Moxa Industrial Linux 3.3 (Debian 11) Manual for VM-1220-T

Version 1.0, December 2024

www.moxa.com/products

Moxa Industrial Linux 3.3 (Debian 11) Manual for VM-1220-T

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.

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 a serial console or Ethernet cable, or remotely via Secure Shell (SSH). Refer to the Hardware Manual to set up the physical connections.

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

The username and password are the same for all active serial consoles and SSH remote logs. You must manually create a password before you can log in with the root account. Because user "**moxa**" is part of the sudo group, you can run system-level commands as that user with **sudo** commands. 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 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

Connecting to the Arm-based computer via terminal software in Linux and Windows environments is shown below.

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
   ```

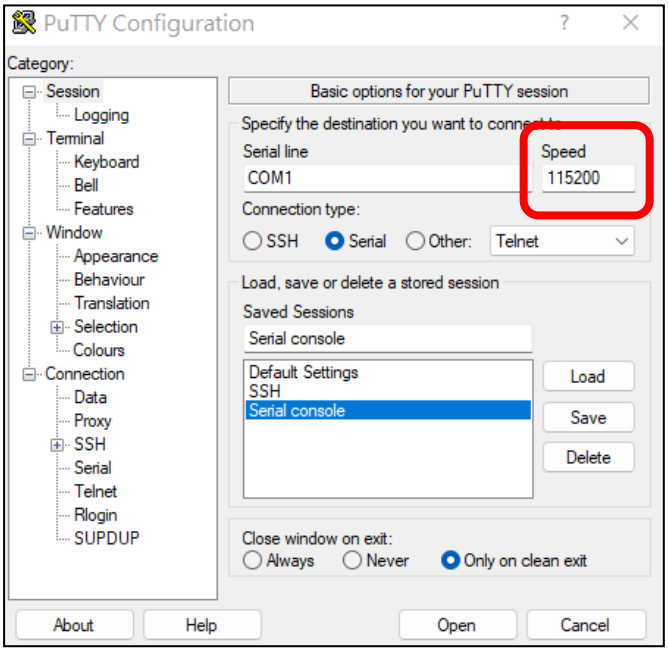
Windows Users

NOTE

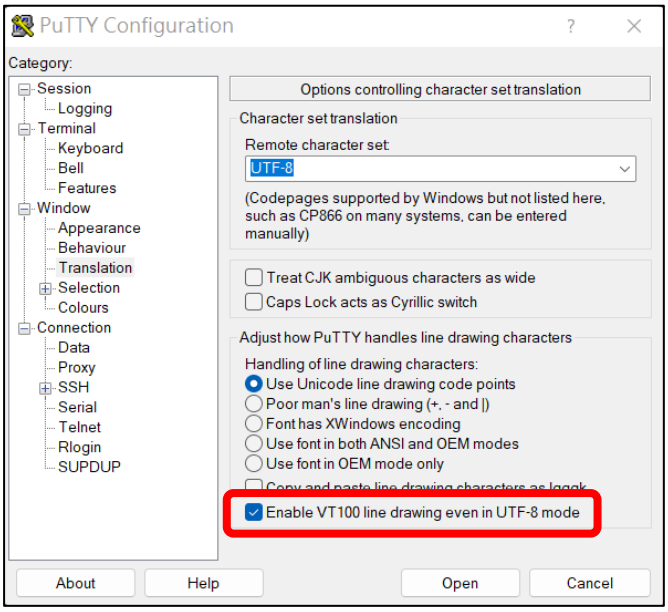
These steps apply to the Windows PC you are using to connect to the Arm-based computer. Do NOT apply these steps to the Arm-based computer itself.

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 3	Static IP	192.168.5.127
LAN 4	Static IP	192.168.6.127

*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 the DHCP server from LAN1.

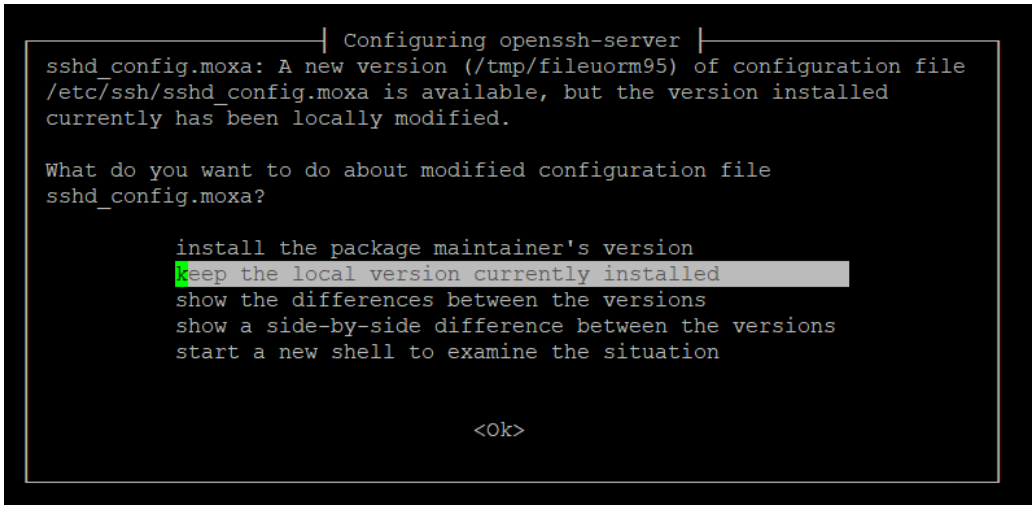
ATTENTION

Regenerate SSH key regularly

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.**" The following is prompted during the rekey process.



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

https://wiki.debian.org/SSH

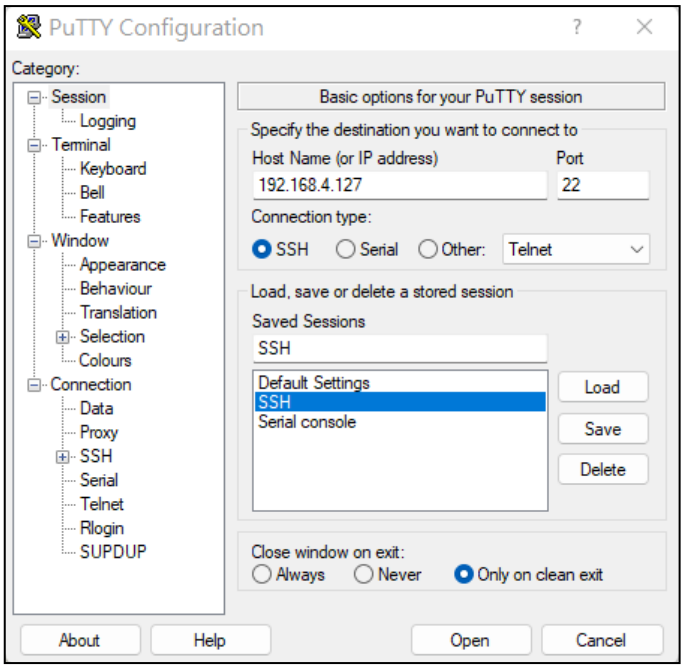
Windows Users

NOTE

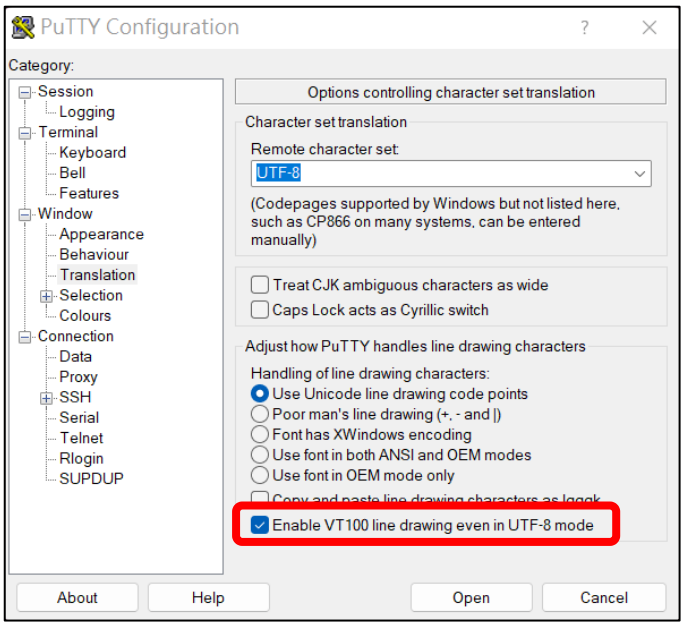
These steps apply to the Windows PC you are using to connect to the Arm-based computer. Do NOT apply these steps to the Arm-based computer itself.

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 must set a new password before logging in again.

- 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, refer to the bootloader configuration section.

Creating and Deleting User Accounts

ATTENTION

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

Use the **useradd** and **userdel** commands to create and delete user accounts. Be sure to reference the main page of these commands to set relevant access privileges for the account. The 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

Use the **usermod** commands to create and change 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

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

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

Changing the Password

Use the **passwd** commands to change the password of a user account. Changing the password will not impact 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 System Image Version

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

```
moxa@moxa-imoxa0000b08:/$ mx-ver
VM-1220-T MIL3 version 1.0.0 Build 24100806
```

```
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
       -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-imoxa0000b08:/$ mx-interface-mgmt deviceinfo
SERIALNUMBER=IMOXA0000B08
MODELNAME=VM-1220-T
SECUREBOOT=Disabled
```

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-imoxa0000b08:/$ sudo df -h
Filesystem      Size  Used Avail Use% Mounted on
udev            916M     0  916M   0% /dev
tmpfs           940M  1.3M  938M   1% /run
/dev/mmcblk0p2  982M  258M  669M  28% /boot_device/p2
/dev/mmcblk0p3  5.0G   66M  4.7G   2% /boot_device/p3
/dev/mmcblk0p4  974M   50M  857M   6% /var/log
/dev/loop0      258M  258M     0 100% /boot_device/p2/lower
overlay         5.0G   66M  4.7G   2% /
/dev/mmcblk0p1  115M   31M   76M  29% /boot_device/p1
tmpfs           940M     0  940M   0% /dev/shm
tmpfs           5.0M     0  5.0M   0% /run/lock
tmpfs           188M     0  188M   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-imoxa0000b08:/$ 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 by Moxa Arm-based computers. Before referring to the sections in this chapter, ensure that they apply to and are supported by the hardware specification of your Arm-based computer.

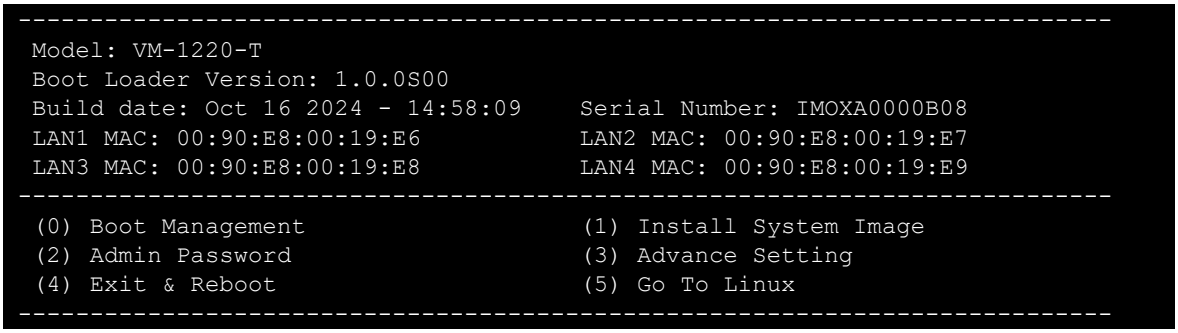
Bootloader Configuration

Accessing the Bootloader Configuration Menu

To access the bootloader menu, first connect to the 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.)



Boot Management

Boot Option

By default, Moxa Arm-based computers boot up from the embedded eMMC flash. It also provides an option to boot up from an external SD card.

The following is an example of changing the first boot to an SD card and setting the secondary boot option to an 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 the SD card; if that fails, they will boot from eMMC.

4. Set the External Storage on the SD card

