industrial Linux
rsyslog.service	System Logging Service
serial-getty@ttymxc0.service	Serial Getty on ttymxc0al-getty@ttymxc0.service
snmpd.service	Linux service for the Simple Network Management Protocol (SNMP)
ssh.service	SSH Server
sysstat.service	A collection of performance monitoring tools for Linux.
systemd-journal-flush.service	Flush journal to persistent storage
systemd-journald.service	Journal service
systemd-logind.service	User login management
systemd-modules-load.service	Early boot service that loads kernel modules
systemd-random-seed.service	Service that loads an on-disk random seed into the kernel entropy pool during boot and saves it at shutdown
systemd-remount-fs.service	early boot service that applies mount options listed in fstab(5)
systemd-sysctl.service	An early boot service that configures sysctl(8) kernel parameters
systemd-sysusers.service	Creates system users and groups, based on the file format and location specified in sysusers.d(5)
systemd-timesyncd.service	System service that synchronizes the local system clock with a remote Network Time Protocol (NTP) server
systemd-tmpfiles-setup-dev.service	Create Static Device Nodes in /dev
systemd-tmpfiles-setup.service	Create Volatile Files and Directories
systemd-udev-trigger.service	Coldplug all udev devicesd-udev-trigger.service
systemd-udevd.service	Listens to kernel uevents
systemd-update-utmp.service	Service that writes SysV runlevel changes to utmp and wtmp, as well as the audit logs
systemd-user-sessions.service	a service that controls user logins through pam_nologin(8)
user-runtime-dir@1000.service	Default user
user@1000.service	Default user
vnstat.service	network traffic monitor
watchdog.service	Watchdog service
wpa_supplicant.service	WPA supplicant

Managing Resources

Setting The Process Priority

A process can be manually adjusted to increase or decrease its priority. Use the **top** or **ps** commands to find out the process priority.

```
moxa@moxa-tbbbb1182827:/# sudo top
top - 22:08:43 up 6 min,  1 user,  load average: 0.01, 0.04, 0.01
Tasks: 105 total,   1 running, 104 sleeping,   0 stopped,   0 zombie
%Cpu(s):  0.2 us,  0.8 sy,  0.0 ni, 98.8 id,  0.1 wa,  0.0 hi,  0.0 si,  0.0 st
KiB Mem :  2068192 total,  1874520 free,    57416 used,   136256 buff/cache
KiB Swap:        0 total,        0 free,        0 used.  1799712 avail Mem

PID USER      PR  NI    VIRT    RES    SHR S %CPU %MEM     TIME+ COMMAND
  1 root      20   0    9492   6220   5236 S  0.0  0.3   0:00.98 systemd
  2 root      20   0       0      0      0 S  0.0  0.0   0:00.00 kthreadd
  3 root      20   0       0      0      0 S  0.0  0.0   0:00.01 ksoftirqd/0
  4 root      20   0       0      0      0 S  0.0  0.0   0:00.02 kworker/0:0
  5 root       0 -20       0      0      0 S  0.0  0.0   0:00.00 kworker/0:0H
  6 root      20   0       0      0      0 S  0.0  0.0   0:00.01 kworker/u2:0
  7 root      20   0       0      0      0 S  0.0  0.0   0:00.02 rcu_sched
 ...
```

You can also use the **ps** command with the **-l**, long list option to find out the priority of the process.

```
moxa@moxa-tbbbb1182827:/# sudo ps -efl
F S UID        PID  PPID  C PRI  NI ADDR SZ WCHAN  STIME TTY  TIME CMD
4 S root         1     0  0  80   0 -  2373 ep_pol 22:02 ?    00:00:01
/sbin/init
1 S root         2     0  0  80   0 -     0 kthrea 22:02 ?    00:00:00
[kthrreadd]
1 S root         3     2  0  80   0 -     0 smpboo 22:02 ?    00:00:00
[ksoftirqd/0]
1 S root         5     2  0  60 -20 -     0 worker 22:02 ?    00:00:00
[kworker/0:0H]
1 S root         6     2  0  80   0 -     0 worker 22:02 ?    00:00:00
[kworker/u2:0]
1 S root         7     2  0  80   0 -     0 rcu_gp 22:02 ?    00:00:00
[rcu_sched]
1 S root         8     2  0  80   0 -     0 rcu_gp 22:02 ?    00:00:00
[rcu__bh]
...
```

The PRI (Priority) or NI (Nice) is the priority of the process. The PRI is adjusted by kernel automatically. The NI can have a value in the range -20 to 19. A smaller value means that the program could use more CPU resources.

The nice utility can be given a specific nice value while running a program. This example shows how to launch the **tar** utility with the nice value 20.

```
moxa@moxa-tbbbb1182827:/# sudo nice -n 20 tar -czvf TheCompressFile.tar /src1
/src2 ...
OR
moxa@moxa-tbbbb1182827:/# sudo nice -adjustment 20 tar -czvf
TheCompressFile.tar /src1 /src2 ...
```

You can use the **renice** utility to dynamically adjust the nice value of a program. This example uses renice to adjust the auditd, PID 639, with highest priority as 20.

```
moxa@moxa-tbbbb1182827:/# sudo renice -n 20 -p 639
moxa@moxa-tbbbb1182827:/# sudo ps -efl|grep auditd
1 S root       639     1  0  75  -20 -  1519 poll_s 22:02 ?        00:00:00
/sbin/auditd -n
...
```

NOTE

Click the following link for more information on usages of nice and renice
https://manpages.debian.org/bullseye/coreutils/nice.1.en.html
https://manpages.debian.org/bullseye/bsdutils/renice.1.en.html

Setting the Process I/O Scheduling Class and Priority

The **ionice** command can adjust the priority of the program using I/O. The class and priority are adjustable for a process.

-c class	0: none 1: realtime 2: best-effort 3: idle
-n classdata	The realtime and best-effort can set from 0 to 7. A smaller value means the program has a higher priority.
-p PID	Process ID

```
moxa@moxa-tbbbb1182827:/# sudo ps -l
F S   UID   PID  PPID  C PRI  NI ADDR SZ WCHAN  TTY          TIME CMD
4 S     0   895   886  0  80   0 -  1794 wait   pts/0    00:00:00 bash
4 S     0  1099   895  0  80   0 -  1659 poll_s pts/0    00:00:00 sudo
4 R     0  1100  1099  0  80   0 -  1850 -      pts/0    00:00:00 ps
moxa@moxa-tbbbb1182827:/# sudo ionice -c 2 -n 0 -p 895
moxa@moxa-tbbbb1182827:/# sudo ionice  -p 895
best-effort: prio 0
```

NOTE

Click the following link for more information on usages of ionice
https://manpages.debian.org/bullseye/util-linux/ionice.1.en.html

Limiting the CPU Usage of a Process Using cpulimit

cpulimit is a simple program that attempts to limit the CPU usage of a process (expressed in percentage, not in CPU time). This is useful to control batch jobs, when you don't want them to eat too much CPU.

This example, use the cpulimit to limit the usage of sshd process CPU limit percentage to 25% in background. The -p is the process ID. The -e switch take the executable program file name. The -l is the CPU limit percentage. The option, -b, to run cpulimit in the background, freeing up the terminal.

```
moxa@moxa-tbbbb1182827:/# sudo cpulimit -p 895 -l 25 -b
```

NOTE

Click the following link for more information on usages of cpulimit
https://manpages.debian.org/bullseye/cpulimit/cpulimit.1.en.html

Limiting the Rate

Refer to the Chapter 8 Security Firewall Rate Limiting to customize the network limitation of the firewall configuration.

Audit Log

In this section, we will introduce the audit event log design in Moxa Industrial Linux and bootloader, including the security event monitored and recommended response and approach for audit processing failures.

Linux Audit log

Auditd is being used in Moxa Industrial Linux for system administrators to monitor detailed information about system operation. It provides a way to track and record security-relevant information on the system.

1. Log partition size:

Computer Series	Log partition size
UC-8200	256 MB
UC-1222A/2222A	MIL 3.1: 256 MB MIL 3.3 and later: 1024 MB
UC-3400A	1024 MB
UC-4400A	1024 MB
V1200	1024 MB

2. Log partition applies Linux Unified Key Setup (LUKS) encryption and restrict non root user from access
3. Logs are stored under **`/var/log/audit/`** and the log format follows **`auditd`** standard.
   - Below is a reference of where to find the commonly used log data fields in audit log

Common Log Data Fields	Data Fields in auditd log
timestamp	msg=audit(TIMESTAMP)
source	proctitle, comm, exec, uid, gid, etc.
category	key
type	type
eventID	pid, ppid

4. Audit log records are automatically rotated daily and up to 14 archived logs are kept at a time. When log rotates, the oldest archive will be deleted if 14 archived logs already exist.
   - Audit log rotation rule can be modified in **`/etc/logrotate.d/auditd`**
5. The log timestamp is the local system time which synchronized with a remote Network Time Protocol (NTP) server.
   - For time synchronization status and configuration, refers to timedatectl(1)

NOTE

Click the following link for more information on usages of auditd and log search
https://manpages.debian.org/bullseye/auditd/auditd.8.en.html
https://manpages.debian.org/bullseye/auditd/ausearch.8.en.html

Below are the security events that Moxa Industrial Linux is pre-configured to monitor in Secure model of Moxa Arm-based computer

Event Category	Event Logged	File or Directory to Monitor	key used for ausearch
Access control	Users logins, logouts, system events, etc.	/var/run/utmp /var/run/btmp /var/run/wtmp	session
Backup and restore	Use of Moxa System Manager tool	/sbin/mx-system-mgmt	system_mgmt
Control System	Shutdown system	/bin/systemctl	control_system
	Power off system		
	Reboot system		
	Halt system		
	Use of APT package management system	/usr/bin/apt	system_package
	Use of aptitude tool	/usr/bin/aptitude	system_package
	Use of add-apt-repository tool	/usr/bin/apt-add-repository	system_package
	Use of apt-get tool	/usr/bin/apt-get	system_package
	Use of dpkg package manager tool	/usr/bin/dpkg	system_package
Security configurations	Add user configuration change	/etc/adduser.conf	adduser
	AIDE configuration and database change	/etc/aide	aide
	Audit configuration and log change	/etc/audit /var/log/audit	auditconfig auditlog
	Chrony configuration change	/etc/chrony	chrony
	Fail2ban configuration change and log change	/etc/fail2ban /var/log/fail2ban.log	fail2ban fail2ban-log
	Fail lock configuration change	/etc/security/faillock.conf	faillock
	Login policy change	/etc/login.defs	login
	Log rotate configuration change	/etc/logrotate.conf /etc/logrotate.d	logrotate
	nftable configuration change	/etc/nftables.conf	nftables
	Moxa Computer Interface Management configuration change	/etc/moxa/ MoxaComputerInterfaceManager	mcim
	Moxa Connection Manger configuration change	/etc/moxa/MoxaConnectionManager	mcm
	Moxa Guardian configuration and log change	/etc/moxa/moxa-guardian /var/lib/moxa-guardian	moxa-guardian moxa-guardian-registry
	Password policy change	/etc/pam.d	pam
	Linux system wide environment configuration change	/etc/profile /etc/profile.d	profile
	Password rule change	pwquality.conf pwquality.conf.d	pwquality
	Rsyslog configuration change	/etc/rsyslog.conf /etc/rsyslog.d	rsyslog
	SSH (Secure Shell) configuration change	/etc/ssh/sshd_config /etc/ssh/sshd_config.d	sshd
	Sudo configuration change	/etc/sudoers	sudo
	Kernel parameters change	/etc/sysctl.conf /etc/sysctl.conf.d	sysctl

Bootloader Audit Log

1. Log is stored in SPI flash with **1MB** storage size
2. Log can be viewed via **(2) Advance Setting > (4) View Bootloader Log** in Bootloader menu
3. Maximum number of logs is 4,000 records, where the oldest log will be overwritten when the maximum capacity is reached.
4. The time stamp of the log read from the local Real-time Clock (RTC) which is synchronize with Network Time Protocol (NTP) server.
5. Log format and log events are described below