```
--------------------------------------------------------------------------------
 Model: VM-1220-T
 Boot Loader Version: 1.0.0S00
 Build date: Oct 16 2024 - 14:58:09    Serial Number: IMOXA0000B08
 LAN1 MAC: 00:90:E8:00:19:E6           LAN2 MAC: 00:90:E8:00:19:E7
 LAN3 MAC: 00:90:E8:00:19:E8           LAN4 MAC: 00:90:E8:00:19:E9
--------------------------------------------------------------------------------
 (0) Set to Default                    (1) Boot Option
 (2) Advance Boot Option               (3) View Current Setting
--------------------------------------------------------------------------------
Command>>1
Boot Management : Boot Option
Boot Order : External First
Embedded Storage : eMMC
External Storage : USB

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, 2 - USB (0-2, Enter to abort): 1
Saving Environment to SPIFlash... Erasing SPI flash...Writing to SPI
flash...done
Valid environment: 2
OK

Boot Management : Boot Option
Boot Order : External First
Embedded Storage : eMMC
External Storage : SD

Set ok.
INFO.bootcfg, Set boot from SD ok
```

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 card	Boot from the external storage
0 – Embedded First	1 – eMMC	1 – SD card	First boot from eMMC; if it fails, boot from the external storage
1 – External First	1 – eMMC	1 – SD card	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 file system.
- **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_VM-
1220_MIL3_V1.0.0_Build_24100806_ImageBuild_241008_085602.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
VM-1220-T MIL3 version 1.0.0 Build 24100806
```

Installing the System Image From SD

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

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

```
--------------------------------------------------------------------------------
 Model: VM-1220-T
 Boot Loader Version: 0.2.0S00
 Build date: Aug 13 2024 - 13:38:13    Serial Number: IMOXA0000B08
 LAN1 MAC: 00:90:E8:00:19:E6           LAN2 MAC: 00:90:E8:00:19:E7
 LAN3 MAC: 00:90:E8:00:19:E8           LAN4 MAC: 00:90:E8:00:19:E9
--------------------------------------------------------------------------------
 (0) Install System Image from TFTP    (1) Install System Image from SD
 (2) Install System Image from USB     (3) TFTP Settings
--------------------------------------------------------------------------------
Command>>1
System image File Name (System_image.img): IMG_VM-
1220_MIL3_V1.0.0_Build_24100806_ImageBuild_241008_085602.img
```

NOTE

1. Make sure to put **the hash file of the system image** in the same folder as an image, as integrity validation is required.
2. The SD format supports FAT32 or ext4 in cluster size ≤ 32 KB.

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

```
moxa@moxa-tbzkb1090923:# sudo mx-ver
VM-1220-T MIL3 version 1.0.0 Build 24100806
```

Administrator Password

Enabling/Disabling Admin Password

Password do not protect the bootloader menu by default. To enhance the security of your Moxa Arm-based computer, it is strongly recommended to set up an administrator password if physical unauthorized access is possible. To set up an administrator password, follow the below procedures:

1. Select **Admin Password > Enable/Disable Admin Password**.
2. Select **1** to set up an administrator password. If **0** (disable) is selected, the currently set password will be cleared.
3. Enter the password you would like to set twice; the password strength requirement is at least 8 characters in length.

```
--------------------------------------------------------------------------------
 Model: VM-1220-T
 Boot Loader Version: 0.2.0S00
 Build date: Aug 13 2024 - 13:38:13    Serial Number: IMOXA0000B08
 LAN1 MAC: 00:90:E8:00:19:E6           LAN2 MAC: 00:90:E8:00:19:E7
 LAN3 MAC: 00:90:E8:00:19:E8           LAN4 MAC: 00:90:E8:00:19:E9
--------------------------------------------------------------------------------
 (0) Set to Default                    (1) Enable/Disable Admin Password
 (2) Configure Admin Password          (3) Configure Admin Password Policy
--------------------------------------------------------------------------------
Command>>1
Current Mode: Disabled
```

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

The current password is empty, please set one.

Admin Password Policy:
 - Minimum length: 8

Enter new password: **********
Retype password: **********
Password set successfully

Password status : Enabled.
INFO.secure, Admin password enabled
```

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

```
SoC:   HS-FS
Model: VM-1220-T
DRAM:  2 GiB
WDT:   Started with servicing (15s timeout)
MMC:   mmc@fa10000: 0, mmc@fa00000: 1
Loading Environment from SPIFlash... SF: Detected mx25l12805d with page size
256 Bytes, erase size 4 KiB, total 16 MiB
OK
In:    serial@2800000
Out:   serial@2800000
Err:   serial@2800000
Net:   eth0: ethernet@8000000port@1, eth1: ethernet@8000000port@2
Date: 2024-08-16 (Friday)    Time:  5:42:34
RTC: OK
tpm@0 v2.0: [open]
Press <DEL> To Enter BIOS configuration Setting:  0

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 the bootloader menu is needed, you will have to contact Moxa technical support to send your Arm-based computer to Moxa to reset the password.

Configuring the Admin Password Policy

To change the administrator password, select **Admin Password > Configure Admin Password** and follow the on-screen instructions. Changing the password will not impact functionalities.

```
--------------------------------------------------------------------------------
 Model: VM-1220-T
 Boot Loader Version: 0.2.0S00
 Build date: Aug 13 2024 - 13:38:13    Serial Number: IMOXA0000B08
 LAN1 MAC: 00:90:E8:00:19:E6           LAN2 MAC: 00:90:E8:00:19:E7
 LAN3 MAC: 00:90:E8:00:19:E8           LAN4 MAC: 00:90:E8:00:19:E9
--------------------------------------------------------------------------------
 (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): 8

- 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 follow the on-screen instructions

Resetting the Admin Password to Default

If you lost your password, follow these 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. Press and hold the **RESET** button on the Moxa Arm-based computer for over 5 seconds to trigger the password reset process. You must complete this step within **10 seconds** after step one for the reset process to start.

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 the bootloader configuration menu when failure login attempts and the time are over the defined threshold.

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

```
--------------------------------------------------------------------------------
 Model: VM-1220-T
 Boot Loader Version: 0.2.0S00
 Build date: Aug 13 2024 - 13:38:13    Serial Number: IMOXA0000B08
 LAN1 MAC: 00:90:E8:00:19:E6           LAN2 MAC: 00:90:E8:00:19:E7
 LAN3 MAC: 00:90:E8:00:19:E8           LAN4 MAC: 00:90:E8:00:19:E9
--------------------------------------------------------------------------------
 (0) Set to Default                    (1) Configure Auto Reboot
 (2) Configure Login Message           (3) Configure Invalid Login Attempts
 (4) Clear TPM                         (5) Configure Linux Security Modules
 (6) Enable/Disable Interfaces         (7) View Bootloader log
--------------------------------------------------------------------------------
Command>>3

Current setting: [5] consecutive invalid login within [60] seconds will reboot
and disable access to bootloader menu for [900] 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 below configuration will disable invalid login check

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

Consecutive Invalid Login Attempts

Configuration	Setting	Factory Default
Consecutive invalid login attempts	Input from 0 to 5	0
Within how many Seconds	Input from 0 to 60	0
Disable access to how many seconds	Input from 0 to 900	0

NOTE

Inputting 0 to any of the above configurations will disable the invalid login check.

Auto Reboot After Inactivity

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

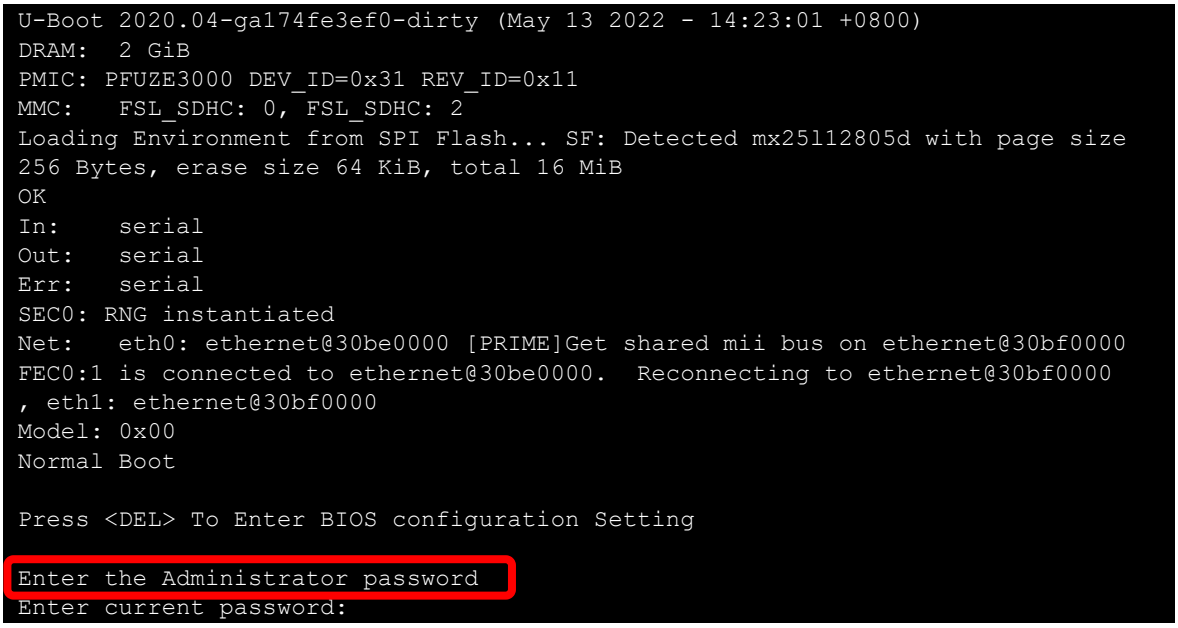
To set the time, select **(2) Advance Setting > (1) Configure Auto Reboot** and follow the on-screen instructions.

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

Login Banner Message

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

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



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 duringstartup.

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

Clearing the TPM Module

Clearing the TPM will erase information stored in 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 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.

Changing the Default Hostname

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

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

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 shows 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 the 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, adjusting the hardware clock for Daylight Saving Time is unnecessary.

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

Hours From Greenwich Mean Time (GMT)	Value	Description
-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.

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)

Frequently, 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). However, configuring the serial port baud rate is not possible in MCIM because Linux provides a standard method for setting the baud rate.

MCIM is a command-line interface (CLI) Moxa utility designed to access and manage Moxa Arm-based computer interfaces. Use the **`# 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 the **# mx-interface-mgmt led** command to get the list of controllable LEDs on your Arm-based computer. In the following example, the returned NAME "L1" refers to the LED for cellular signal. For **LEDs** with **multiple colors** such as RDY (green and red), 2 LED names will appear (RDY_Green and RDY_Red). For this type of LEDs, you must set the state of a color to "off" before setting another color to "on" or "heartbeat".