Audit Log Structure

Header	Explanation	Possible Values
Time	Time stamp of the device	Format: [YYYY-MM-DDThh:mm:ss] For example: [2022-06-03T15:54:38]
User	Identifies the authenticated user	Admin
Category	Event category	• System • Bootcfg (refers to boot configuration) • Install • Security
Event ID	ID of a logged event	1 ~ 223
Event Message	Description of the logged event	See below table for the list of events

Audit Events

Category	Event ID	Event Type	Event Message
System	1	Info	All bootloader configuration set to default
System	2	Info	Exit bootloader and reboot system
System	3	Info	Exit bootloader and boot to Linux
bootcfg	4	Info	Set boot configuration to default ok
bootcfg		Warning	Set boot configuration to default fail
bootcfg	5	Info	Set boot from SD/USB/eMMC ok
bootcfg		Warning	Set boot from SD/USB/eMMC fail
bootcfg	6	Warning	USB is not available on this device
bootcfg	7	Info	Bootarg and bootcmd changed
Install	8	Info	Install system image from TFTP ok
Install		Warning	Destination net unreachable
Install		Warning	Hash/Signature file not find
Install		Warning	System image file error
Install		Warning	File size is too large
Install		Warning	Upgrade system image fail
Install		Alert	System image authenticity check fail
Install	9	Info	Install system image from SD ok
Install		Warning	SD/USB/eMMC device not find
Install		Warning	Hash/Signature file not find
Install		Warning	System image file error
Install		Warning	File size is too large
Install		Warning	Upgrade system image fail
Install		Alert	System image authenticity check fail
Secure	10	Info	Install system image from USB ok
Secure		Warning	SD/USB/eMMC device not find
Secure		Warning	Hash/Signature file not find
Secure		Warning	System image file error
Secure		Warning	File size is too large
Secure		Warning	Upgrade system image fail
Secure		Alert	System image authenticity check fail
Secure	11	Info	TFTP setting changed
Secure	12	Info	Login success
Secure		Warning	login fail
Secure	13	Alert	Boot failure due to system image integrity or authenticity check fail

Category	Event ID	Event Type	Event Message
Secure	14	Info	Admin password disabled
Secure		Info	Admin password enabled
Secure	15	Info	Admin password set to default
Secure	16	Info	Admin password changed
Secure	17	Info	Admin password policy changed
Secure	18	Info	Advance settings set to default
Secure	19	Info	Auto reboot threshold changed
Secure	20	Info	Login message changed
Secure	21	Info	Invalid Login Attempts changed
Secure	22	Info	Clear TPM ok
Secure		Warning	Clear TPM fail
audit	23	Info	View bootloader log ok

Audit Failure Response

The section is a guideline for protection of critical system functions in case of audit processing failure. Without appropriate response to audit processing failure, an attacker's activities can go unnoticed, and evidence of whether the attack led to a breach can be inconclusive. Following are some common approaches:

1. **Log rotation**

   Log rotation is enabled by default in Moxa Arm-based computer to prevent audit storage capacity full. Refers to **Linux Audit Log** and **Bootloader Audit log** sections for details.

   In Linux, configure the logrotate to manage disk space usage effectively and prevent running out of storage. The logrotate configuration file is at /etc/logrotate.config and all the files in /etc/logrotate.d/* to rotate the log file.

   This example we configure /etc/logrotate.d/rsyslog to rotate /var/log/syslog while it overs the size 2M with only 3 rotation.

```
/var/log/syslog

  {
    {
      rotate 3
      maxsize 2M
      ...
    }
  }
```

2. **Saving the logs in external storage**
   - For auditd, change the file path of parameter **log_file** in **/etc/audit/auditd.conf**
   - For rsyslog, change the default file path **/var/log/** in **/etc/rsyslog.conf** to external storage
3. **Use a centralized log Server**

   Use a centralized log managements system to collect and store the logs from Log from multiple devices. Refers to How to Set Up Centralized Logging on Linux with Rsyslog
4. **Assign appropriate action when audit storage space is full, or error occurs**

   You can configure **space_left** and **space_left_action** parameters in **/etc/audit/auditd.conf** to specify the remaining space (in megabytes or %) for low disk alert and what action to take. The actions are ignore, syslog, rotate, exec, suspend, single, and halt.

   In example below, warning email will be sent to email account specified in **action_mail_acct** parameter when the free space in the filesystem containing log files drop below 75 megabytes

```
space_left = 75
space_left_action = email
```

   Configure **disk_full_action** and **disk_error_action** in **/etc/audit/auditd.conf** to specify what actions to take when audit storage disk got error or full. The actions are ignore, syslog, rotate (for disk full only), exec, suspend, single, and halt.

   Refers to auditd(8) for detail explanation of each action and parameters.

Security Diagnosis Tool (Moxa Guardian)

The secure models of Moxa's Arm-based computers come with built-in security compliant with the IEC 62443-4-2 Security Level 2 requirements. However, on many occasions, the default security settings could unintentionally change, especially when customizing the computer, making them not adhere to the standard.

Moxa Guardian is a security diagnosis tool that provides an overview of the gap between your current security configurations and the IEC 62443-4-2 Security Level 2 standards. You can also use the tool to restore the security configurations to the default out-of-box secured configurations.

Use the **# mx-guardian** command to display the menu page.

```
Moxa Guardian is a cli tool allows users to operate security configs

Moxa Guardian is a CLI security diagnosis tool that gives you an overview of
the gap between the current security configurations against the IEC 62443-4-2
Security Level 2 host device requirement and the Moxa recommended security
configurations.

Usage:
  mx-guardian [command]

Available Commands:
  diagnose   Diagnose security settings and output report
  help        Help about any command
  set         Apply a pre-defined security profile
  version     Show Moxa Guardian version and build info

Flags:
  -f, --force      force mode
  -h, --help       help for mx-guardian
      --no-color   disable color
  -q, --quiet      quiet mode (imply force)
  -v, --verbose    verbose mode
      --version    get version

Use "mx-guardian [command] --help" for more information about a command.
```

ATTENTION

As the Moxa computer is an open platform that allows users to install any software they desire, Moxa Guardian's diagnosis tool only compares the current configurations against the default out-of-box IEC 62443-4-2 compliance configurations. For example, if additional protocols are installed, Moxa Guardian will not diagnose such protocols' communication integrity and authenticity capabilities. It is the responsibility of the user to follow the hardening guidelines and the IEC 62443 standard to meet the security requirements.

Diagnosing Issues in the Current Security Configuration

Use **# mx-guardian diagnose <flags>** to initiate a diagnosis of the current security configurations against the default out-of-box secured configuration, which include all IEC 62443-4-2 security level 2 compliance configurations and also additional Moxa recommended security setting not covered in IEC 62443 standard. The diagnosed result are shown in the sequential orders of IEC 62443-4-2 requirement (CR 1.1 to CR 7.8), followed by Moxa's recommended security settings.

Flags	Description
-d or –detail	Show details including the reason and guideline for the failed requirements
-h or –help	Print the help menu for diagnose command
-o or –output <target filepath>	Output the diagnose result to a file

The diagnosis result could be one of the following:

- **PASS:** The device's security configuration meets the IEC 62443-4-2 security level 2 standard or Moxa recommended setting.
- **FAIL:** The device's security configuration fails to meet the IEC 62443-4-2 security level 2 standard or Moxa recommended setting.
- **INFO:** The device's security configuration meet the IEC 62443-4-2 security level 2 standard but additional configuration can be applied if suitable.

An example of Moxa Guardian's diagnosis output is given below:

```
root@moxa-tbbbb1182816:/home/moxa# mx-guardian diagnose -d

INFO[2022-11-14T12:19:33Z] start diagnosing requirement
INFO[2022-11-14T12:19:33Z] diagnose requirement all                    detail=true

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

As the Moxa computer is an open platform that allows users to install any software
they desire, Moxa Guardian's diagnosis tool only compares the current configurations
against the default out-of-box IEC-62443-4-2 compliance configurations

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


CR 1.1: Human user identification and authentication
--------------------------------------------------------------------------------------------------------
[+] Precondition
    > Package
      - openssh-server                                                                          [PASS]
      - openssh-client                                                                          [PASS]
      - libpam-modules                                                                          [PASS]

[+] Check
    > Option: SSHD:UsePAM                                                                       [PASS]
      - info:  Check UsePAM is set to yes in sshd
      - guide: Modifiy or add "UsePAM yes" in /etc/ssh/sshd_config or /etc/sshd/sshd_config.d/*.conf

CR 1.2: Software process and device identification and authentication
--------------------------------------------------------------------------------------------------------
[+] Precondition
    > Package
      - openssh-server                                                                          [PASS]
      - libpam-modules                                                                          [PASS]

[+] Check
    > Option: SSHD:UsePAM                                                                       [PASS]
      - info:  Check UsePAM is set to yes in sshd
      - guide: Modify or add "UsePAM yes" in /etc/ssh/sshd_config or /etc/sshd/sshd_config.d/*.conf
    > Option: SSHD:PubKeyAuthentication                                                         [PASS]
      - info:  Check PubkeyAuthentication is set to yes in sshd
      - guide: Modify or add "PubkeyAuthentication yes" in /etc/ssh/sshd_config or
               /etc/sshd/sshd_config.d/*.conf

CR 1.3: Account management
--------------------------------------------------------------------------------------------------------
[+] Precondition
    > Package
      - passwd                                                                                  [PASS]

CR 1.4: Identifier management
--------------------------------------------------------------------------------------------------------
[+] Precondition
    > Package
      - base-passwd                                                                             [PASS]
      - passwd                                                                                  [PASS]

CR 1.5: Authenticator management
```

Restoring the Security Configuration to the Default

Use **# mx-guardian set <command> <flags>** to restore the Moxa Arm-based security configuration to the to the default out-of-box IEC 62443-4-2 compliance secured configurations.

Command	Description
secure	Restore the Moxa Arm-based configuration to a pre-defined security profile

Flags	Description
-d or –detail	Show details including the reason and guideline for the failed requirements
-h or –help	Print the help menu
-m or --mode <string>	The <string> parameter support 2 values (m1 or m2) Description of each mode is given below : **M1:** Apply only the IEC 62443-4-2 security level 2 required settings **M2:** Apply both M1 and Moxa recommended settings *Note : M2 is the default out-of-box security setting*

An example of restoring the computer's security profile to M2 (IEC 62443-4-2 security level 2 and Moxa recommended settings) is give below:

```
moxa@moxa-tbzkb1090923:~$ sudo mx-guardian set secure -m m2
INFO[2022-11-10T05:53:51Z] start setting secure command
INFO[2022-11-10T05:53:51Z] apply all changes with
force=false mode="IEC62443-4-2 and MOXA suggested settings" quiet=false
INFO[2022-11-10T05:53:51Z] no changes
file=/etc/adduser.conf
INFO[2022-11-10T05:53:51Z] no changes
file=/etc/audit/auditd.conf
INFO[2022-11-10T05:53:51Z] no changes
file=/etc/profile.d/99-moxa-profile.conf
INFO[2022-11-10T05:53:51Z] no changes
file=/etc/security/faillock.conf
INFO[2022-11-10T05:53:51Z] no changes
file=/etc/security/pwquality.conf.d/99-moxa-pwquality.conf
INFO[2022-11-10T05:53:51Z] no changes
file=/etc/login.defs
INFO[2022-11-10T05:53:51Z] no changes
file=/etc/logrotate.d/00-moxa-logrotate.conf
INFO[2022-11-10T05:53:51Z] no changes
file=/etc/ssh/sshd_config.d/00-moxa-sshd.conf
INFO[2022-11-10T05:53:51Z] no changes
file=/etc/sysctl.d/99-moxa-sysctl.conf
INFO[2022-11-10T05:53:51Z] no changes
file=/etc/rsyslog.d/99-moxa-rsyslog.conf

Attention : you must reboot your computer for the changes to take effect
```

ATTENTION

You must reboot your computer for the changes to take effect.

Compliance With EN 18031:1 EU RED Cybersecurity Certification

For standard models, configuration modifications are required when upgrading from MIL 3.0, 3.1, 3.2, or 3.3 to MIL 3.4 or later.

You can use the command **# mx-guardian red <command> <flags>** to diagnose the computer's compliance with RED requirements and configure it to meet RED compliance standards.

```
Use "mx-guardian red [command] --help" for more information about a command.
root@moxa-imoxa1000038:/# mx-guardian red -h

Your device is configured RED profile by default.
Use 'mx-guardian red diagnose' to check RED profile is changed or not.

 Usage:
  mx-guardian red [command]

 Available Commands:
  diagnose  Diagnose RED settings and output report
  set       Apply RED profile

 Flags:
  -h, --help   help for red

 Global Flags:
  -f, --force     force mode
  -q, --quiet     quiet mode (imply force)
  -v, --verbose   verbose mode

Use "mx-guardian red [command] --help" for more information about a command.
root@moxa-imoxa1000038:/#
```

When you run **# mx-guardian red set**, the following configurations are applied to bring your computer into compliance with RED requirements.

Configure file	Description
/etc/ssh/sshd_config.d/00-moxa-sshd.conf	Set sshd **Cipher** to **chacha20-poly1305@openssh.com,aes128-ctr,aes192-ctr,aes256-ctr,aes128-gcm@openssh.com,aes256-gcm@openssh.com**
	Set sshd **kexalgorithm** to **curve25519-sha256,curve25519-sha256@libssh.org,ecdh-sha2-nistp256,ecdh-sha2-nistp384,ecdh-sha2-nistp521,diffie-hellman-group-exchange-sha256,diffie-hellman-group16-sha512,diffie-hellman-group18-sha512,diffie-hellman-group14-sha256**
	Set sshd **MACs** to **hmac-sha2-256-etm@openssh.com,hmac-sha2-512-etm@openssh.com,hmac-sha2-256,hmac-sha2-512**
	Set sshd **MaxAuthTries** to **5**
	Set sshd **PubkeyAuthentication** to **yes**
	Set sshd **usePAM** to **yes**
	Set sshd **LoginGraceTime** to **60**
/etc/security/faillock.conf	Set **deny=5**
	Enable **even_deny_root**
	Set **fail_interval=60**
	Set **root_unlock_time=300**
	Set **unlock_time=300**
/etc/snmp/snmpd.conf	Remove **rocommunity**
	If createUser exists, disable **MD5|SHA-224|md5|sha-224|DES|des**
/etc/security/pwquality.conf.d/99-moxa-pwquality.conf	Set **dictcheck=1**
	Enable **enforce_for_root**

Configure file	Description
/etc/pam.d/common-password	Enable **libpam-pwquality.so**
/etc/login.defs	Set **LOGIN_RETRIES=5**
	Set **LOGIN_TIMEOUT=60**

9. Security Hardening Guide

In this chapter, we will provide guidance on how to deploy and operate Secure model of Moxa Arm-based computer in a secure manner.

Defense-in-depth Strategy

Security Layer	Security Measures	Threat mitigated/handled	Responsibility
Policy and procedure	Establish policies and procedures to guide employee on their role and responsibilities to for safe use of security sensitive assets. Refers to Operation and Maintenance section for some recommendations	Vulnerabilities created due to employee lack of security policies and procedures awareness	Asset owner (Essential)
		Malicious code attack that could create or exploit system vulnerabilities (Threat ID #6)	
Perimeter Security	Use LTE service provide with Carrier Grade NAT (CGNAT) and firewall	Unauthorized and malicious communications from untrusted network	Asset owner (Essential)
	Perimeter firewall	Unauthorized and malicious communications from untrusted network	Asset owner (Essential)
	Physical security (Refers to section Physical Installation)	Physical modification, manipulation, theft, removal, or destruction of asset	
Network Security	Network IDS/IPS	Network attacks from various sources such as port scanning, DDOS, etc.	Asset owner (Recommended)
	VPN	Man-in-the-middle attacks that allow hackers to intercept and manipulate network traffic (Threat ID #4)	
Endpoint Security	End point Firewall (nftable)	Unauthorized and malicious communications from untrusted network (Threat ID #2, Threat ID #5)	Provided by Moxa Arm-based Computer
	Brute-force attacks IPS (fail2ban)	Trial and error attack attempting to crack login credentials (Threat ID #3)	
	Automatic network Connection failover (Refers to MCM failover configuration)	Radio jamming attack (Threat ID #1)	
	Patch management	Vulnerabilities from outdated software could expose to security breach.	
	Secure transmission protocol	Man-in-the-middle attacks that allow hackers to intercept and manipulate network traffic (Threat ID #4)	Asset owner / Moxa Arm-based Computer (Essential)
	Audit processing failure response	Audit processing failure without appropriate response results in the attacker's activities can go unnoticed, and evidence of whether the attack led to a breach can be inconclusive (Threat ID #7)	
Application Security	IEC 62443-4-1 compliant secure design, implementation, validation, and defect management process	Potential vulnerabilities generated from the development and testing process that doesn't follow security best practices.	Provided by Moxa Arm-based Computer

Security Layer	Security Measures	Threat mitigated/handled	Responsibility
Data Security	Host Intrusion Detection System (AIDE)	Unexpected changes to important files that could potentially lead to security breach.	Provided by Moxa Arm-based Computer
	Access control and login policy including limit invalid login attempts, automatic session termination and login banner	Unauthorized operation to Moxa Arm-based computer that could lead to system confidentiality and integrity breach or availability attack.	
	Disk encryption	Access to confidential data in storage without authorization.	
	Secure boot	Tampering of bootloader, OS kernel and rootFS.	

Table 9.1 – Defense-in-Depth Strategy

*Essential: Security measure that must be taken by asset owner to ensure secure use of Moxa Arm-based computer *Recommended: Security measures that need to be taken by the asset owner if the threats apply.

Potential Threats and Corresponding Security Measures

Below is a list of potential security threats that can harm Moxa Arm-based computers and the corresponding security measures that need to be taken by the **asset owner** if the threats apply.

Threat ID	Threat mitigated/handled	Security Measures
1	Radio jamming attack resulting in Wi-Fi and cellular connection DOS	• For Moxa Arm-based computer with both Wi-Fi and cellular interface, configure connection failover to use backup connection when primary connection is attacked by radio jamming • Extend the perimeter of physical security to reduce the impact from radio jamming attack
2	Network data flow through ethernet, Wi-Fi, cellular interface could be potentially interrupted, crashed or stopped by DOS attack	• Setup network monitoring tool to detect abnormal traffic • Configure rate limiting to limit the network traffic
3	SSH server could be potentially interrupted, crashed or stopped by DOS attack	1. Following parameters are set in SSH server configuration file by Moxa as countermeasure. ➢ MaxSessions: set to 6 to protect a system from denial of service due to a large number of concurrent sessions ➢ MaxStartups: set to 6:30:60 to protect a system from denial of service due to a large number of pending authentication connection attempts 2. Fail2ban is pre-installed and running in Moxa Arm-based computer to automatically ban malicious IP
4	Data flowing across ethernet may be sniffed by an attacker	1. Make sure secure protocol with encryption and authentication are used for data transmission (e.g., SSHv2, HTTPS) 2. Install and use VPN for secure data transmission
5	DOS attack from untrusted NTP server when Moxa Arm-based computer attempt to synchronize time	If a public NTP server without NTS support is used, it is vulnerable to an NTP amplification attack in which the attacker could exploit public NTP servers to overwhelm Moxa Arm-based computer with UDP traffic; therefore, refers to Mitigate NTP Amplification Attack to mitigate it.
6	Data read from USB or SD card could be spoofed	1. Use sha256 or other checksums tools to check the integrity of the file before installing or transferring to device. If the file is Debian package (.deb), refers to "How to manually check for package's integrity" to validate. 2. Scan the file with Clamav before installing or transferring it to the device 3. Use OpenSSL to verify the signature of the file before installing or transferring to the device.
7	Insufficient auditing storage causes logs to rotate frequently	Store logs in external storage or use a centralized log management system to collect and store the logs from multiple devices. Refers to How to Set Up Centralized Logging on Linux with Rsyslog

Threat ID	Threat mitigated/handled	Security Measures
8	If the Ethernet connection between the NPort device and the Moxa computer crosses an internet boundary without proper security measures, data transmission is vulnerable to interception, tampering, or unauthorized access. This risk is heightened if the NPort 5000 series is used, as it does not support SSL-enabled RealTTY, leaving communication unencrypted.	For the NPort 6000 series, use SSL-enabled RealTTY drivers to encrypt communication and ensure secure data transfer. For the NPort 5000 series, implement a site-to-site VPN connection or a private network to encrypt data and protect it from interception, tampering, and unauthorized access.
9	If the Ethernet connection between the ioLogik and the Moxa computer crosses an internet boundary without adequate security measures, the data transmitted is at risk of being intercepted, tampered with, or accessed by unauthorized parties, potentially leading to data breaches or system compromises.	Implement a site-to-site VPN connection or a private network to encrypt the data and safeguard it against interception, tampering, and unauthorized access, ensuring secure communication between the ioLogik and the Moxa computer.
10	The NTP (Chrony) service may lack sufficient input validation mechanisms, potentially allowing malicious or malformed input to be processed. This vulnerability could be exploited to disrupt time synchronization, inject inaccurate timestamps, or compromise system operations relying on precise timing. Such risks may lead to degraded performance, incorrect logging, or cascading failures in dependent systems and applications.	Refer to Enhance DNS Security **DNSSEC:** • Ensures that the DNS responses used by NTP to resolve NTP server domain names are authentic and have not been tampered with. • Prevents attackers from redirecting NTP clients to malicious servers via DNS spoofing. **DNS-over-TLS (DoT):** • Encrypts DNS queries to prevent interception and eavesdropping. • Protects against Man-in-the-Middle (MitM) attacks that could manipulate DNS responses

Installation

Physical Installation

1. Secure model of Moxa Arm-based computer MUST be used to ensure safe use. Refer to Secure and Standard Model for details of model difference.
2. The secure model of Moxa Arm-based computer MUST be protected by physical security that can include CCTV surveillance, security guards, protective barriers, locks, access control, perimeter intrusion detection, etc. The proper form of physical security should apply depending on the environment and the physical attack risk level.
3. Moxa Arm-based computer has anti-tamper labels on the enclosures. This allows the administrator to tell whether the device has been tampered with.
4. Moxa Arm-based computer uses security screw on the enclosures as physical tamper resistance measure to increase the difficulty of probing the product internals in case of physical security breach.
5. Moxa Arm-based computer MUST not be used to **control** the operation of mission-critical IACS component, where failure to maintain control of such device could result in threat to human, safety, environment or massive financial lost.

Environment Requirement

1. If Moxa Arm-based computer connects to untrust network (e.g., Internet) via ethernet or Wi-Fi, it MUST NOT directly connected to the untrust network, which means a firewall must be set up between ethernet and Wi-Fi connection from Moxa Arm-based computer and the untrust network.
2. For security-critical applications, we strongly recommend using a private APN for cellular networks.

Access Control

1. The default user account **Moxa** of Linux belongs to the sudo group. Before deploying Moxa Arm-based computer after development, you must disable this default account and create new account(s) following the least privilege principle, granting only the necessary access right and permission for the intended operation.
2. Each account should be assigned the correct privileges. Moxa Industrial Linux uses Discretionary Access Control (DAC) based on Access Control Lists (ACLs) to manage permissions and privileges. Refers to Permissions and Privileges Control for details.
3. The default password policy requires the password to be at least 8 characters in length. We strongly recommend keeping the default setting, or you can reduce the password length by adding additional complexity rules to the password, such as special character or numeric character enforcement. Refers to instructions to configure the policy for Linux and Bootloader, respectively.
4. Update user passwords on a timely manner. For administrator, we recommend refreshing password at least every 3 months.
5. Bootloader configuration menu comes with a single administrator account shared by all users. Asset owner MUST have access and identity records of the personnel who accessed the bootloader to ensure non-repudiation in case of security breach incidents.
6. Below is a list of all services in Moxa Arm-based computer uses to connect with external processes and components.