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

NAME        LABEL                    STATE  ALIAS
L1          L1:green:signal          off    N/A
L2          L2:green:signal          off    N/A
L3          L3:green:signal          off    N/A
RDY_Green   RDY:green:status         on     RDY
RDY_Red     RDY:red:alarm            off    ALARM
SIM         SIM:green:status         off    N/A
USR1        USR1:green:programming   off    N/A
USR2        USR2:green:programming   off    N/A
USR3        USR3:green:programming   off    N/A
USR4        USR4:green:programming   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>***	Shows 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. Use the alias in place of <led_name>.

An example of changing the current state of RDY LED from **green** (steady) to **red** (heartbeat) is given below:

```
moxa@moxa-imoxa0000b08:/$ sudo mx-interface-mgmt led RDY_Green
NAME=RDY_Green
LABEL=RDY:green:status
STATE=on
ALIAS=RDY
moxa@moxa-imoxa0000b08:/$ sudo mx-interface-mgmt led RDY_Green set_state off
moxa@moxa-imoxa0000b08:/$ sudo mx-interface-mgmt led RDY_Red set_state
heartbeat
```

Storage and Partitions

Use **`# mx-interface-mgmt disk and # 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, SD card) • 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, SD card) • 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., SD card) 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 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, SD_p1) • Device node (e.g., /dev/mmcblk0p1) • Partition mounted (Y/N) • Partition mount point (e.g., /boot_device/p1) • File system (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., SD card) using LUKS. The encrypted disk will only be mountable on a Moxa computer with the corresponding LUKS key. file *Note: You will be prompted to set a minimum 8-character password. This password can be used to recreate the LUKS key file if needed.* **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 the 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
`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 the command history.
`partition <partition_name> remap_luks -i <password>`	Performs the above remapping function, but with the password provided as a parameter, bypassing the password prompt

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

```
moxa@moxa-imoxa0000b08:/$ mx-interface-mgmt disk
NAME DEVICE       SYSTEM_DISK NUMBER_OF_PARTITIONS AUTOMOUNT_SETTING IO_STATE
SD   /dev/mmcblk1 N           1                    false              enabled
eMMC /dev/mmcblk0 Y           4                    false              N/A
moxa@moxa-imoxa0000b08:/$ sudo mx-interface-mgmt disk SD set_automount true
```

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

```
moxa@moxa-imoxa0000b08:/$ sudo mx-interface-mgmt partition
NAME     DEVICE          IS_MOUNTED  FS_TYPE  MOUNTPOINT       MAPPER_DEVICE
SD_p1    /dev/mmcblk1p1  N           N/A      N/A              N/A
eMMC_p1  /dev/mmcblk0p1  Y           ext4     /boot_device/p1  N/A
eMMC_p2  /dev/mmcblk0p2  Y           ext4     /boot_device/p2  N/A
eMMC_p3  /dev/mmcblk0p3  Y           ext4     /boot_device/p3  N/A
eMMC_p4  /dev/mmcblk0p4  Y           ext4     /boot_device/p4  N/A

moxa@moxa-imoxa0000b08:/$ sudo mx-interface-mgmt partition SD_p1 mount
moxa@moxa-imoxa0000b08:/$ sudo mx-interface-mgmt partition SD_p1
NAME=SD_p1
DEVICE=/dev/mmcblk1p1
IS_MOUNTED=Y
FS_TYPE=exfat
MOUNTPOINT=/media/SD_p1
```

WARNING

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

Creating an Encrypted SD Card

Below is an example of how to create an encrypted SD card:

1. Insert an SD card and use **mx-interface-mgmt partition** command to check the name of the available partition.

   ```
   moxa@moxa-imoxa0920070:/home/moxa# sudo mx-interface-mgmt partition
   NAME     DEVICE          IS_MOUNTED  FS_TYPE  MOUNTPOINT       MAPPER_DEVICE
   SD_p1    /dev/mmcblk1p1  N           N/A      N/A              N/A
   eMMC_p1  /dev/mmcblk0p1  Y           ext4     /boot_device/p1  N/A
   eMMC_p2  /dev/mmcblk0p2  Y           ext4     /boot_device/p2  N/A
   eMMC_p3  /dev/mmcblk0p3  Y           ext4     /boot_device/p3  N/A
   eMMC_p4  /dev/mmcblk0p4  Y           ext4     /boot_device/p4  N/A
   ```

2. Select a partition on the SD card to encrypt and set a password with a minimum length of 8 characters.

   ```
   moxa@moxa-imoxa0000b08:~$ sudo mx-interface-mgmt partition SD_p1
   initialize_luks
   [Warning]: Initializing a partition as LUKS will erase all data on the
   partition.
   Enter password:
   Re-enter password:
   ```

3. Now, SD_p1 is LUKS encrypted, and the corresponding LUKS key file is securely hashed using SHA-512 and stored on this computer. As a result, SD_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 SD card on a different Moxa computer with MIL3.

   ```
   moxa@moxa-imoxa0000b08:~# sudo mx-interface-mgmt partition SD_p1 mount
   Error: GDBus.Error:com.moxa.ComputerInterfaceManager.Error.Core.Failed: LUKS
   open process failed: cannot get passphrase from config
   ```

```
moxa@moxa-imoxa0000b08:~# sudo mx-interface-mgmt partition SD_p1 remap_luks
Enter password:
root@moxa-imoxa0000b08:~# sudo mx-interface-mgmt partition SD_p1 mount
[  257.929778] EXT4-fs (dm-0): mounted filesystem with ordered data mode.
Opts: (null)
moxa@moxa-imoxa0000b08:~# mx-interface-mgmt partition
NAME    DEVICE         IS_MOUNTED FS_TYPE MOUNTPOINT     MAPPER_DEVIC UUID
SD_p1   /dev/mmcblk1p1 Y          ext4    /media/SD_p1   mmcblk1p1_encrypted
8cc44ae3-73d...
eMMC_p1 /dev/mmcblk0p1 Y          ext4    /boot_device/p1 N/A f658ef6b-
6db...
eMMC_p2 /dev/mmcblk0p2 Y          ext4    /boot_device/p2 N/A e702dbfa-
651...
eMMC_p3 /dev/mmcblk0p3 Y          ext4    /boot_device/p3 N/A 3001336d-
7d0...
eMMC_p4 /dev/mmcblk0p4 Y          ext4    /boot_device/p4 N/A 4614e6e9-
f7a...
```

Serial Port

Configuring the Serial Interface via MCIM

In VM-1220-T, the serial ports from P1 to P3 support RS-485 2-wire; the P4 supports both RS-232 (default) and RS-485 2-wire with flexible baudrate settings.

Use the **`# mx-interface-mgmt serialport`** command to query and configure the operation mode for the serial ports.

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)
`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 <serial_io_state>`	Sets the serial port enabled or disabled. *Note: Changing the I/O state requires a system reboot*

To change the configuration of the pull high/low resistor or the termination, refer to the VM-1220-T hardware manual to switch the jumper or the slide switch.

Changing the Serial Port Operation Mode

For example, to change the mode of 4 P4 serial port from default RS-232 mode to the RS-485 2-wire mode, use the following command:

```
moxa@moxa-imoxa0000b08:~$ mx-interface-mgmt serialport
NAME  DEVICE      INTERFACE  IO_STATE  PULL_UP_DOWN_RESISTOR  TERMINATOR
P1    /dev/ttyS3  RS-485-2W  enabled   N/A                    N/A
P2    /dev/ttyS4  RS-485-2W  enabled   N/A                    N/A
P3    /dev/ttyS5  RS-485-2W  enabled   N/A                    N/A
P4    /dev/ttyS6  RS-232     enabled   N/A                    N/A
moxa@moxa-imoxa0000b08:~$ mx-interface-mgmt serialport P4

NAME=P4
DEVICE=/dev/ttyS6
```

```
SUPPORTED_INTERFACES=RS-232,RS-485-2W
SUPPORTED_BAUDRATES=300,600,1200,1800,2400,4800,9600,19200,38400,57600,115200
INTERFACE=RS-232
IO_STATE=enabled
PULL_UP_DOWN_RESISTOR=N/A
TERMINATOR=N/A

moxa@moxa-imoxa0000b08:~$ sudo mx-interface-mgmt serialport P4 set_interface
RS-485-2W
moxa@moxa-imoxa0000b08:~$ mx-interface-mgmt serialport P4 get_interface
RS-485-2W
```

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-imoxa0000b08:~$  mx-interface-mgmt serialport
NAME  DEVICE      INTERFACE  IO_STATE  PULL_UP_DOWN_RESISTOR  TERMINATOR
P1    /dev/ttyS3  RS-485-2W  enabled   N/A                    N/A
P2    /dev/ttyS4  RS-485-2W  enabled   N/A                    N/A
P3    /dev/ttyS5  RS-485-2W  enabled   N/A                    N/A
P4    /dev/ttyS6  RS-232     enabled   N/A                    N/A

moxa@moxa-imoxa0000b08:~$ sudo stty -a -F /dev/ttyS3
speed 9600 baud; rows 0; columns 0; line = 0;
intr = ^C; quit = ^\; erase = ^?; kill = ^U; eof = ^D; eol = <undef>;
eol2 = <undef>; swtch = <undef>; start = ^Q; stop = ^S; susp = ^Z; rprnt = ^R;
werase = ^W; lnext = ^V; discard = ^O; min = 1; time = 0;
-parenb -parodd -cmspar cs8 hupcl -cstopb cread clocal -crtscts
-ignbrk -brkint -ignpar -parmrk -inpck -istrip -inlcr -igncr icrnl ixon -ixoff
-iuclc -ixany -imaxbel -iutf8
opost -olcuc -ocrnl onlcr -onocr -onlret -ofill -ofdel nl0 cr0 tab0 bs0 vt0 ff0
isig icanon iexten echo echoe echok -echonl -noflsh -xcase -tostop -echoprt
echoctl echoke -flusho -extproc
```

Configuring Serial Settings

The following example changes the baudrate to 115200.

```
moxa@moxa-imoxa0000b08:~$ sudo stty 115200 -F /dev/ttyS3
```

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

```
moxa@moxa-imoxa0000b08:~$ sudo stty -a -F /dev/ttyS3
speed 115200 baud; rows 0; columns 0; line = 0;
intr = ^C; quit = ^\; erase = ^?; kill = ^U; eof = ^D; eol = <undef>;
eol2 = <undef>; swtch = <undef>; start = ^Q; stop = ^S; susp = ^Z; rprnt = ^R;
werase = ^W; lnext = ^V; discard = ^O; min = 1; time = 0;
-parenb -parodd -cmspar cs8 hupcl -cstopb cread clocal -crtscts
-ignbrk -brkint -ignpar -parmrk -inpck -istrip -inlcr -igncr icrnl ixon -ixoff
-iuclc -ixany -imaxbel -iutf8
opost -olcuc -ocrnl onlcr -onocr -onlret -ofill -ofdel nl0 cr0 tab0 bs0 vt0 ff0
isig icanon iexten echo echoe echok -echonl -noflsh -xcase -tostop -echoprt
echoctl echoke -flusho -extproc
```

NOTE

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

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

Ethernet Interface

Use **# mx-interface-mgmt ethernet** command to display information on the Ethernet ports.

Command and Usage	Description
ethernet	Shows the following information of **all** Ethernet ports on the device. • Name (as labeled on device) • Network interface name (eth0, eth1, etc.)
ethernet <ethernet_name>	Shows the above information of a **specified** Ethernet port
ethernet **<ethernet_name>** set_io_state **<io_state>**	Sets 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-imoxa0000b08:~$ mx-interface-mgmt ethernet
NAME   DEVICE_NAME   IO_STATE
LAN1   eth0          enabled
LAN2   eth1          enabled
LAN3   eth2          enabled
LAN4   eth3          enabled

moxa@moxa-imoxa0000b08:~$ mx-interface-mgmt ethernet LAN1
NAME=LAN1
DEVICE_NAME=eth0
IO_STATE=enabled
```

Serial Console Interface

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

Command and Usage	Description
console	Shows the following information for the console port. • Name (as labeled on the device) • Device node (e.g., /dev/ttyS2)
console <console_name>	Shows the above information of a specified serial console interface
Console <console_name> set_io_state <io_state>	Sets 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.