Service	Protocol	Interfaces	Owner (uid/gid)	Authorization Enforcement
SSH server	SSH	Ethernet, cellular, Wi-Fi	root/root	Yes
SFTP server	SSH	Ethernet, cellular, Wi-Fi	root/root	Yes
SCP server	SSH	Ethernet, cellular, Wi-Fi	root/root	Yes
Serial Getty service	RS-232	Serial console port	root/root	Yes
APT client	HTTPS	Ethernet, cellular, Wi-Fi	root/root	Yes
NTP client (NTS support)	TLS/SSL, NTP	Ethernet, cellular, Wi-Fi	root/root	Yes
Device Discovery	mDNS	Ethernet	Root/root	Yes

Security Configuration Check

The secure models of Moxa's Arm-based computers are secure-by-default and are compliant with the IEC 62443-4-2 SL2 requirements. However, on many occasions, the default security settings could unintentionally change, especially when customizing the computer, making them not adhere to the standard.

Moxa Guardian is a security diagnosis tool that gives you an overview of the gap between the current security configurations and the IEC 62443-4-2 Security Level 2 standards. Make sure you run the security diagnosis before deploying the product. Refer to Security Diagnosis Tool section for details usage of Moxa Guardian.

Operation

Administrator

1. **Disable default account**

   Use the **passwd** command to lock the default user account so that the **moxa** user cannot log in. Make sure to create a new account before disable the default account.

   ```
   moxa@moxa-tbzkb1090923:# sudo passwd -l moxa
   ```

2. **Disabled interfaces that are not in use**

   The interfaces that are not in use should be deactivated. Please refer to Disabled Unused Interface for detailed instructions.

3. **Periodically regenerate the SSH server key**

   Periodically regenerate the SSH server key in order to secure your system in case the key is compromised. Please refer to Rekey SSH

4. **Trusted administrator**

   Make sure only trusted and reliable persons are registered in the sudo groups for root privilege.

5. **Audit failure response**

   Refer to Audit Failure Response Guideline to protection of critical system functions in case of audit processing failure.

6. **System integrity validation**

   - Frequently run system integrity check to protect your system against malware, viruses and detect unauthorized activities. Refers to Intrusion Detection System for the utility that come with Moxa Arm-based computer.
   - We recommend you reset Moxa Arm-based computer to factory default upon receiving it to avoid the risk of potential software tampering before the computer reaches your hand.

7. **Only use secure cryptographic**

   - Moxa Industrial Linux on Moxa Arm-based computer only uses secure cryptographic that are commonly accepted industry best practices and recommendations as defined in NIST SP 800-57.
   - Moxa Industrial Linux installed OpenSSL by default but doesn't disable weak algorithms such as TLS 1.0/1.1 and SSLv3. It is recommended that your application deployed on Moxa Industrial Linux only uses secure algorithms defined in NIST SP 800-57. You can disable the weaker cryptographic algorithm in OpenSSL by setting CipherString = DEFAULT@SECLEVEL=[desired level] in /etc/ssl/openssl.cnf to a higher level. For details, refers to: https://www.openssl.org/docs/man1.1.1/man3/SSL_CTX_set_security_level.html

8. **Malicious code protection**

   - Downloading file from untrusted sources is not recommended. If you still want to do it, make sure to verify the file using following recommendation:
     - Use sha256 or stronger algorithms checksums tools to check the integrity of the file before installing or transferring to device.
     - If the file is Debian package (.deb), follow "How to manually check for package's integrity" for the instruction.
     - Use OpenSSL to verify the signature of the file before installing or transferring to the device.

Administrator and User

1. **Periodically refresh password**

   Update user passwords on a timely manner. For administrator, we recommend refreshing password at least every 3 months.

2. **Encrypt confidential file**

   Use GPG or openSSL to encrypt confidential file or directory with a password in Linux. You can reference How To Encrypt And Decrypt Files With A Password for quick instructions.

Maintenance

1. **Perform Update Frequently**
   - Perform software upgrades frequently to enhance features, deploy security patches, or fix bugs.
   - We recommend you enable System Failback Recovery before performing critical update.
2. **Perform Backup Frequently**

   Frequently backup of system on timely manner
3. **Examine Audit Logs Frequently**

   Examine audit logs frequently to detect any anomalies.
4. **Report Vulnerability to Moxa**

   To report vulnerabilities of Moxa products, please submit your findings on the following web page: https://www.moxa.com/en/support/product-support/security-advisory/report-a-vulnerability.

Decommissioning

1. To avoid any sensitive information such as your account password or certificate from being disclosed, always use the **`mx-system-mgmt default decommission`** command to reset the computer to factory default and further wipe out all user data, including logs, in an unrecoverable manner before removing the Moxa Arm-based computer from.

   You must use sudo or run the command with the root permission.

   ```
   moxa@moxa-tbzkb1090923:/# sudo mx-system-mgmt default decommission
   ```

   The decommissioning process will do the following actions:

   a. Overwrite the system partition 4 times with shred so that all user files will be deleted and cannot be recovered.

   b. Overwrite the log partition 4 times with shred so that all log files will be deleted and cannot be recovered.

   c. Trigger the bootloader decommissioning function, so all configurations and log messages in the bootloader are also deleted and cannot be recovered.
2. If asset owner key or sensitive data is stored in the TPM, switch to bootloader Developer Mode and then perform Clear TPM action will clear all data stored in TPM.

10. Customization and Programming

MIL1 (Debian 9) to MIL3 (Debian 11) Migration

Moxa Arm-based computers with MIL1 (Debian 9) does not support direct upgrade to MIL3 (Debian 11). If you have such request, contact your regional sales representative.

If you are migrating an application previously developed on MIL1 to MIL3 please reference the below table for the major changes.

Category	Description	MIL1 (Debian 9)	MIL3 (Debian 11)
Password rule	Password change enforced upon first log-in	n/a	✓
	Password complexity enforcement	n/a	At least 8 characters in length Password dictionary check
Backup & Restore utilities	Reinstall a system image	Via bootloader menu	Via bootloader menu
	Create a backup & restore	n/a	Moxa System Manager (MSM) Use **mx-system-mgmt**
	Create a snapshot & restore	n/a	
	Automatic system failback recovery	n/a	
	Reset to factory default	Use **mx-set-def**	
Network connection utilities	Default LAN (ethernet) port configuration	LAN1(static IP):192.168.3.127 LAN2(static IP):192.168.4.127	• LAN1: Assigned by DHCP server. Link-local IP addresses will be assigned when DHCP server is not available • LAN2(static IP):192.168.4.127
	Cellular connection utility	Use **cell_mgmt**	Use **mx-connection-mgmt** Refers to Moxa Connection Manager (MCM) with **additional** features added below: • GUI to configure and manage network • Connection keep-alive • Connection failover/failback • Cellular, Wi-Fi and ethernet management • DHCP server • Data usage monitoring • IPv6 support • Cellular connection diagnosis • Cellular modem firmware upgrade • C API for network and connection status inquiry
	Wi-Fi connection utility	Use **wifi_mgmt**	
I/O and Interface Management utilities	Serial port mode change (RS-232, RS-422, RS-485 2-wire, and RS-485 4-wire)	Use **mx-uart-ctl**	Use **mx-interface-mgmt** Refers to serial port in Moxa Computer Interface Manager (MCIM) section

Category	Description	MIL1 (Debian 9)	MIL3 (Debian 11)
	Disabled unused port • Serial port • Serial console • CAN port • Ethernet port • External storage (e.g., USB, SD) Create LUKS encrypted storage (e.g., USB, SD)	n/a	
	Module control including power control, module detection, initialize setting, and SIM slot switching	Use **mx-module-ctl** or **cell_mgmt** for cellular module control	
	Buzzer control	n/a	
	LED control	Use **mx-led-ctl**	
	Digital I/O control	Use **moxa-dio-control**	
	Mount a SD/USB storage device	Use **moxa-auto-mountd.service**	
	Push button control	n/a	
Other Configuration	Check product serial number	Use **fw_printenv serialnumber**	Use **mx-interface-mgmt deviceinfo**
	Check system image version	Use **kversion or mx-ver**	Use **mx-ver**
	APT repository source list	All repository in **/etc/apt/sources.list**	3rd party repository in **/etc/apt/sources.list** Moxa repository in **/etc/apt/sources.list.d/moxa.list**
API and libraries	Moxa Platform Libraries	✓	API and libraries not available. Use **mx-interface-mgmt** Refers to Moxa Computer Interface Manager (MCIM)

Building an Application

Introduction

Moxa's Arm-based computers support both native and cross-compiling of code. Native compiling is more straightforward since all the coding and compiling can be done directly on the device. However, Arm architecture is less powerful and hence the compiling speed is slower. To overcome this, you can cross compile your code on a Linux machine using a toolchain; the compiling speed is much faster.

Native Compilation

Follow these steps to update the package menu:

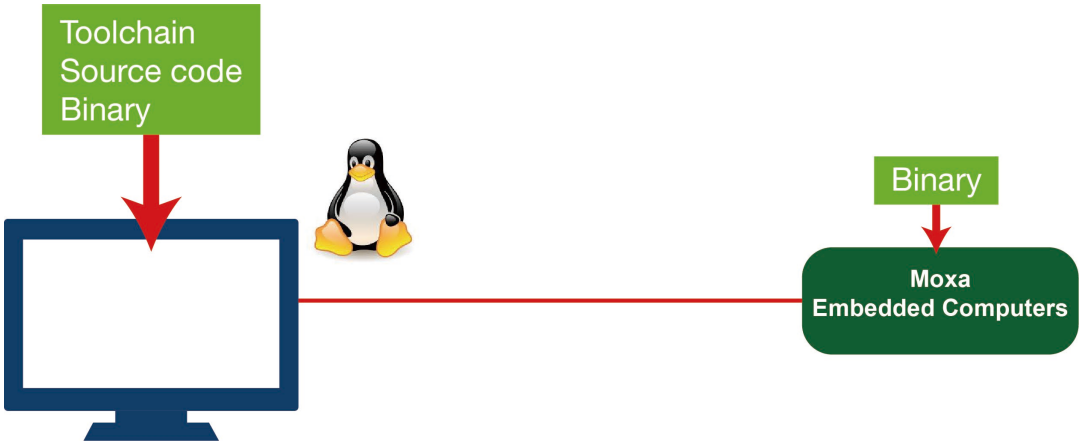
1. Make sure a network connection is available.
2. Use **aptupdate** to update the Debian package list.

```
moxa@Moxa-tbzkb1090923:~$ sudo apt update
```

3. Install the native compiler and necessary packages.

```
moxa@Moxa-tbzkb1090923:~$ sudo apt install gcc build-essential flex bison
automake
```

Cross Compilation



Moxa Industrial Linux (MIL) in Moxa's Arm-based computers is based on Debian. So, we recommend setting up a Debian environment on the host device to ensure best compatibility during cross compilation.

The toolchain will need about 300 MB of hard disk space on your PC.

To cross compile your code, do the following:

1. Set up a Debian 11 environment using a VM or Docker.
2. Open moxa.source.list in the vi editor.

   ```
   user@Linux:~$ sudo vi /etc/apt/sources.list.d/moxa.sources.list
   ```

   Add the following line to **moxa.source.list**:

   deb https://debian.moxa.com/mil3 bullseye main
3. Update the apt information.