The following is an example of showing the console port device node:

```
moxa@moxa-imoxa0000b08:~$ mx-interface-mgmt console
NAME      DEVICE       IO_STATE
Console   /dev/ttyS2   enabled
```

Digital Input/Output (DIO)

Use the **# mx-interface-mgmt dio** command to query and configure the state for each digital input/output (DIO) interface, and 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 (change/falling/none/rising) • Activation of script (yes/no) • Direction (input/output)
dio <dio_name>	Shows the above information of a **specified** DI, DO, or DIO interface
dio <dio_name> get_state	Gets the current state (high/low) of a **specified** DI, DO, or DIO interface
dio <dio_name> set_state <dio_state>	Sets the state (high/low) of a **specified** DO interface
dio <dio_name> get_event	Gets the trigger condition for executing a script of a specified DI
dio <dio_name> set_event <di_event>	Sets the event condition (none, falling, rising, or change) for executing a script of a specified DI
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 the DIO interface.
dio <dio_name> reload	Reloads the DIO configuration after using the set_event or set_direction command

NOTE

- The default state of the digital output is high (open circuit).
- When the system performs a warm reboot, the state of the digital output remains the same.
- When the system performs a cold reboot, the state of the digital output will be changed to default - high (open circuit).

NOTE

- The default direction of the DIO interface is *input*.
- When the system performs a warm reboot, the direction of the DIO remains the same.
- When the system performs a cold reboot, the direction of the DIO remains the same.

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' :

```
root@moxa-tbzgb1057611:/etc/moxa/MoxaComputerInterfaceManager/dio-scripts#
vim 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. Use **# mx-interface-mgmt dio <dio_name> set_event <dio_event>** to set the event value for [DI1] to rising edges:
5. root@moxa-tbzgb1057611: mx-interface-mgmt dio DI1 set_event rising. Apply the changes by reloading the command:

```
root@moxa-tbzgb1057611: mx-interface-mgmt dio DI1 reload
```

Do not interrupt until the MoxaComputerInterfaceManager is fully restarted.

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**':

```
NAME   STATE   EVENT    ACTIVE   GPIO_PIN   DIRECTION
DI0    high    none     no       496        input
DI1    low     change   yes      497        input
DI2    high    none     no       498        input
DI3    high    none     no       499        input
DI4    high    none     no       500        input
DI5    high    none     no       501        input
DI6    high    none     no       502        input
DI7    high    none     no       503        input
DI8    high    none     no       480        input
DI9    high    none     no       481        input
DI10   high    none     no       482        input
DI11   high    none     no       483        input
DI12   high    none     no       484        input
DI13   high    none     no       485        input
DI14   high    none     no       486        input
DI15   high    none     no       487        input
DIO0   high    none     no       488        output
DIO1   high    none     no       489        input
DIO2   high    none     no       490        input
DIO3   high    none     no       491        input
DO0    high    none     no       504        output
DO1    high    none     no       505        output
DO2    high    none     no       506        output
DO3    high    none     no       507        output
```

Cellular Module Interface

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

Command and Usage	Description
cellular	Shows the following information for **all** cellular modules. • Name (e.g., Cellular1) • Network interface name (wwan0, wwan1, etc.) • Cellular module detected (true/false)
cellular <name>	Shows the detailed 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 # corresponds to the labeled slot # on the device*
cellular <name> get_power	Gets the cellular module power status (on/off)
cellular <name> set_power <power_state>	Sets the cellular module power status (on/off) *Note: Module will power on when the device reboots*
cellular <name> get_sim_slot	Gets the SIM slot # that is currently used by the cellular module

Command and Usage	Description
`cellular <name> set_ sim_slot <sim_slot>`	Sets 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 reboots*

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.

Socket Interface

Use the **`# mx-interface-mgmt socket`** command to 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 # corresponds 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 # 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 **(VM-1220 doesn't support terminator detection on MIL3)**
`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 <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*

Configuring the Socket CAN Interface

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

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

```
# ip link
5: can0: <NOARP,ECHO> mtu 16 qdisc noop state DOWN mode DEFAULT group default
qlen 10
    link/can
6: can1: <NOARP,ECHO> mtu 16 qdisc noop state DOWN 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

Getting the Button List and Status

Use **`# mx-interface-mgmt button`** command to display the available buttons and the button-configured actions.

Command and Usage	Description
`button`	Show the following information for **all** buttons on the device: • Name (as labeled on device) • Action (default/user-defined/disabled) ➢ Default: Button behavior is default ➢ User-defined: The button behavior has been customized by the user ➢ Disabled: The button has no function when pushed
`button <name>`	Show the above information for a **specified** button
`Button <name> get_action`	Gets the current action (default/user-defined/disabled) for a **specified** button
`Button <name> set_action <button action>`	Sets the action (default/user-defined/disabled) for a **specified** button

The following is an example of using MCIM to query an available button (**RESET** button) of the VM-1220 Series.

```
root@moxa-tbzkb1090923:~# mx-interface-mgmt button
NAME      Action
RESET     default
```

Customize the Button Action

Use the two scripts (default and custom) available in the following path to customize button actions: **/etc/moxa/MoxaComputerInterfaceManager/button-scripts/**. For example, in the VM-1220 Series, the default script is "**vm1220-default.script**" and custom script is "**custom.script**".

By default, the **RESET** button will load the default script when pressed. The default script will perform designed tasks based on the actions of the **RESET** button. The following table gives a detailed description of the default script:

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

To customize the **RESET** button action, a configuration file at **/etc/moxa/MoxaComputerInterfaceManager/peripheral-settings.conf** could be modified. The device needs to reboot for the settings to take effect. The **set_action** command can be used instead, and a reboot is not required.

The **Action** parameter in the configuration file 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
```

If **Action** is set to 2 (custom script), **/etc/moxa/MoxaComputerInterfaceManager/button-scripts/custom.script** should be edited to add the desired actions. To make it easier to configure the actions in the script file, copy the content of the default script to a custom script file and then make the required changes.