   ```
   user@Linux:~$ apt update
   ```
4. (Optional) During the update process, if you don't want to see messages related to "server certificate verification failed", you can install Moxa apt **keyring**. These messages, however, will not affect the operation.

   ```
   user@Linux:~$ apt install moxa-archive-keyring
   ```
5. In order to install non-amd64 packages, such as armhf and u386, add the external architecture.

   In the example, we are adding the armhf architecture.

   ```
   user@Linux:~$ dpkg --add-architecture armhf
   ```
6. Update the apt information again.

   ```
   user@Linux:~$ apt update
   ```
7. Download the toolchain file from apt server (all Moxa UC series computers use the official Debian toolchain).

   For UC computer with **armhf** architecture

   ```
   user@Linux:~$ apt install crossbuild-essential-armhf
   ```

   For UC computer with **arm64** architecture

   ```
   user@Linux:~$ apt install crossbuild-essential-arm64
   ```
8. Install **dev** or **lib** packages depending on whether Debian or Moxa packages are applicable for the procedure.

   Example for installing an armhf Debian official package:

   ```
   user@Linux:~$ apt install libssl-dev:armhf
   ```

You can now start compiling programs using the toolchain.

NOTE

For all available libraries and headers offered by Debian, visit: https://packages.debian.org/index.

Example Program—hello

In this section, we use the standard "hello" example program to illustrate how to develop a program for Moxa computers. All example codes can be downloaded from Moxa's website. The "hello" example code is available in the **hello** folder; hello/hello.c:

```
#include <stdio.h>

int main(int argc, char *argv[])
{
    printf("Hello World\n");
    return 0;
}
```

Native Compilation

1. Compile the hello.c code.

```
moxa@Moxa-tbzkb1090923:~$ gcc -o hello hello.c
moxa@Moxa-tbzkb1090923:~$ strip -s hello
```

or

use the Makefile as follows:

```
moxa@Moxa-tbzkb1090923:~$ make
```

2. Run the program.

```
moxa@Moxa-tbzkb1090923:~$ ./hello
Hello World
```

Cross Compiling

1. Compile the hello.c code.

```
user@Linux:~$ arm-linux-gnueabihf-gcc -o hello \
    hello.c
user@Linux:~$ arm-linux-gnueabihf-strip -s hello
```

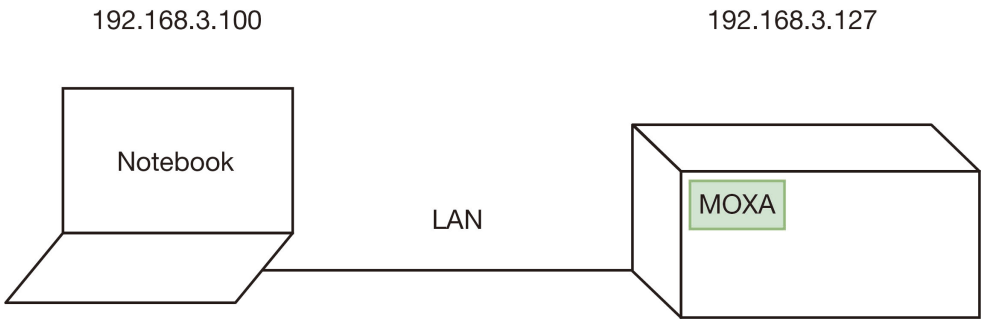
or

use the Makefile as follows:

```
user@Linux:~$ make CC=arm-linux-gnueabihf-gcc \
    STRIP=arm-linux-gnueabihf-strip
```

2. Copy the program to a Moxa computer:

For example, if the IP address of your device used for cross compiling the code is "192.168.3.100" and the IP address of the Moxa computer is "192.168.3.127", use the following command:




```
user@Linux:~$ scp hello moxa@192.168.3.127:~
```

3. Run the hello.c program on the Moxa computer.

```
moxa@Moxa-tbzkb1090923:~$ ./hello
Hello World
```

Example Makefile

You can create a Makefile for the "hello" example program using the following code. By default, the Makefile is set for native compiling.

"hello/Makefile":

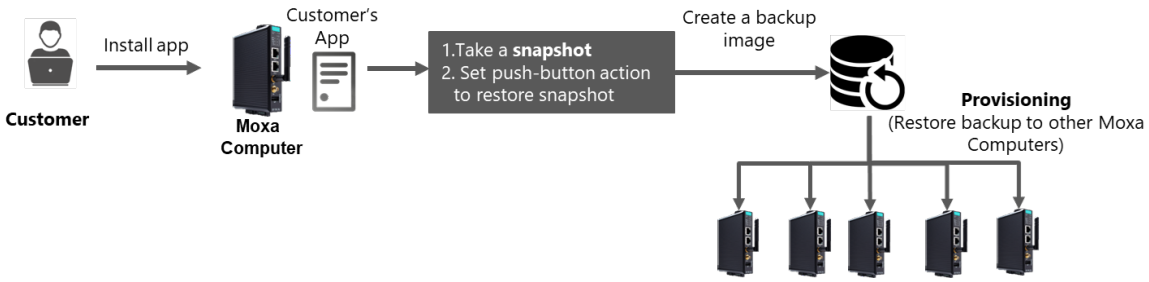
```
CC:=gcc
STRIP:=strip

all:
    $(CC) -o hello hello.c
    $(STRIP) -s hello

.PHONY: clean
clean:
    rm -f hello
```

To set the hello.c program for cross compilation, modify the toolchain settings as follows:

```
CC:=arm-linux-gnueabihf-gcc
STRIP:=arm-linux-gnueabihf-strip
```

Creating a Customized Image for Batch Provisioning

Introduction



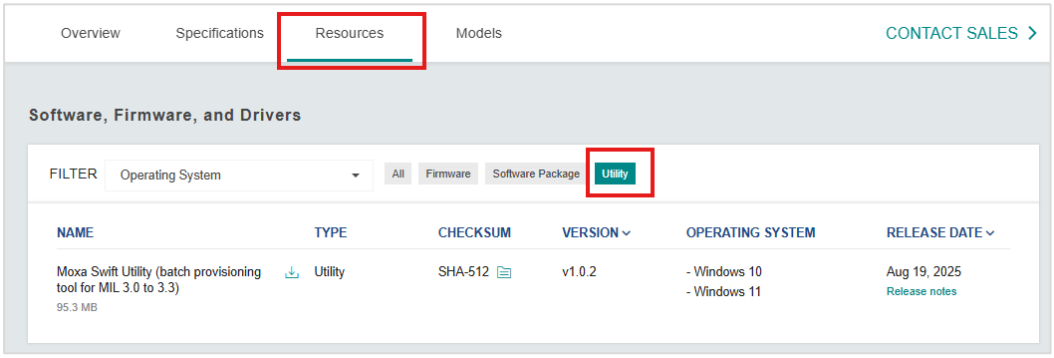
Creating and Using System Snapshots and Backups

1. Configure your Moxa Arm-based computer and install applications.
2. Create a Snapshot.
3. Configure a push-button on the computer to restore a snapshot.
   See Configure the Button Action.
4. Create a Backup Image.
   The backup will also include the snapshot taken earlier.

   The backup image can be used via the backup restore command to provisioning Moxa computers whose model name is the same as the computer used to create the backup image.

Alternatively, you can use the Moxa Swift utility to mass-provision multiple Moxa computers with the same model name using the backup image. The utility can be downloaded from the Resources tab of product page of any UC Series product that supports Moxa Industrial Linux 3.

For example, UC-3400A's product page



Connecting to Bluetooth

To establish a Bluetooth connection use the **bluetoothctl** and **hcitool** commands.

> **NOTE**
>
> This feature is only available for UC-3400A and UC-8200 Series.

Configuring Bluetooth HCI UART Transport

Before establishing a Bluetooth connection, you must configure Bluetooth HCI UART Transport with hardware flow control and set the baudrate at 115200.

To configure Bluetooth HCI UART Transport, do the following:

1. **Attach the Bluetooth interface.**

   Run the following command to attach the Bluetooth interface **`/dev/ttyS3`** to **`hci0`** with a baud rate of **`115200`** and enable hardware flow control.

   The **`hci0`** interface represents the first Host Controller Interface (HCI) device managed by the Linux Bluetooth stack.

   If multiple Bluetooth adapters are connected, they appear as hci1, hci2, and so on.

```
root@moxa-imoxa1234567:/home/moxa# hciattach /dev/ttyS3 any 115200 flow
[    27.238266] Bluetooth: Core ver 2.22
[    27.238454] NET: Registered protocol family 31
[    27.238459] Bluetooth: HCI device and connection manager initialized
[    27.238484] Bluetooth: HCI socket layer initialized
[    27.238492] Bluetooth: L2CAP socket layer initialized
[    27.238508] Bluetooth: SCO socket layer initialized
[    27.269348] Bluetooth: HCI UART driver ver 2.3
Device setup complete
[    27.269372] Bluetooth: HCI UART protocol H4 registered
root@moxa-imoxa1234567:/home/moxa# [    27.269479] Bluetooth: HCI UART protocol LL registered
[    27.269782] Bluetooth: HCI UART protocol Broadcom registered
[    27.269815] Bluetooth: HCI UART protocol QCA registered
[    27.450601] Bluetooth: BNEP (Ethernet Emulation) ver 1.3
[    27.450625] Bluetooth: BNEP filters: protocol multicast
[    27.450649] Bluetooth: BNEP socket layer initialized
[    27.474781] NET: Registered protocol family 38
```

2. **Bring up the Bluetooth device.**

   Once the interface is attached, bring up the Bluetooth device using **`# hciconfig hci0 up`**.

```
root@moxa-imoxa1234567:/home/moxa# hciconfig hci0 up
root@moxa-imoxa1234567:/home/moxa#
```

Using bluetoothctl to manage Bluetooth interface

The **bluetoothctl** utility provides an interactive command-line interface for managing Bluetooth devices. Use it to configure, control, and test Bluetooth connections directly from the terminal.

1. Open a terminal and run the **`# bluetoothctl`** to start the Bluetooth control utility.
2. Display controller information by using **`# show`**. This command displays details about the Bluetooth adapter, such as its name, MAC address, supported profiles, and advertising capabilities.

```
[DeviceB]# show
Controller 6C:1D:EB:98:72:08 (public)
        Name: moxa-imoxa34000a1
        Alias: moxa-imoxa34000a1
        Class: 0x002c0000
        Powered: yes
        Discoverable: no
        DiscoverableTimeout: 0x000000b4
        Pairable: yes
        UUID: A/V Remote Control        (0000110e-0000-1000-8000-00805f9b34fb)
        UUID: Audio Source              (0000110a-0000-1000-8000-00805f9b34fb)
        UUID: PnP Information           (00001200-0000-1000-8000-00805f9b34fb)
        UUID: Audio Sink                (0000110b-0000-1000-8000-00805f9b34fb)
        UUID: Headset                   (00001108-0000-1000-8000-00805f9b34fb)
        UUID: A/V Remote Control Target (0000110c-0000-1000-8000-00805f9b34fb)
        UUID: Generic Access Profile    (00001800-0000-1000-8000-00805f9b34fb)
        UUID: Generic Attribute Profile (00001801-0000-1000-8000-00805f9b34fb)
        UUID: Device Information        (0000180a-0000-1000-8000-00805f9b34fb)
        UUID: Headset AG                (00001112-0000-1000-8000-00805f9b34fb)
        Modalias: usb:v1D6Bp0246d0537
        Discovering: no
        Roles: central
        Roles: peripheral
Advertising Features:
        ActiveInstances: 0x00 (0)
        SupportedInstances: 0x06 (6)
        SupportedIncludes: tx-power
        SupportedIncludes: appearance
        SupportedIncludes: local-name
        SupportedSecondaryChannels: 1M
        SupportedSecondaryChannels: 2M
        SupportedSecondaryChannels: Coded
[DeviceB]#
```

3. Use the **`list`** command to see available Bluetooth Controllers.
4. If the bluetooth interface is not already on, use the **`power on`** command to enable it.
5. Make your device discoverable with the command **discoverable on**.
6. Begin scanning for nearby Bluetooth devices using the command **scan on**. You can also use **hcitool scan** for this step.

```
root@moxa-imoxa0000050:/home/moxa# bluetoothctl
Agent registered
[bluetooth]# list
Controller 18:62:E4:11:47:83 moxa-imoxa0000050 [default]
[bluetooth]# power on
Changing power on succeeded
[bluetooth]# scan on
Discovery started
[CHG] Controller 18:62:E4:11:47:83 Discovering: yes
[NEW] Device 58:91:60:11:4B:F3 58-91-60-11-4B-F3
[NEW] Device 5C:9C:BA:B0:1B:BE 5C-9C-BA-B0-1B-BE
[NEW] Device 3C:61:23:ED:74:EB 3C-61-23-ED-74-EB
[NEW] Device CE:2B:93:CB:F5:2F CE-2B-93-CB-F5-2F
[NEW] Device 48:13:38:6B:6F:4B 48-13-38-6B-6F-4B
[NEW] Device D0:D2:B0:8C:4A:F5 D0-D2-B0-8C-4A-F5
[NEW] Device 43:02:1D:00:23:65 43-02-1D-00-23-65
[NEW] Device 67:D5:52:FF:24:0D 67-D5-52-FF-24-0D
[CHG] Device D0:D2:B0:8C:4A:F5 RSSI: -73
[CHG] Device 5C:9C:BA:B0:1B:BE RSSI: -89
[NEW] Device C8:6B:BA:55:AA:FF C8-6B-BA-55-AA-FF
```

```
root@moxa-imoxa0000050:/home/moxa# hcitool scan
Scanning ...
        D4:C8:B0:4A:06:E8       n/a
        40:9C:28:E5:05:B5       DSAP
```

7. Identify the MAC address of the Bluetooth device you want to connect to from the scan results.

8. Stop scanning with the command **scan off**.

```
[bluetooth]# scan off
Discovery stopped
[CHG] Controller 18:62:E4:11:47:83 Discovering: no
[CHG] Device C8:6B:BA:55:AA:FF RSSI is nil
[CHG] Device 67:D5:52:FF:24:0D TxPower is nil
[CHG] Device 67:D5:52:FF:24:0D RSSI is nil
[CHG] Device 43:02:1D:00:23:65 TxPower is nil
[CHG] Device 43:02:1D:00:23:65 RSSI is nil
[CHG] Device D0:D2:B0:8C:4A:F5 TxPower is nil
[CHG] Device D0:D2:B0:8C:4A:F5 RSSI is nil
[CHG] Device 48:13:38:6B:6F:4B TxPower is nil
[CHG] Device 48:13:38:6B:6F:4B RSSI is nil
[CHG] Device CE:2B:93:CB:F5:2F RSSI is nil
[CHG] Device 3C:61:23:ED:74:EB RSSI is nil
[CHG] Device 5C:9C:BA:B0:1B:BE TxPower is nil
[CHG] Device 5C:9C:BA:B0:1B:BE RSSI is nil
[CHG] Device 58:91:60:11:4B:F3 TxPower is nil
[CHG] Device 58:91:60:11:4B:F3 RSSI is nil
```

9. Start an agent using the command **agent on**.
10. Trust, Pair, and Connect:
    a. Trust the device using the command **trust MACADDRESS**.
    b. Pair with the device using the command **pair MACADDRESS**.

```
[bluetooth]# trust 40:9C:28:E5:05:B5
[CHG] Device 40:9C:28:E5:05:B5 Trusted: yes
Changing 40:9C:28:E5:05:B5 trust succeeded
[bluetooth]# pair 40:9C:28:E5:05:B5
Attempting to pair with 40:9C:28:E5:05:B5
[CHG] Device 40:9C:28:E5:05:B5 Connected: yes
Request confirmation
[agent] Confirm passkey 710172 (yes/no): yes
[CHG] Device 40:9C:28:E5:05:B5 Modalias: bluetooth:v004Cp710Ed0F50
[CHG] Device 40:9C:28:E5:05:B5 UUIDs: 00000000-deca-fade-deca-deafdecacafe
[CHG] Device 40:9C:28:E5:05:B5 UUIDs: 00001000-0000-1000-8000-00805f9b34fb
[CHG] Device 40:9C:28:E5:05:B5 UUIDs: 0000110a-0000-1000-8000-00805f9b34fb
[CHG] Device 40:9C:28:E5:05:B5 UUIDs: 0000110c-0000-1000-8000-00805f9b34fb
[CHG] Device 40:9C:28:E5:05:B5 UUIDs: 0000110e-0000-1000-8000-00805f9b34fb
[CHG] Device 40:9C:28:E5:05:B5 UUIDs: 00001116-0000-1000-8000-00805f9b34fb
[CHG] Device 40:9C:28:E5:05:B5 UUIDs: 0000111f-0000-1000-8000-00805f9b34fb
[CHG] Device 40:9C:28:E5:05:B5 UUIDs: 0000112f-0000-1000-8000-00805f9b34fb
[CHG] Device 40:9C:28:E5:05:B5 UUIDs: 00001132-0000-1000-8000-00805f9b34fb
[CHG] Device 40:9C:28:E5:05:B5 UUIDs: 00001200-0000-1000-8000-00805f9b34fb
[CHG] Device 40:9C:28:E5:05:B5 UUIDs: 00001801-0000-1000-8000-00805f9b34fb
[CHG] Device 40:9C:28:E5:05:B5 UUIDs: 02030302-1d19-415f-86f2-22a2106a0a77
[CHG] Device 40:9C:28:E5:05:B5 ServicesResolved: yes
[CHG] Device 40:9C:28:E5:05:B5 Paired: yes
Pairing successful
[CHG] Device 40:9C:28:E5:05:B5 ServicesResolved: no
[CHG] Device 40:9C:28:E5:05:B5 Connected: no
```

    c. Connect to the device using the command **connect MACADDRESS** or initiate the connection from your phone.

```
[CHG] Device 40:9C:28:E5:05:B5 Connected: yes
[DSAP]#
```

11. Use **info MACADDRESS** to display information about the connected device. It will show paired device profiles such as Audio Source Profile.

```
root@moxa-imoxa0000050:/home/moxa# bluetoothctl info 40:9C:28:E5:05:B5
Device 40:9C:28:E5:05:B5 (public)
        Name: DSAP
        Alias: DSAP
        Class: 0x007a020c
        Icon: phone
        Paired: yes
        Trusted: yes
        Blocked: no
        Connected: yes
        LegacyPairing: no
        UUID: Vendor specific           (00000000-deca-fade-deca-deafdecacafe)
        UUID: Service Discovery Serve.. (00001000-0000-1000-8000-00805f9b34fb)
        UUID: Audio Source              (0000110a-0000-1000-8000-00805f9b34fb)
        UUID: A/V Remote Control Target (0000110c-0000-1000-8000-00805f9b34fb)
        UUID: A/V Remote Control        (0000110e-0000-1000-8000-00805f9b34fb)
        UUID: NAP                       (00001116-0000-1000-8000-00805f9b34fb)
        UUID: Handsfree Audio Gateway   (0000111f-0000-1000-8000-00805f9b34fb)
        UUID: Phonebook Access Server   (0000112f-0000-1000-8000-00805f9b34fb)
        UUID: Message Access Server     (00001132-0000-1000-8000-00805f9b34fb)
        UUID: PnP Information           (00001200-0000-1000-8000-00805f9b34fb)
        UUID: Generic Attribute Profile (00001801-0000-1000-8000-00805f9b34fb)
        UUID: Vendor specific           (02030302-1d19-415f-86f2-22a2106a0a77)
        Modalias: bluetooth:v004Cp710Ed0F50
```

Using hcitool for sending HCI commands

1. You can use **hcitool cmd** to send HCI commands.
2. The command format and example can be found in the provided link: https://software-dl.ti.com/simplelink/esd/simplelink_cc13x2_sdk/1.60.00.29_new/exports/docs/ble5stack/vendor_specific_guide/BLE_Vendor_Specific_HCI_Guide/hci_interface.html
3. For example, to reset the module, use the command **0x3 0x3**.

```
root@moxa-imoxa0000050:/home/moxa# hcitool cmd 0x03 0x03
< HCI Command: ogf 0x03, ocf 0x0003, plen 0
> HCI Event: 0x05 plen 4
  00 01 00 16
```

Troubleshooting

- If the **bluetoothctl connect** command fails, ensure that both devices have a compatible profile.
- Error messages such as **connect failed for MACADDRESS: Protocol not available2dp-sink profile** or **connect failed for MACADDRESS: Protocol not available2dp-source profile** indicate a missing profile.

```
root@moxa-imoxa34000a1:/home/moxa# systemctl status bluetooth
● bluetooth.service - Bluetooth service
     Loaded: loaded (/lib/systemd/system/bluetooth.service; enabled; vendor preset: enabled)
     Active: active (running) since Wed 2024-10-16 23:12:40 GMT; 37s ago
       Docs: man:bluetoothd(8)
   Main PID: 3420 (bluetoothd)
     Status: "Running"
      Tasks: 1 (limit: 4011)
     Memory: 752.0K
        CPU: 83ms
     CGroup: /system.slice/bluetooth.service
             └─3420 /usr/libexec/bluetooth/bluetoothd

Oct 16 23:12:40 moxa-imoxa34000a1 systemd[1]: Starting Bluetooth service...
Oct 16 23:12:40 moxa-imoxa34000a1 bluetoothd[3420]: Bluetooth daemon 5.55
Oct 16 23:12:40 moxa-imoxa34000a1 systemd[1]: Started Bluetooth service.
Oct 16 23:12:40 moxa-imoxa34000a1 bluetoothd[3420]: Starting SDP server
Oct 16 23:12:40 moxa-imoxa34000a1 bluetoothd[3420]: Bluetooth management interface 1.18 initialized
Oct 16 23:12:40 moxa-imoxa34000a1 bluetoothd[3420]: profiles/sap/server.c:sap_server_register() Sap driver initialization faile
d.
Oct 16 23:12:40 moxa-imoxa34000a1 bluetoothd[3420]: sap-server: Operation not permitted (1)
Oct 16 23:13:12 moxa-imoxa34000a1 bluetoothd[3420]: src/service.c:btd_service_connect() a2dp-sink profile connect failed for 20
:BA:36:53:7E:AC: Protocol not available
Oct 16 23:13:12 moxa-imoxa34000a1 bluetoothd[3420]: src/service.c:btd_service_connect() a2dp-source profile connect failed for
20:BA:36:53:7E:AC: Protocol not available
```

- Install **pulseaudio-utils** and **pulseaudio-module-bluetooth** to resolve the issue. Use the command **apt install pulseaudio-utils pulseaudio-module-bluetooth**.