```
root@moxa-tbzkb1090923:/etc/moxa/MoxaComputerInterfaceManager/button-scripts#
cp vm-1220-default.script custom.script
```

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

NOTE

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 an IP the from DHCP server for 20 seconds, then link-local IP addresses is automatically 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
LAN3	No	Static IPv4, 192.168.5.127
LAN4	No	Static IPv4, 192.168.6.127
Cellular/ Wi-Fi	No	Not configured

To run **MCM**, you must use root permission to run **# 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 the 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

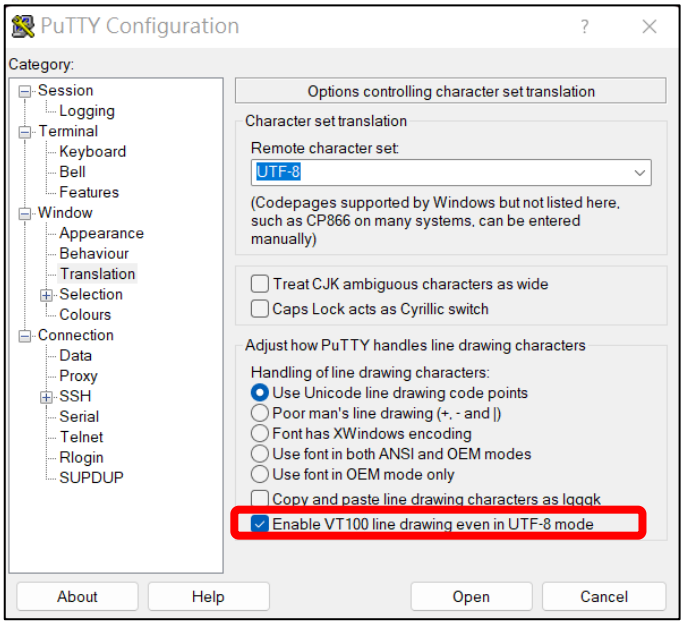
- When modification is made to the configuration file, you must use **# mx-connect-mgmt reload** to make the change effective.
- You can find the detailed configuration file structure in the "Configuration File" chapter of the Moxa Connection Manager Reference Manual.
- We highly recommend using the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

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 the **VT100 line drawing** option under **Windows > Translation** for the GUI to show correctly



1. Go to the main page.

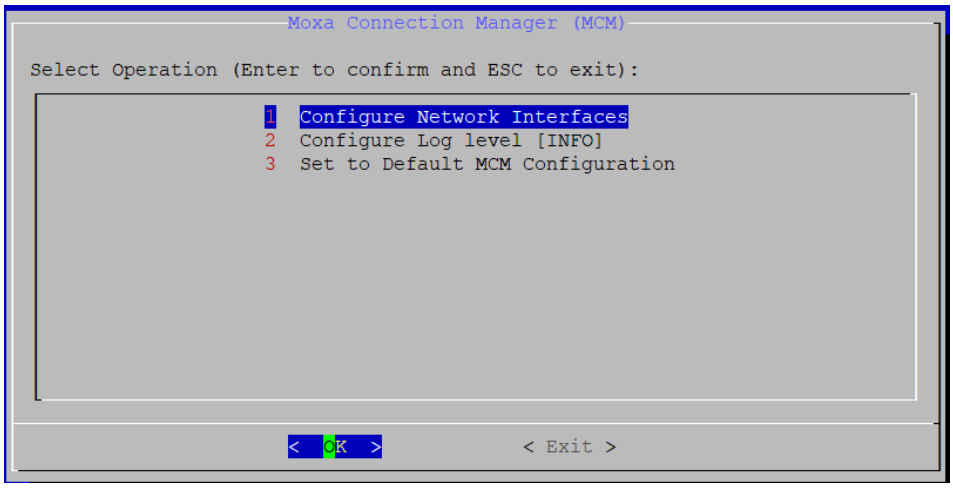


Figure 5.1—Main page

Option Name	Description
Configure Network Interface	Configure network setting for
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.

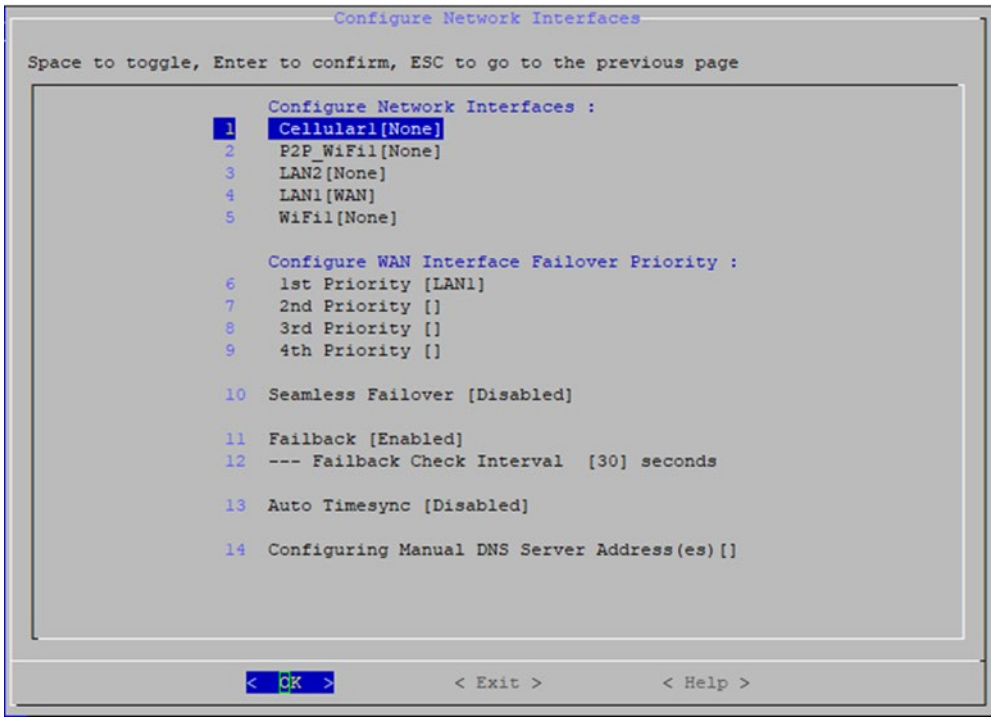


Figure 5.2—Configure network interface

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 the 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 to this interface onetime 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. • **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 switch 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 the first Priority as the default gateway. When the first priority interface becomes unavailable, MCM will automatically failover to the next priority interface.
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

Option Name	Description
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.

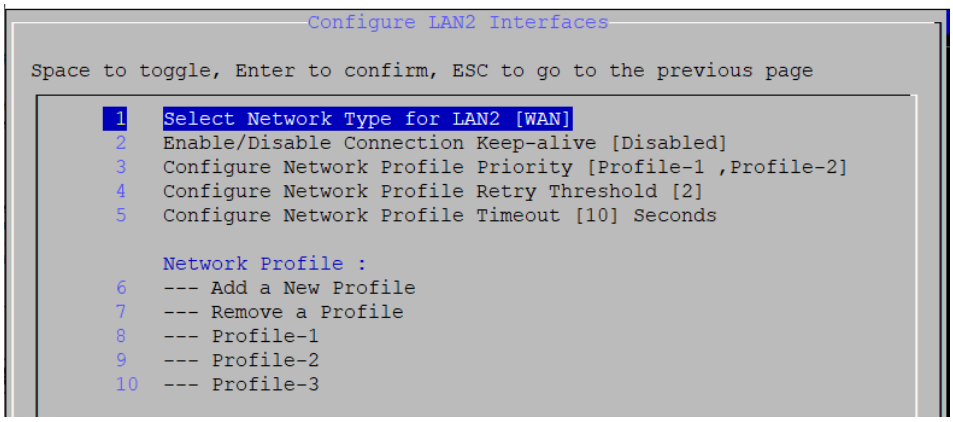


Figure 5.3—Configurable options for WAN interface

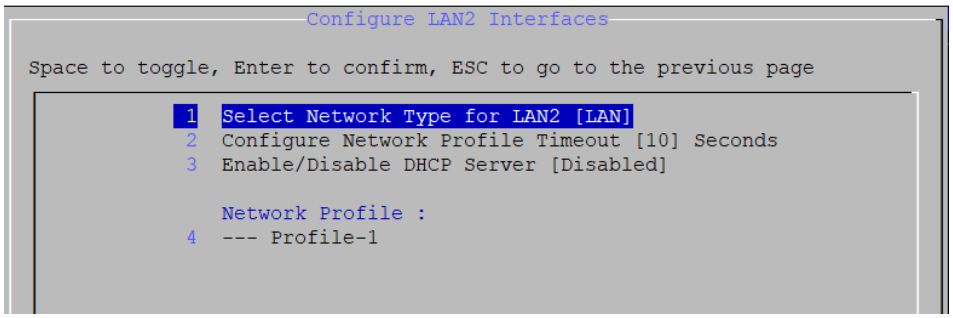


Figure 5.4—Configurable options for LAN interface

Option Name	Supported Network Type	Description
Select Network Type	All	Available options are WAN/LAN/LAN Bridge/Manual/None
Configure Network Profile Priority	WAN	When the first 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
Enable/Disable DHCP Server	LAN, LAN-bridge	Configure a specific LAN or bridged LAN interfaces as a DHCP server

Option Name	Supported Network Type	Description
Network Profile	WAN, LAN, Manual	• This section displays all network profiles in a list with an option to add, modify, or remove a profile. • If the 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.

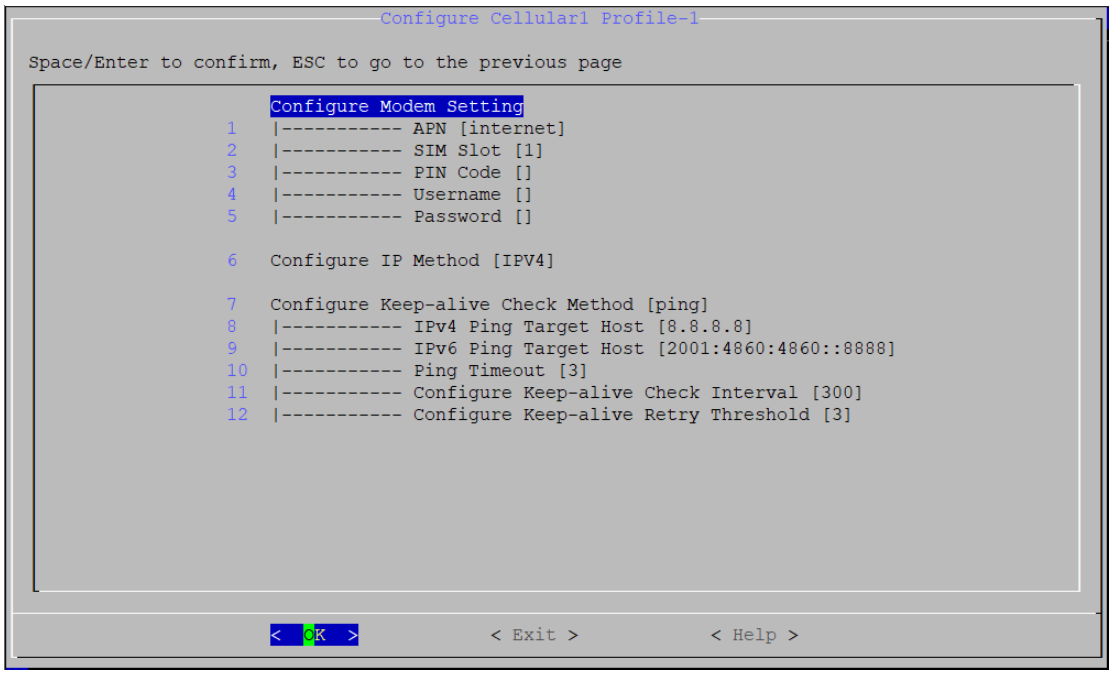


Figure 5.5—Network profile setting (cellular interface as an example)

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 if the 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 the DHCP server assigns IP to LAN port), consider the 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 supports Stateless Address Auto-Configuration (SLACC) and Stateless for DHCPv6.

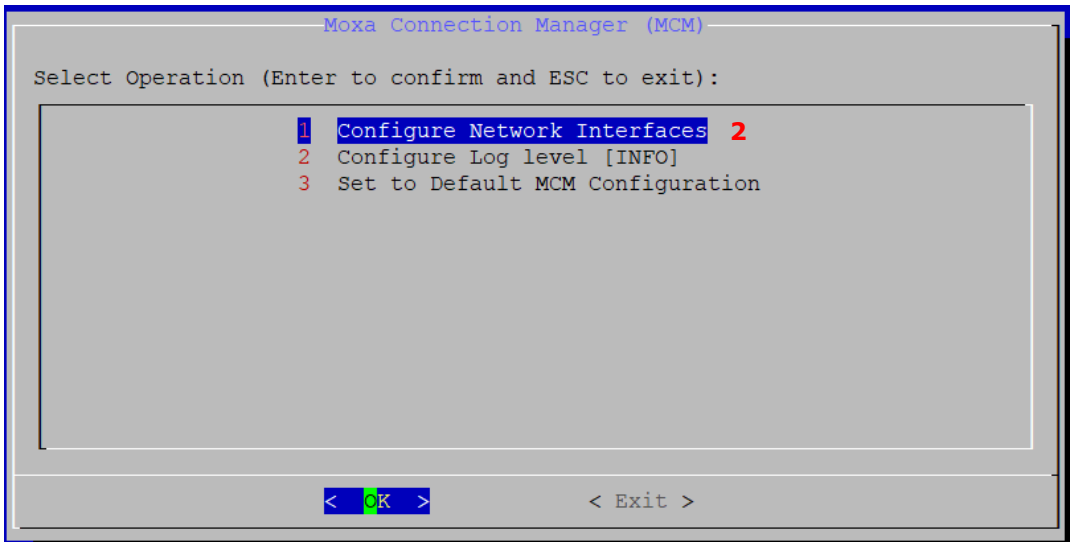
Cellular and Wi-Fi Failover/Failback

One of the key features in MCM is the WAN connection auto-failover, where you can configure multiple backup WAN networks. When the primary connection becomes unavailable, MCM will automatically failover 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 the following example, we will set the 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.

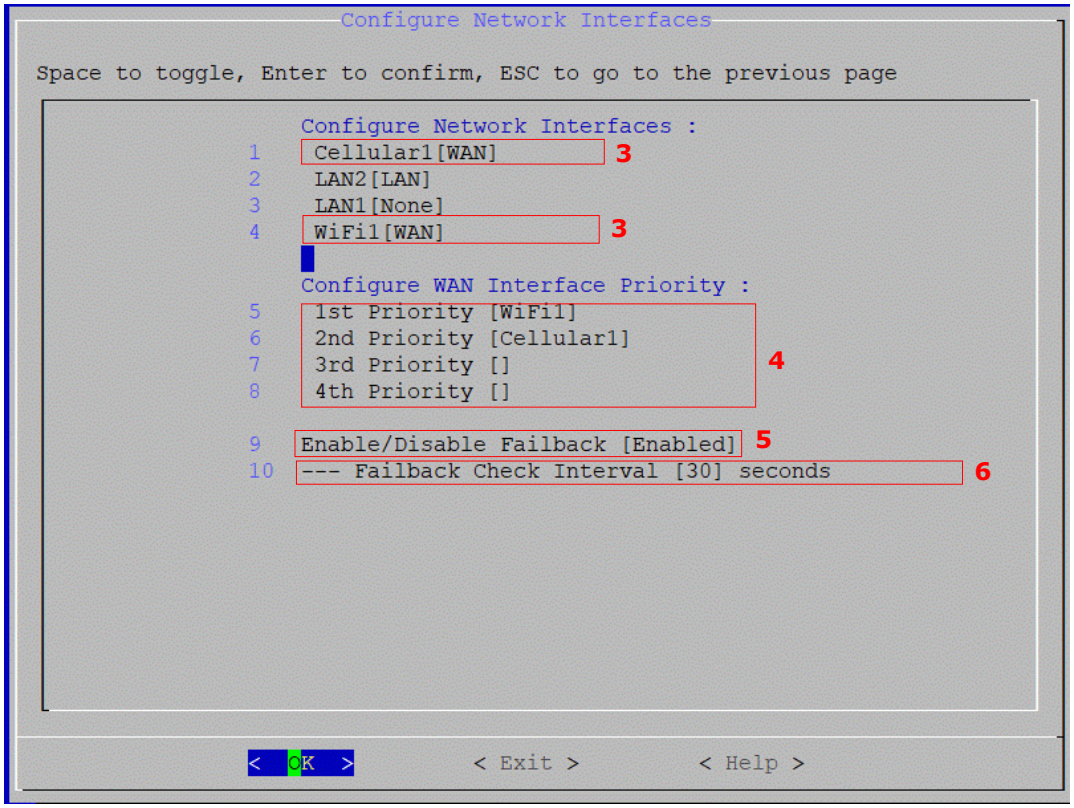
1. Run **# mx-connect-mgmt configure** to launch a simple GUI dialog configurator

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



2. Select "Configure Network Interfaces"
3. Set interface Cellular1 and WiFi1 both to WAN, and
4. Set WiFi1 as the first priority and Cellular1 as the second priority
5. Ensure Failback is enabled if you want MCM to automatically switch back to Wi-Fi from cellular when it is back online.

6. Failback Check Interval [30] seconds means MCM will make sure the 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 start the cellular connection when failover is triggered.

   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. Change the default profile-1 and profile-2 or add/remove a profile.

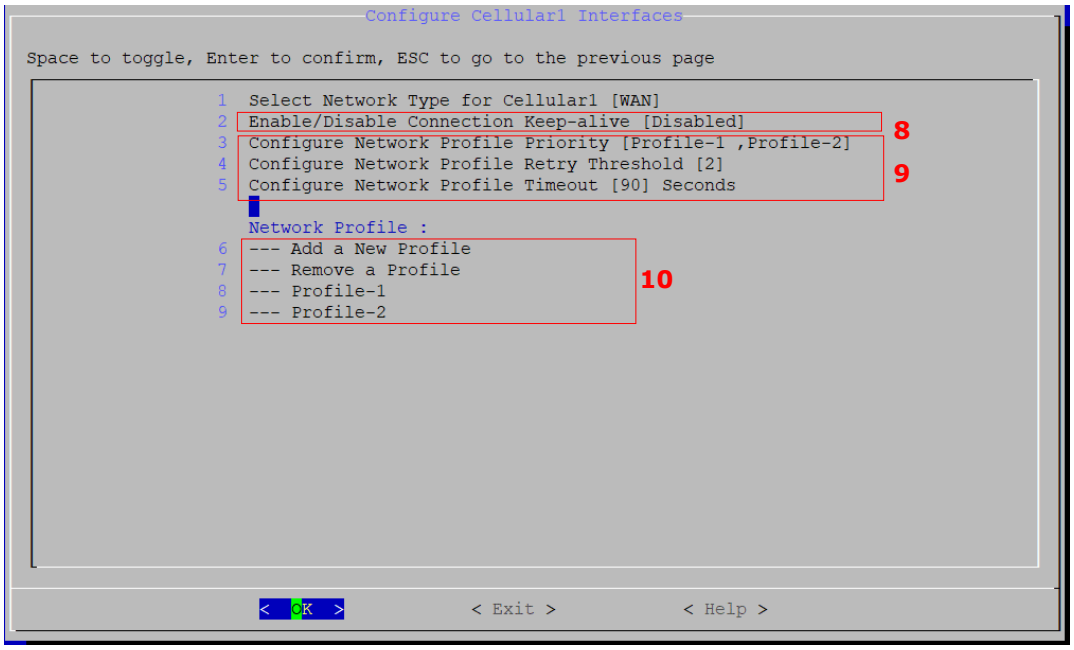


Figure 5.6—Interface configuration page of Cellular1

11. Go to the 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 determines the connection is alive. Currently, only "ping' method is supported for WAN network. In this example, the following configuration is set for Profile-1 of Cellular1 interface.
    - MCM will ping the Google DNS once every 700 seconds.
    - MCM will try to ping the target host maximum three times (Retry Threshold) before concluding profile-1 cannot connect. For each ping attempt, MCM will consider ping fails if the server doesn't respond in three seconds (Ping timeout).
15. Once completed the configuration, exit MCM and select save and reload configuration file for the configuration to take effect

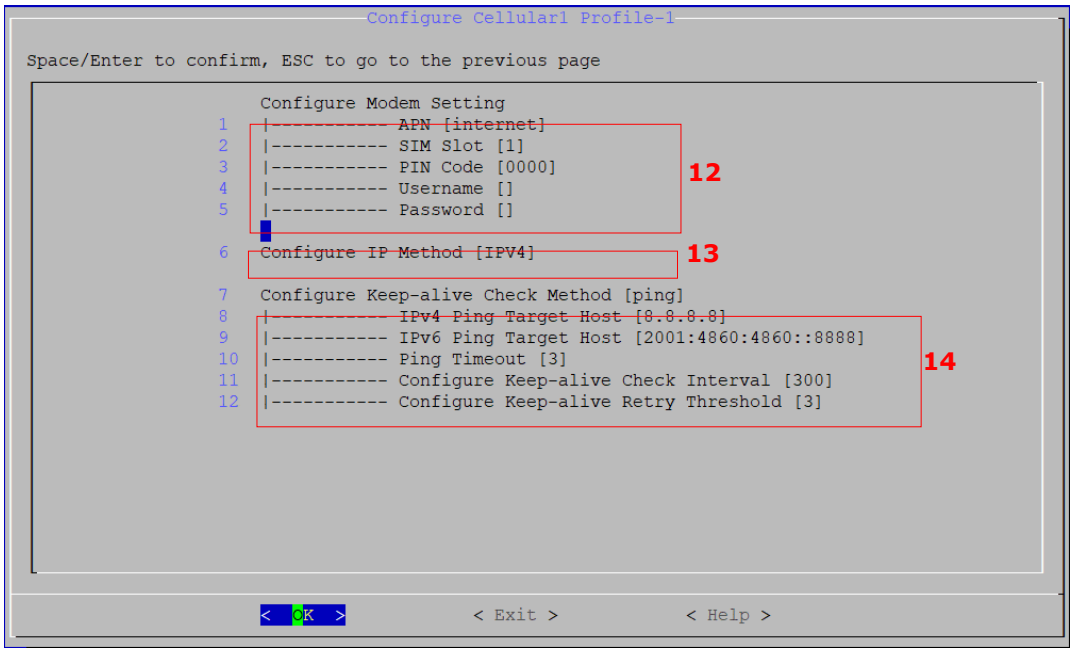


Figure 5.7—Network profile configuration page of Cellular1 interface

Checking the Network Status

Checking the Interface and Connection Status

- Use **# mx-connect-mgmt nwk_info [Interface name]** to check the interface and connection status
- Use **# 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. Additional details to some of the data fields are:

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

Fields	Description	Available Interface
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, 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 determined by the Keep-alive Check Method in Figure 5.5. For example, if the method is set to ping, the connection is considered 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 the default route • **False:** This interface is not the default route	Wi-Fi, Ethernet, Cellular

Cellular Signal Strength

Signal Indicator

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-to-Noise Ratio): Measures the quality of the reference signal by evaluating the signal-to-noise ratio in dB.

Signal Level Criteria

Below are the criteria that MCM uses to determine the signal strength for 4G(LTE):

- 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.

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

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 temporarily 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 on downgrading the firmware, you can add -f flag to force the downgrade.
- Use **mx-connect-mgmt nwk_info [interface name] -a** command to check the current cellular modem firmware version.
- MCM will perform auto-reinstallation if the 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 the 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 shows 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 as a 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.

Refer to the following link for the C API document:

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

To integrate your applications securely with the MC C API, you should follow the following guidelines:

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.

6. System Installation and Update

In this chapter, we will introduce how to install and update Moxa Industrial Linux and the bootloader.

Full System Installation Using .img File

Using a TFTP Server From the Bootloader Menu

Refers to instruction in the Accessing Bootloader Menu section

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

Using an SD Card From the Bootloader Menu

Refers to instruction in the Accessing Bootloader Menu section

Automatic Installation From an SD Card

Besides manually installing the system image from the bootloader menu, you can also trigger the image installation process within the operating system using **`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 SD attached to the system. The new image will be available upon the next system boot-up.

> **NOTE**
>
> The formats supported by the SD card are FAT32 and ext4, respectively.

Command	Description
-d, --disk	Display the name of the external storage (e.g., SD card) where the image file is located. Use the **`mx-interface-mgmt disk`** command to query the external storage name.
-f, --file	Display the name of the image file in the external storage
-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-bootloader-mgmt image-auto-install-tool`**

The following is an example of the automatic installation of the system image from a SD card:

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

```
moxa@moxa-tbzkb1090918:~# sudo mx-interface-mgmt disk
NAME   DEVICE        SYSTEM_DISK  NUMBER_OF_PARTITIONS  AUTOMOUNT_SETTING
SD     /dev/mmcblk1  N            1                     false
eMMC   /dev/mmcblk0  Y            4                     false
```

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

```
moxa@moxa-tbzkb1090923:~$ sudo mx-interface-mgmt partition
NAME      DEVICE          IS_MOUNTED  FS_TYPE  MOUNTPOINT        MAPPER_DEVICE
SD_p1     /dev/mmcblk1p1  N           N/A      N/A               N/A
eMMC_p1   /dev/mmcblk0p1  Y           ext4     /boot_device/p1   N/A
eMMC_p2   /dev/mmcblk0p2  Y           ext4     /boot_device/p2   N/A
eMMC_p3   /dev/mmcblk0p3  Y           ext4     /boot_device/p3   N/A
eMMC_p4   /dev/mmcblk0p4  Y           ext4     /boot_device/p4   N/A

moxa@moxa-tbzkb1090923:~$ sudo mx-interface-mgmt partition SD_p1 mount
```

3. Configure an auto-installation event in partition 1 of the SD device with the image file **IMG_VM-1220_MIL3_V1.0_Build_24072602_ImageBuild_240726_174004.img** :

```
moxa@moxa-tbzkb1090918:~# sudo mx-bootloader-mgmt image_auto_install -d SD -
f IMG_VM-1220_MIL3_V1.0_Build_24072602_ImageBuild_240726_174004.img
```

> **NOTE**
>
> Ensure that the image file and sha256 hash files are available in partition 1 of the SD card before configuring the event.

4. Reboot the system to trigger the auto installation of the system image from the SD card.

```
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.

> **NOTE**
>
> For VM-1220-T, contact your local distributor for the upgrade pack.

To use Moxa Software Updater (MSU), run the command **`# 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 refreshed 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.
`update`	The **`mx-sw-updater update`** 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.

Command and Usage	Description
`upgrade[flags]`	The **`mx-sw-updater upgrade`** 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:** Upgrades to the newest version in the local cache • **--remote:** Upgrades remotely via the APT server (default option). • **--local:** Upgrades 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).
`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 change log. **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>:** Specifies 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*

Command and Usage	Description
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.