A. Software Process List

Below is a list of software processes of Moxa Industrial Linux in Moxa Arm-based Computer

Software Process for Adminstrator	UID	GID
accessdb	root	root
addgnupghome	root	root
addgroup	root	root
add-shell	root	root
adduser	root	root
agetty	root	root
applygnupgdefaults	root	root
arp	root	root
arpd	root	root
audisp-syslog	root	root
auditctl	root	root
auditd	root	root
augenrules	root	root
aureport	root	root
ausearch	root	root
autrace	root	root
avahi-daemon	root	root
badblocks	root	root
blkdeactivate	root	root
blkdiscard	root	root
blkid	root	root
blkzone	root	root
blockdev	root	root
bluetoothd	root	root
bridge	root	root
capsh	root	root
cellular_module_control.sh	root	root
cfdisk	root	root
chcpu	root	root
chgpasswd	root	root
chmem	root	root
chpasswd	root	root
chronyd	root	root
chroot	root	root
cpgr	root	root
cppw	root	root
cracklib-check	root	root
cracklib-format	root	root
cracklib-packer	root	root
cracklib-unpacker	root	root
create-cracklib-dict	root	root
ctrlaltdel	root	root
debugfs	root	root
delgroup	root	root
deluser	root	root
depmod	root	root
devlink	root	root
dhclient	root	root
dhclient-script	root	root
dmsetup	root	root

Software Process for Adminstrator	UID	GID
dmstats	root	root
dnsmasq	root	root
docfdisk	root	root
doc_loadbios	root	root
dpkg-fsys-usrunmess	root	root
dpkg-preconfigure	root	root
dpkg-reconfigure	root	root
dumpe2fs	root	root
e2freefrag	root	root
e2fsck	root	root
e2image	root	root
e2label	root	root
e2mmpstatus	root	root
e2scrub	root	root
e2scrub_all	root	root
e2undo	root	root
e4crypt	root	root
e4defrag	root	root
ethtool	root	root
faillock	root	root
fdformat	root	root
fdisk	root	root
filefrag	root	root
findfs	root	root
flashcp	root	root
flash_erase	root	root
flash_eraseall	root	root
flash_lock	root	root
flash_otp_dump	root	root
flash_otp_info	root	root
flash_otp_lock	root	root
flash_otp_write	root	root
flash_unlock	root	root
fsck	root	root
fsck.cramfs	root	root
fsck.ext2	root	root
fsck.ext3	root	root
fsck.ext4	root	root
fsck.minix	root	root
fsfreeze	root	root
fstab-decode	root	root
fstrim	root	root
ftl_check	root	root
ftl_format	root	root
genl	root	root
getcap	root	root
getpcaps	root	root
getty	root	root
gpsd	root	root
gpsdctl	root	root
groupadd	root	root
groupdel	root	root
groupmems	root	root
groupmod	root	root
grpck	root	root
grpconv	root	root
grpunconv	root	root
halt	root	root

Software Process for Adminstrator	UID	GID
hwclock	root	root
i2cdetect	root	root
i2cdump	root	root
i2cget	root	root
i2cset	root	root
i2c-stub-from-dump	root	root
i2ctransfer	root	root
iconvconfig	root	root
ifconfig	root	root
ifdown	root	root
ifquery	root	root
ifup	root	root
init	root	root
insmod	root	root
installkernel	root	root
invoke-rc.d	root	root
ip	root	root
ipmaddr	root	root
iptunnel	root	root
isosize	root	root
iw	root	root
jffs2dump	root	root
jffs2reader	root	root
killall5	root	root
ldattach	root	root
ldconfig	root	root
lldpcli	_lldpd	adm
lldpctl	root	root
lldpd	root	root
locale-gen	root	root
logrotate	root	root
logsave	root	root
losetup	root	root
lsmod	root	root
lsmtd	root	root
lxfp	root	root
mcmd	root	root
mg	root	root
mii-tool	root	root
mke2fs	root	root
mkfs	root	root
mkfs.bfs	root	root
mkfs.cramfs	root	root
mkfs.ext2	root	root
mkfs.ext3	root	root
mkfs.ext4	root	root
mkfs.jffs2	root	root
mkfs.minix	root	root
mkfs.ubifs	root	root
mkhomedir_helper	root	root
mklost+found	root	root
mkswap	root	root
ModemManager	root	root
modinfo	root	root
modprobe	root	root
moxa-cellular-firmware-upgrade-tool	root	root
MoxaComputerInterfaceManager	root	root
moxa-edge	root	root

Software Process for Adminstrator	UID	GID
moxa-mdns	root	root
moxa-mdns-set-hostname.sh	root	root
mtd_debug	root	root
mtdinfo	root	root
mtdpart	root	root
mx-connect-mgmt	root	root
mx-guardian	root	root
mx-guardian-init	root	root
mx-pwr-mgmt	root	root
mx-system-mgmt	root	root
nameif	root	root
nanddump	root	root
nandtest	root	root
nandwrite	root	root
NetworkManager	root	root
newusers	root	root
nft	root	root
nftldump	root	root
nftl_format	root	root
nginx	root	root
nologin	root	root
pam-auth-update	root	root
pam_getenv	root	root
pam_timestamp_check	root	root
parted	root	root
partprobe	root	root
pivot_root	root	root
plipconfig	root	root
poweroff	root	root
pwck	root	root
pwconv	root	root
pwunconv	root	root
QFirehose	root	root
rarp	root	root
raw	root	root
readprofile	root	root
reboot	root	root
recv_image	root	root
regdbdump	root	root
remove-shell	root	root
resize2fs	root	root
rfddump	root	root
rfdformat	root	root
rmmod	root	root
rmt	root	root
rmt-tar	root	root
route	root	root
rsyslogd	root	root
rtacct	root	root
rtcwake	root	root
rtmon	root	root
runlevel	root	root
runuser	root	root
sensors-detect	root	root
serve_image	root	root
service	root	root
setcap	root	root
sfdisk	root	root

Software Process for Adminstrator	UID	GID
shadowconfig	root	root
shutdown	root	root
slattach	root	root
snmpd	root	root
sshd	root	root
start-stop-daemon	root	root
sudo_logsrvd	root	root
sudo_sendlog	root	root
sulogin	root	root
sumtool	root	root
swaplabel	root	root
swapoff	root	root
swapon	root	root
switch_root	root	root
sysctl	root	root
tarcat	root	root
tc	root	root
telinit	root	root
tipc	root	root
tune2fs	root	root
tzconfig	root	root
ubiattach	root	root
ubiblock	root	root
ubicrc32	root	root
ubidetach	root	root
ubiformat	root	root
ubihealthd	root	root
ubimkvol	root	root
ubinfo	root	root
ubinize	root	root
ubirename	root	root
ubirmvol	root	root
ubirsvol	root	root
ubiupdatevol	root	root
unix_chkpwd	root	shadow
unix_update	root	root
update-ca-certificates	root	root
update-cracklib	root	root
update-locale	root	root
update-passwd	root	root
update-rc.d	root	root
useradd	root	root
userdel	root	root
usermod	root	root
uxfp	root	root
validlocale	root	root
vigr	root	root
vipw	root	root
visudo	root	root
vnstatd	root	root
watchdog	root	root
wd_identify	root	root
wd_keepalive	root	root
wipefs	root	root
wpa_action	root	root
wpa_cli	root	root
wpa_supplicant	root	root
zic	root	root

Software Process for Adminstrator	UID	GID
zramctl	root	root

Software Process for Non-Adminstrator	UID	GID
addpart	root	root
addr2line	root	root
aide	root	root
apt	root	root
apt-cache	root	root
apt-cdrom	root	root
apt-config	root	root
apt-extracttemplates	root	root
apt-ftparchive	root	root
apt-get	root	root
apt-key	root	root
apt-mark	root	root
apt-sortpkgs	root	root
ar	root	root
arch	root	root
arm-linux-gnueabihf-addr2line	root	root
arm-linux-gnueabihf-ar	root	root
arm-linux-gnueabihf-as	root	root
arm-linux-gnueabihf-c++filt	root	root
arm-linux-gnueabihf-dwp	root	root
arm-linux-gnueabihf-elfedit	root	root
arm-linux-gnueabihf-gold	root	root
arm-linux-gnueabihf-gprof	root	root
arm-linux-gnueabihf-ld	root	root
arm-linux-gnueabihf-ld.bfd	root	root
arm-linux-gnueabihf-ld.gold	root	root
arm-linux-gnueabihf-nm	root	root
arm-linux-gnueabihf-objcopy	root	root
arm-linux-gnueabihf-objdump	root	root
arm-linux-gnueabihf-ranlib	root	root
arm-linux-gnueabihf-readelf	root	root
arm-linux-gnueabihf-size	root	root
arm-linux-gnueabihf-strings	root	root
arm-linux-gnueabihf-strip	root	root
as	root	root
asc2log	root	root
aulast	root	root
aulastlog	root	root
ausyscall	root	root
auvirt	root	root
awk	root	root
b2sum	root	root
base32	root	root
base64	root	root
basename	root	root
basenc	root	root
bash	root	root
bashbug	root	root
bcmserver	root	root
bootctl	root	root
busctl	root	root
cal	root	root
canbusload	root	root
can-calc-bit-timing	root	root

Software Process for Non-Adminstrator	UID	GID
candump	root	root
canfdtest	root	root
cangen	root	root
cangw	root	root
canlogserver	root	root
canplayer	root	root
cansend	root	root
cansequence	root	root
cansniffer	root	root
captoinfo	root	root
cat	root	root
catchsegv	root	root
c++filt	root	root
chacl	root	root
chage	root	shadow
chattr	root	root
chcon	root	root
chfn	root	root
chgrp	root	root
chmod	root	root
choom	root	root
chown	root	root
chronyc	root	root
chrt	root	root
chsh	root	root
cksum	root	root
clear	root	root
clear_console	root	root
cmp	root	root
col	root	root
colcrt	root	root
colrm	root	root
column	root	root
comm	root	root
corelist	root	root
cp	root	root
cpan	root	root
cpan5.32-arm-linux-gnueabihf	root	root
cpulimit	root	root
c_rehash	root	root
csplit	root	root
ctstat	root	root
curl	root	root
cut	root	root
cvtsudoers	root	root
dash	root	root
date	root	root
dbus-cleanup-sockets	root	root
dbus-daemon	root	root
dbus-monitor	root	root
dbus-run-session	root	root
dbus-send	root	root
dbus-update-activation-environment	root	root
dbus-uuidgen	root	root
dd	root	root
debconf	root	root
debconf-apt-progress	root	root

Software Process for Non-Adminstrator	UID	GID
debconf-communicate	root	root
debconf-copydb	root	root
debconf-escape	root	root
debconf-set-selections	root	root
debconf-show	root	root
debsums	root	root
deb-systemd-helper	root	root
deb-systemd-invoke	root	root
delpart	root	root
df	root	root
dh_bash-completion	root	root
dialog	root	root
diff	root	root
diff3	root	root
dir	root	root
dircolors	root	root
dirmngr	root	root
dirmngr-client	root	root
dirname	root	root
dmesg	root	root
dnsdomainname	root	root
domainname	root	root
dpkg	root	root
dpkg-deb	root	root
dpkg-divert	root	root
dpkg-maintscript-helper	root	root
dpkg-query	root	root
dpkg-realpath	root	root
dpkg-split	root	root
dpkg-statoverride	root	root
dpkg-trigger	root	root
du	root	root
dumpimage	root	root
dwp	root	root
echo	root	root
editor	root	root
egrep	root	root
elfedit	root	root
enc2xs	root	root
encguess	root	root
env	root	root
ex	root	root
expand	root	root
expiry	root	shadow
expr	root	root
factor	root	root
fail2ban-client	root	root
fail2ban-python	root	root
fail2ban-regex	root	root
fail2ban-server	root	root
fail2ban-testcases	root	root
faillog	root	root
fallocate	root	root
FALSE	root	root
fgrep	root	root
file	root	root
fincore	root	root

Software Process for Non-Adminstrator	UID	GID
find	root	root
findmnt	root	root
flock	root	root
fmt	root	root
fold	root	root
free	root	root
fw_printenv	root	root
fw_setenv	root	root
getconf	root	root
getent	root	root
getfacl	root	root
getopt	root	root
gold	root	root
gpasswd	root	root
gpg	root	root
gpg-agent	root	root
gpgcompose	root	root
gpgconf	root	root
gpg-connect-agent	root	root
gpgparsemail	root	root
gpgsm	root	root
gpgsplit	root	root
gpgtar	root	root
gpgv	root	root
gpg-wks-server	root	root
gpg-zip	root	root
gprof	root	root
grep	root	root
groups	root	root
gunzip	root	root
gzexe	root	root
gzip	root	root
h2ph	root	root
h2xs	root	root
hd	root	root
head	root	root
helpztags	root	root
hexdump	root	root
hostid	root	root
hostname	root	root
hostnamectl	root	root
iconv	root	root
id	root	root
infocmp	root	root
infotocap	root	root
install	root	root
instmodsh	root	root
ionice	root	root
ip	root	root
ipcmk	root	root
ipcrm	root	root
ipcs	root	root
ischroot	root	root
isotpdump	root	root
isotpperf	root	root
isotprecv	root	root
isotpsend	root	root