Example of Upgrading System from V1.0 to V1.1

Here's an example of using **mx-sw-updater** to upgrade Moxa's UC-4434A-I-T computer from system image V1.0 to V1.1:

1. Download the V1.1 upgrade packs to the target computer
   - moxa-UC-4400A_MIL3_V1.1-upgrade-pack.sha512
   - 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_V1.1-upgrade-pack
moxa-UC-4400A_MIL3_V1.1-upgrade-pack.sha512
moxa-UC-4400A_MIL3_V1.1-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, download the refreshed upgrade pack instead:
   - moxa-UC-4400A_MIL3_V1.1-refresh-upgrade-pack
   - moxa-UC-4400A_MIL3_V1.1-refresh-upgrade-pack.sha512.bin.signed
   - moxa-UC-4400A_MIL3_V1.1-refresh-upgrade-pack.sha512
2. 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_V1.1-upgrade-pack** command.

```
root@moxa-imoxa0920070:/home/moxa# mx-sw-updater configure -p moxa-UC-
4400A_MIL3_V1.1-upgrade-pack
INFO[2024-10-13T04:24:44Z] configure successfully
```

3. Check what packages will be upgraded by applying V1.1 upgrade pack using the **mx-sw-updater list --compare V1.1** command.

```
root@moxa-imoxa0920070:/home/moxa# mx-sw-updater list --compare V1.1
INFO[2024-10-13T11:08:01Z] compare two packages: system and 1.1
 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
```

4. Upgrade system to V1.1 with auto-recovery enabled using the **mx-sw-updater upgrade --local --system-failback** command. For remote upgrades via the Moxa APT repo, use **mx-sw-updater upgrade --remote --system-failback** instead.

```
root@moxa-imoxa0920070:/home/moxa# mx-sw-updater upgrade --local --system-
failback
INFO[2024-10-13T11:32:22Z] current version: V1.0, target version: 1.1

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.
```

5. Reboot the computer after the upgrade is complete.

```
Setting up moxa-connection-manager (1.5.14-1+deb11) ...
Setting up moxa-computer-interface-manager (1.37.0-1+deb11) ...
Setting up moxa-bootloader-manager (2.5.0-1+deb11) ...
*** moxa-bootloader-manager successfully updated. ***
Setting up moxa-mil-base-system-arm64 (3.3.0-1+deb11u2) ...
Installing new version of config file /etc/mil_build ...
Installing new version of config file /etc/mil_release ...
Installing new version of config file /etc/mil_version ...
Setting up moxa-uc-4400a-base-system (3.3.0+deb11u1) ...
Installing new version of config file /etc/moxa-version.conf ...
Processing triggers for libc-bin (2.31-13+deb11u10) ...
Processing triggers for rsyslog (8.2102.0-2+deb11u1) ...
Processing triggers for dbus (1.12.28-0+deb11u1) ...
INFO[2024-10-13T11:36:07Z] upgrade successfully, please reboot the system to
take effect
root@moxa-imoxa0920070:/home/moxa# reboot
```

6. Verify the system has been upgraded to V1.1 by using the **mx-ver command**.

```
root@moxa-imoxa0920070:/home/moxa# mx-ver
UC-4434A-I-T MIL3 version 1.1 Build 24093010
```

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-imoxa1234567:~$ mx-ver
VM-1220-T-BESS MIL3 version 1.0.0 Build 24102503
```

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 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-imoxa1234567:~# mx-bootloader-mgmt upgrade -i
Current bootloader information:
compatible model: VM-1220-T
bootloader version: 0.2.0S00
sha256sum: f862270c4e5bbe2ddb5b87de50bb8ff281a556efd0fc035c0515c4fbc1ae1584
md5sum: 67025d5d6ab0792e952d1e0126850913
```

Updating Bootloader With the Firmware Binary

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

```
root@moxa-tbbbb1182827:# mx-bootloader-mgmt upgrade -f
/media/SD_p1/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
```

Updating with the Failback Function Enabled

We highly recommend you enable failback before performing the bootloader update because a power outage may cause the device to be unable to boot. Refer to failback feature of Moxa System Manager (MSM) tool.

7. Backup, Decommission, and Recovery

In this chapter, we will introduce how to use the 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).

The following diagram illustrates an overview of MIL3 system layout:

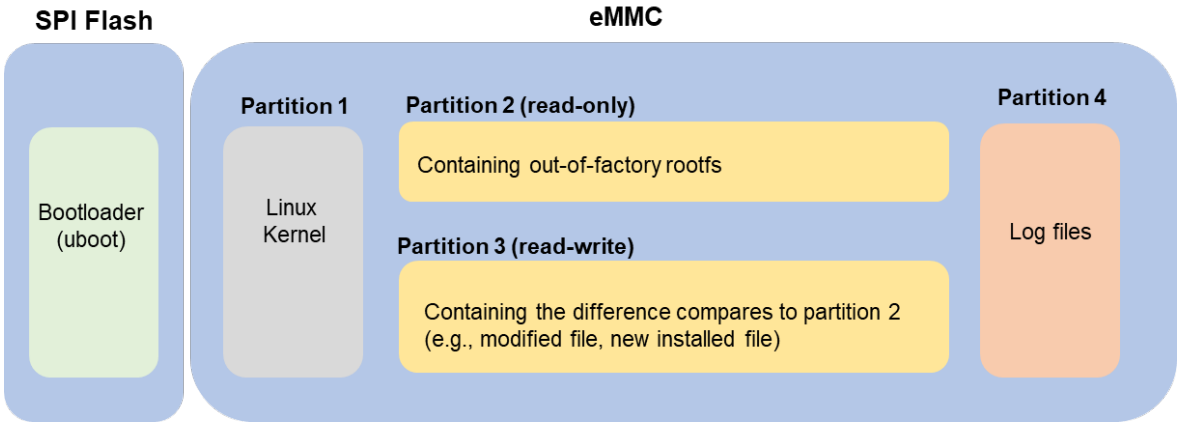


Figure 7.1—Layout Overview of Arm-based Computer with MIL3

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

Sub-commands	Description
create	Creates a snapshot of the 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 systemis 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 used to restore a Moxa Arm-based computer of the same model with MIL 3.0. For example, if you create a backup on VM-1220 with MIL3, you can use the backup to restore another VM-1220 with MIL3.

Use **`# 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 the 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 restoration process begins. A system reboot is required after restoration.
info	Displays the time of creation 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 backup 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 because of 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. 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 while creating, deleting, and restoring.

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 shows how to back up a system to a SD card with the mounting point:

/media/SD_p1:

```
root@moxa-imoxa1234567:~# mx-system-mgmt backup create -D /media/SD_p1
Set /media/SD_p1 as backup directory.
It is recommended to use cold creation mode 'mx-system-mgmt backup create --
cold'
There is no backup information
Estimation of Required Space: 323MB
Available Space: 59616MB
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...
There is no /boot_device/p2/rootfs.sqfs.sha512sum.bin.signed file, the system
environment is insecure.
 323MiB 0:00:03 [ 105MiB/s]
[   <=>                                             ]
Type: backup
Create Time: 2024.09.05-05:07:48
Size: 324MB
The backup has been created successfully under: /media/SD_p1
```

The following example shows how to restore a backup from the SD with the mounting point

/media/SD_p1:

```
root@moxa-imoxa1234567:~# mx-system-mgmt backup restore -D /media/SD_p1
Set /media/SD_p1 as backup directory.
Type: backup
Create Time: 2024.09.05-05:07:48
Size: 324MB
Start verifying backup file, please wait...
Verified OK!
Estimation of Required Space: 324MB
Available Space: 4767MB
Would you like to continue? (y/N)
y
There is no snapshot information
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...
 323MiB 0:00:02 [ 134MiB/s] [=============================================>]
100%
There is no /boot_device/p2/rootfs.sqfs.sha512sum.bin.signed file, the system
environment is insecure.
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.

root@moxa-imoxa1234567:~# 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 **RESET** 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 becomes 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 in the boot-up storage

Back up your files before resetting the system to factory defaults. The data stored in the Arm-based computer's boot-up storage will be destroyed after resetting to factory defaults.

You can also use the **`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 **RESET** button for a different action (e.g., restore to a snapshot), refer to the 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 to help erase security-sensitive information.

ATTENTION

Decommission will erase all the data, including event and audit logs

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 **`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
```

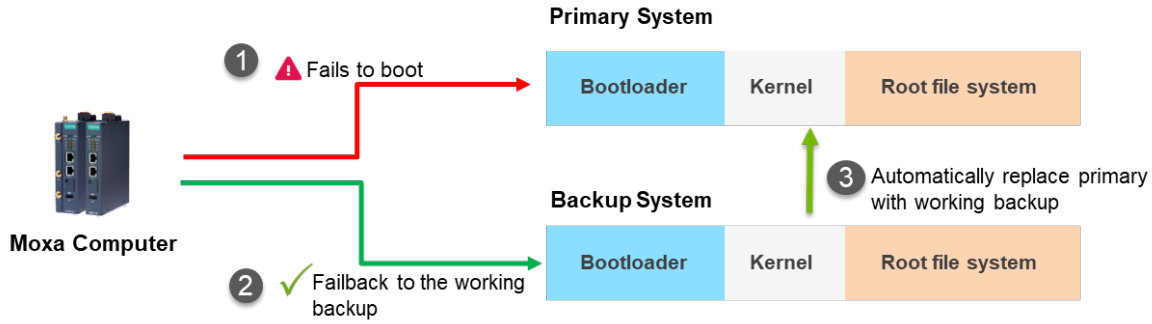
The decommissioning process will do the following:

1. Overwrite the system partition four times with shred so that all user files will be deleted and cannot be recovered.
2. Overwrite the log partition four 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 boot-up 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 recover 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 the user.

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



Use **`# 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 the 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 deletes the existing system replica
info	Displays the time of creation 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.

The following is an example of how to enable system failback using the cold 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 the script below to add criteria you would like Moxa System Manager to check.

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

In the example below 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 securely.

Communication Integrity and Authentication

The following 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

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 the system to compensate for 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 a 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 with no 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)
```

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

Session Termination After Inactivity

This setting automatically terminates the login sessions after a standard period of inactivity. The VM-1220-T by default disables this function. Follow the instructions below 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 the variable **`ClientAliveInterval`** in **`/etc/ssh/sshd_config.d/00-moxa-sshd.conf`** • To apply the rule to sudo user, make sure the variable **`env_keep+="TMOUT"`** exists in **`/etc/sudoers.d/00-moxa-sudoers-conf`**

Login Banner Message

Set a message banner message to displayg a welcome, informational, or warning message to unauthorized users. Follow the instructions below 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 the variable **`Banner /etc/issue.net`** in **`/etc/ssh/sshd_config.d/00-moxa-sshd.conf`**

Bootloader Login Policy

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

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 a 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 and tool are maintained by tpm2-software community: https://tpm2-software.github.io/

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

Default Monitored Files

The security configuration files and directories included in the default database created by Moxa are as follows

- 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
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

To run a comparison between the current system against the Moxa AIDE database, run **aide --check -c /etc/aide/aide-moxa.conf**

```
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 made configuration changes, run **aide --init -c /etc/aide/aide-moxa.conf**

You should see the following output that 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
```

You can run **aide --check -c /etc/aide/aide-moxa.conf** to compare current system against the updated AIDE database.

How to Perform Authenticity an 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 an openSSL signed database containing every single file under the file systems, then validate the authenticity of the database before using AIDE to check the integrity of all files in the file system. Follow the steps below to create such an AIDE database.

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

   ```
   moxa@moxa-tbbbb1182827:/# sudo aide --init -c /etc/aide/aide-fs.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 another location, such as a centralized remote system.
7. Verify whether 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 the current system against the AIDE database using **aide --check -c /etc/aide/aide-fs.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 detailed instructions of Fail2ban usage
https://www.fail2ban.org/wiki/index.php/Main_Page

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. Follow 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] Change the interface list according to the interface you like to monitor. For example, add LAN1, LAN2, and cellular (4G/LTE) on 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 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 by **Start** command (Use CTRL+D to exit if initializing is successful).

```
[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 detailed 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.

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 the 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 5.

```
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 the 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 links 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 the background. The -p is the process ID. The -e switch takes 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: **1024MB**
2. Log partition applies Linux Unified Key Setup (LUKS) encryption and restricts non-root users 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 the 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 achieved logs are kept at a time. When log rotates, the oldest archive will be deleted if 14 achieved logs exist.
   - Audit log rotation rule can be modified in **`/etc/logrotate.d/auditd`**
5. The log timestamp is the local system time which synchronize with a remote Network Time Protocol (NTP) server.
   - For time synchronization status and configuration, refer to timedatectl(1)

NOTE

Click the following link for more information on the usage 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

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 the Bootloader menu
3. The maximum number of logs is 4,000 records, where the oldest log will be overwritten when the maximum capacity is reached.
4. The timestamp of the log read from the local Real-time Clock (RTC) that is synchronized with Network Time Protocol (NTP) server.
5. Log format and log events are described below

Audit Log Structure

Header	Explanation	Possible Values
Time	Timestamp 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 ~ 15
Event Message	Description of the logged event	See the table below for the list of events

Audit Events

Category	Event ID	Event Type	Event Message
System	1	Info	All bootloader configurations 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 found
Install		Warning	System image file error
Install		Warning	File size is too large
Install		Warning	Upgrade system image failure
Install		Alert	System image authenticity check fails
Install	9	Info	Install system image from the SD card ok
Install		Warning	SD/USB/eMMC device not found
Install		Warning	Hash/Signature file not found
Install		Warning	System image file error
Install		Warning	File size is too large
Install		Warning	Upgrade system image failure
Install		Alert	System image authenticity check fails
Secure	10	Info	Install system image from USB ok
Secure		Warning	SD/USB/eMMC device not found
Secure		Warning	Hash/Signature file not found
Secure		Warning	System image file error
Secure		Warning	File size is too large
Secure		Warning	Upgrade system image failure
Secure		Alert	System image authenticity check fails
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 failed
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 failure
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 an 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. Some common approaches are:

1. **Log rotation**

   Log rotation is enabled by default in Moxa Arm-based computer to prevent audit storage capacity full. Refer to **Linux Audit Log** and **Bootloader Audit log** sections for details.

   In Linux, configure the logrotate to limit the disk space usage to prevent running out of space. The log-rotate 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 is over the size 2M with only three rotations.

```
/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. Refer to How to Set Up Centralized Logging on Linux with Rsyslog
4. **Assign appropriate action when audit storage space is full or error occurs**

   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 the example below, a warning email will be sent to an email account specified in **action_mail_acct** parameter when the free space in the file system containing log files drops 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 the audit storage disk got an error or is full. The actions are ignore, syslog, rotate (for disk full only), exec, suspend, single, and halt.

   Refer to auditd(8) for a detailed explanation of each action and parameters.

9. Customization and Programming

MIL1 (Debian 9) to MIL3 (Debian 11) Migration

Moxa Arm-based computers with MIL1 (Debian 9) do not support a direct upgrade to MIL3 (Debian 11). If you have such a request, contact your regional sales representative.

If you are migrating an application previously developed on MIL1 to MIL3, reference the below table for the major changes.

<table>
<tr><th>Category</th><th>Description</th><th>MIL1 (Debian 9)</th><th>MIL3 (Debian 11)</th></tr>
<tr><td rowspan="2">Password rule</td><td>Password change enforced upon first login</td><td>N/A</td><td>✓</td></tr>
<tr><td>Password complexity enforcement</td><td>N/A</td><td>At least 8 characters in length
Password dictionary check</td></tr>
<tr><td rowspan="5">Backup and restore utilities</td><td>Reinstall a system image</td><td>Via bootloader menu</td><td>Via bootloader menu</td></tr>
<tr><td>Create a backup and restore</td><td>N/A</td><td rowspan="4">Moxa System Manager (MSM)
Use mx-system-mgmt</td></tr>
<tr><td>Create a snapshot and restore</td><td>N/A</td></tr>
<tr><td>Automatic system failback recovery</td><td>N/A</td></tr>
<tr><td>Reset to factory default</td><td>Use mx-set-def</td></tr>
<tr><td rowspan="3">Network connection utilities</td><td>Default LAN (Ethernet) port configuration</td><td>LAN1(static IP):192.168.3.127
LAN2(static IP):192.168.4.127
…</td><td>• LAN1: Assigned by DHCP server. Link-local IP addresses will be assigned when the DHCP server is not available.
• LAN2(static IP):192.168.4.127
• LAN3(static IP):192.168.5.127
• LAN4(static IP):192.168.6.127</td></tr>
<tr><td>Cellular connection utility</td><td>Use cell_mgmt</td><td rowspan="2">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</td></tr>
<tr><td>Wi-Fi connection utility</td><td>Use wifi_mgmt</td></tr>
<tr><td>I/O and interface management utilities</td><td>Serial port mode change (RS-232, RS-422, and RS-485 2-wire)</td><td>Use mx-uart-ctl</td><td>Use mx-interface-mgmt
Refers to a serial port in Moxa Computer Interface Manager (MCIM) section</td></tr>
</table>

Category	Description	MIL1 (Debian 9)	MIL3 (Debian 11)
	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 are 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:

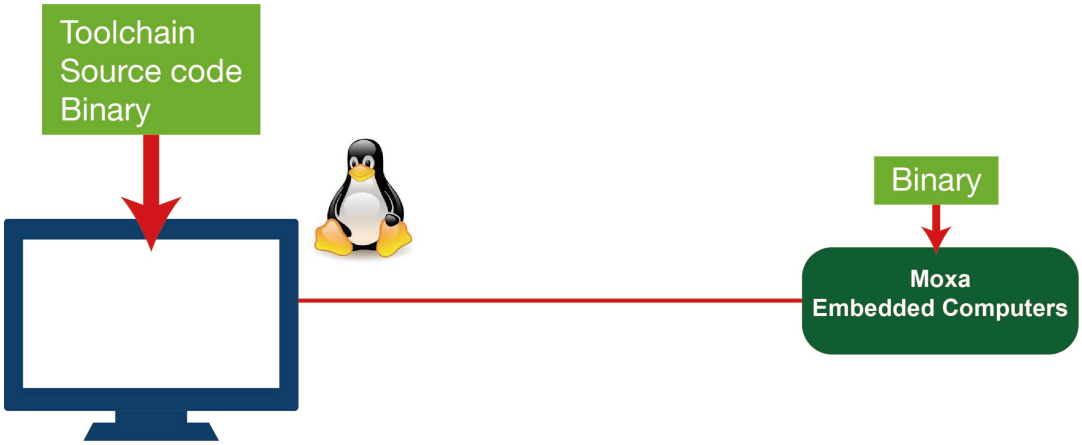
1. Make sure a network connection is available.
2. Use **aptupdate** to update the Debian package list.

```
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 the 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 http://debian.moxa.com/debian stretch main contrib non-free
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 the apt server (all Moxa UC Series computers use the official Debian toolchain).

   For UC computers with **armhf** architecture

   ```
   user@Linux:~$ apt install crossbuild-essential-armhf
   ```

   For UC computers 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 of 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
```

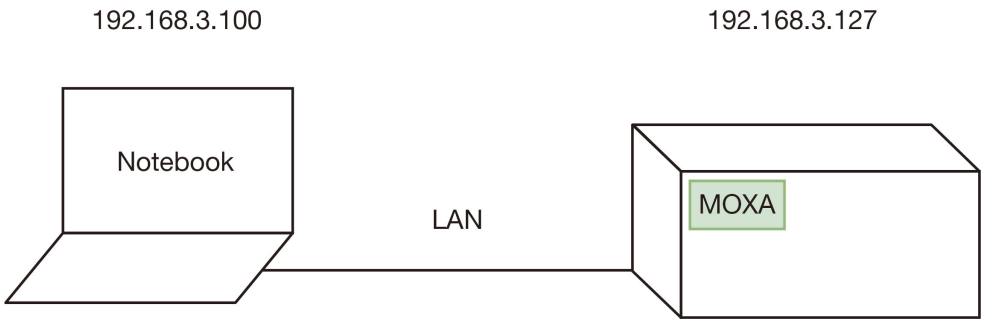
or

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, change the toolchain settings as follows:

```
CC:=arm-linux-gnueabihf-gcc
STRIP:=arm-linux-gnueabihf-strip
```

Using I/O Programming Guide

VM-1220-T also provides I/O programming guides for better developing programs when using digital inputs and outputs, which includes the following sections:

Tutorials:

Shows users how to build code and use C or Python to access I/O data.

I/O Libraries:

Shows users how to access digital inputs and digital outputs.

User Defined LED Indicator:

Shows users how to access LED indicators.

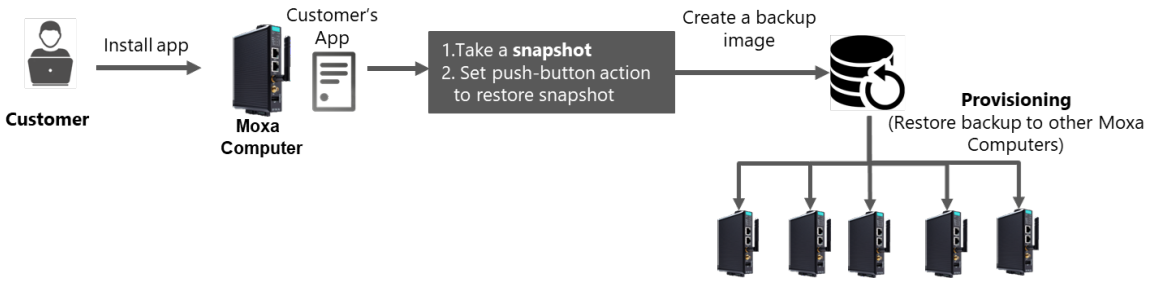
Error Codes:

Provides the meaning of the return code to help users perform troubleshooting tasks.

Creating a Customized Image

Introduction

This section introduces how to build a customized image that set the push-button on Moxa computers to reset to customized environment instead of the Moxa out-of-factory setting. This customized image can also be used for provisioning other Moxa computers.



Using System Snapshots and Backups

1. Configure the Moxa Arm-based computer and install application
2. Create a Snapshot
3. Reference the Customize the Button Action section to configure the action of push-button on the Moxa Arm-based computer to restore Snapshot
   - Copy content of default script to custom.script, change to reset-to-default
   - Change to set-to-factory-default command (mx-system-mgmt default restore –y) of button to restore snapshot (**mx-system-mgmt snapshot restore -y**)

```
#!/bin/sh
ACTION="${1}"
SECONDS="${2}"
if [ "${ACTION}" = "press" ]; then
        /usr/bin/mx-interface-mgmt led SYS set_state heartbeat
elif [ "${ACTION}" = "hold" ]; then
        if [ ${SECONDS} -eq 7 ]; then
                /usr/bin/mx-interface-mgmt led SYS set_state on
        elif [ ${SECONDS} -eq 9 ]; then
                /usr/bin/mx-interface-mgmt led SYS set_state off
        fi
elif [ "${ACTION}" = "release" ]; then
        if [ ${SECONDS} -lt 1 ]; then
                /usr/sbin/reboot
        elif [ ${SECONDS} -ge 7 ] && [ ${SECONDS} -lt 9 ]; then
                /usr/sbin/mx-system-mgmt snapshot restore -y
                /usr/sbin/reboot
        fi
        /usr/bin/mx-interface-mgmt led SYS set_state on
fi
```

4. Create a Backup Image; the backup will include the snapshot taken in step #2
5. The Backup Image can now be used for provisioning other Moxa computers of the same model using the backup restore command.