Software Process for Non-Adminstrator	UID	GID
isotpserver	root	root
isotpsniffer	root	root
isotptun	root	root
j1939acd	root	root
j1939cat	root	root
j1939spy	root	root
j1939sr	root	root
join	root	root
journalctl	root	root
jq	root	root
json_pp	root	root
kbxutil	root	root
kernel-install	root	root
kill	root	root
kmod	root	root
kwboot	root	root
last	root	root
lastb	root	root
lastlog	root	root
lcf	root	root
ld	root	root
ld.bfd	root	root
ldd	root	root
ld.gold	root	root
libnetcfg	root	root
link	root	root
linux32	root	root
linux64	root	root
ln	root	root
lnstat	root	root
locale	root	root
localectl	root	root
localedef	root	root
log2asc	root	root
log2long	root	root
logger	root	root
login	root	root
loginctl	root	root
logname	root	root
look	root	root
ls	root	root
lsattr	root	root
lsblk	root	root
lscpu	root	root
lsipc	root	root
lslocks	root	root
lslogins	root	root
lsmem	root	root
lsmod	root	root
lsns	root	root
lspgpot	root	root
mawk	root	root
mcookie	root	root
md5sum	root	root
md5sum.textutils	root	root
mesg	root	root
migrate-pubring-from-classic-gpg	root	root

Software Process for Non-Adminstrator	UID	GID
mkdir	root	root
mkenvimage	root	root
mkfifo	root	root
mkimage	root	root
mknod	root	root
mksunxiboot	root	root
mktemp	root	root
mmcli	root	root
more	root	root
mount	root	root
mountpoint	root	root
mv	root	root
mx-interface-mgmt	root	root
mx-ver	root	root
namei	root	root
nawk	root	root
ncal	root	root
netstat	root	root
networkctl	root	root
newgrp	root	root
nice	root	root
nisdomainname	root	root
nl	root	root
nm	root	root
nmcli	root	root
nm-online	root	root
nmtui	root	root
nmtui-connect	root	root
nmtui-edit	root	root
nmtui-hostname	root	root
nohup	root	root
nproc	root	root
nsenter	root	root
nstat	root	root
numfmt	root	root
objcopy	root	root
objdump	root	root
od	root	root
openssl	root	root
pager	root	root
partx	root	root
passwd	root	root
paste	root	root
pathchk	root	root
pdb3	root	root
pdb3.9	root	root
perl	root	root
perl5.32.1	root	root
perl5.32-arm-linux-gnueabihf	root	root
perlbug	root	root
perldoc	root	root
perlivp	root	root
perlthanks	root	root
pgrep	root	root
piconv	root	root
pidof	root	root
pidwait	root	root

Software Process for Non-Adminstrator	UID	GID
pinentry	root	root
pinentry-curses	root	root
ping	root	root
ping4	root	root
ping6	root	root
pinky	root	root
pkaction	root	root
pkcheck	root	root
pkexec	root	root
pkill	root	root
pkttyagent	root	root
pl2pm	root	root
pldd	root	root
pmap	root	root
pod2html	root	root
pod2man	root	root
pod2text	root	root
pod2usage	root	root
podchecker	root	root
pr	root	root
printenv	root	root
printf	root	root
prlimit	root	root
prove	root	root
ps	root	root
ptar	root	root
ptardiff	root	root
ptargrep	root	root
ptx	root	root
pv	root	root
pwd	root	root
pwdx	root	root
py3clean	root	root
py3compile	root	root
py3versions	root	root
pydoc3	root	root
pydoc3.9	root	root
pygettext3	root	root
pygettext3.9	root	root
python3	root	root
python3.9	root	root
ranlib	root	root
rbash	root	root
rcp	root	root
rdebsums	root	root
rdma	root	root
readelf	root	root
readlink	root	root
realpath	root	root
renice	root	root
reset	root	root
resizepart	root	root
resolvectl	root	root
rev	root	root
rgrep	root	root
rlogin	root	root
rm	root	root

Software Process for Non-Adminstrator	UID	GID
rmdir	root	root
routef	root	root
routel	root	root
rrsync	root	root
rsh	root	root
rsync	root	root
rsync-ssl	root	root
rtstat	root	root
runcon	root	root
run-parts	root	root
rview	root	root
rvim	root	root
savelog	root	root
scp	root	root
script	root	root
scriptlive	root	root
scriptreplay	root	root
sdiff	root	root
sed	root	root
select-editor	root	root
sensible-browser	root	root
sensible-editor	root	root
sensible-pager	root	root
seq	root	root
setarch	root	root
setfacl	root	root
setpriv	root	root
setsid	root	root
setterm	root	root
sftp	root	root
sg	root	root
sh	root	root
sha1sum	root	root
sha224sum	root	root
sha256sum	root	root
sha384sum	root	root
sha512sum	root	root
shasum	root	root
shred	root	root
shuf	root	root
size	root	root
skill	root	root
slabtop	root	root
slcan_attach	root	root
slcand	root	root
slcanpty	root	root
sleep	root	root
slogin	root	root
snice	root	root
sort	root	root
splain	root	root
split	root	root
ss	root	root
ssh	root	root
ssh-add	root	root
ssh-agent	root	ssh
ssh-argv0	root	root

Software Process for Non-Adminstrator	UID	GID
ssh-copy-id	root	root
ssh-keygen	root	root
ssh-keyscan	root	root
stat	root	root
stdbuf	root	root
streamzip	root	root
strings	root	root
strip	root	root
stty	root	root
su	root	root
sudo	root	root
sudoedit	root	root
sudoreplay	root	root
sum	root	root
sync	root	root
systemctl	root	root
systemd	root	root
systemd-analyze	root	root
systemd-ask-password	root	root
systemd-cat	root	root
systemd-cgls	root	root
systemd-cgtop	root	root
systemd-delta	root	root
systemd-detect-virt	root	root
systemd-escape	root	root
systemd-hwdb	root	root
systemd-id128	root	root
systemd-inhibit	root	root
systemd-machine-id-setup	root	root
systemd-mount	root	root
systemd-notify	root	root
systemd-path	root	root
systemd-resolve	root	root
systemd-run	root	root
systemd-socket-activate	root	root
systemd-stdio-bridge	root	root
systemd-sysusers	root	root
systemd-tmpfiles	root	root
systemd-tty-ask-password-agent	root	root
systemd-umount	root	root
tabs	root	root
tac	root	root
tail	root	root
tar	root	root
taskset	root	root
tee	root	root
tempfile	root	root
test	root	root
testj1939	root	root
tic	root	root
timedatectl	root	root
timeout	root	root
tload	root	root
toe	root	root
top	root	root
touch	root	root
tpm2	root	root

Software Process for Non-Adminstrator	UID	GID
tpm2_activatecredential	root	root
tpm2_certify	root	root
tpm2_certifycreation	root	root
tpm2_certifyX509certutil	root	root
tpm2_changeauth	root	root
tpm2_changeeps	root	root
tpm2_changepps	root	root
tpm2_checkquote	root	root
tpm2_clear	root	root
tpm2_clearcontrol	root	root
tpm2_clockrateadjust	root	root
tpm2_commit	root	root
tpm2_create	root	root
tpm2_createak	root	root
tpm2_createek	root	root
tpm2_createpolicy	root	root
tpm2_createprimary	root	root
tpm2_dictionarylockout	root	root
tpm2_duplicate	root	root
tpm2_ecdhkeygen	root	root
tpm2_ecdhzgen	root	root
tpm2_ecephemeral	root	root
tpm2_encryptdecrypt	root	root
tpm2_eventlog	root	root
tpm2_evictcontrol	root	root
tpm2_flushcontext	root	root
tpm2_getcap	root	root
tpm2_getcommandauditdigest	root	root
tpm2_geteccparameters	root	root
tpm2_getekcertificate	root	root
tpm2_getrandom	root	root
tpm2_getsessionauditdigest	root	root
tpm2_gettestresult	root	root
tpm2_gettime	root	root
tpm2_hash	root	root
tpm2_hierarchycontrol	root	root
tpm2_hmac	root	root
tpm2_import	root	root
tpm2_incrementalselftest	root	root
tpm2_load	root	root
tpm2_loadexternal	root	root
tpm2_makecredential	root	root
tpm2_nvcertify	root	root
tpm2_nvdefine	root	root
tpm2_nvextend	root	root
tpm2_nvincrement	root	root
tpm2_nvread	root	root
tpm2_nvreadlock	root	root
tpm2_nvreadpublic	root	root
tpm2_nvsetbits	root	root
tpm2_nvundefine	root	root
tpm2_nvwrite	root	root
tpm2_nvwritelock	root	root
tpm2_pcrallocate	root	root
tpm2_pcrevent	root	root
tpm2_pcrextend	root	root
tpm2_pcrread	root	root

Software Process for Non-Adminstrator	UID	GID
tpm2_pcrreset	root	root
tpm2_policyauthorize	root	root
tpm2_policyauthorizenv	root	root
tpm2_policyauthvalue	root	root
tpm2_policycommandcode	root	root
tpm2_policycountertimer	root	root
tpm2_policycphash	root	root
tpm2_policyduplicationselect	root	root
tpm2_policylocality	root	root
tpm2_policynamehash	root	root
tpm2_policynv	root	root
tpm2_policynvwritten	root	root
tpm2_policyor	root	root
tpm2_policypassword	root	root
tpm2_policypcr	root	root
tpm2_policyrestart	root	root
tpm2_policysecret	root	root
tpm2_policysigned	root	root
tpm2_policytemplate	root	root
tpm2_policyticket	root	root
tpm2_print	root	root
tpm2_quote	root	root
tpm2_rc_decode	root	root
tpm2_readclock	root	root
tpm2_readpublic	root	root
tpm2_rsadecrypt	root	root
tpm2_rsaencrypt	root	root
tpm2_selftest	root	root
tpm2_send	root	root
tpm2_setclock	root	root
tpm2_setcommandauditstatus	root	root
tpm2_setprimarypolicy	root	root
tpm2_shutdown	root	root
tpm2_sign	root	root
tpm2_startauthsession	root	root
tpm2_startup	root	root
tpm2_stirrandom	root	root
tpm2_testparms	root	root
tpm2_unseal	root	root
tpm2_verifysignature	root	root
tpm2_zgen2phase	root	root
tput	root	root
tr	root	root
TRUE	root	root
truncate	root	root
tset	root	root
tsort	root	root
tty	root	root
tzselect	root	root
ucf	root	root
ucfq	root	root
ucfr	root	root
udevadm	root	root
ul	root	root
umount	root	root
uname	root	root
uncompress	root	root

Software Process for Non-Adminstrator	UID	GID
unexpand	root	root
uniq	root	root
unlink	root	root
unshare	root	root
update-alternatives	root	root
uptime	root	root
users	root	root
utmpdump	root	root
vdir	root	root
vi	root	root
view	root	root
vim	root	root
vim.basic	root	root
vimdiff	root	root
vimtutor	root	root
vmstat	root	root
vnstat	root	root
w	root	root
wall	root	tty
watch	root	root
watchgnupg	root	root
wc	root	root
wdctl	root	root
wget	root	root
whereis	root	root
which	root	root
whiptail	root	root
who	root	root
whoami	root	root
wpa_passphrase	root	root
write	root	root
write.ul	root	tty
xargs	root	root
xsubpp	root	root
xxd	root	root
yes	root	root
ypdomainname	root	root
zcat	root	root
zcmp	root	root
zdiff	root	root
zdump	root	root
zegrep	root	root
zfgrep	root	root
zforce	root	root
zgrep	root	root
zipdetails	root	root
zless	root	root
zmore	root	root
znew	root	root