Moxa x86 Linux Software User Manual

x86 Linux Drivers and SDK

Exported on 2024-11-12 18:29:48

Table of Contents

1 1. Introduction 6

1.1 Available Model Series 6

2 2. Getting Started 7

2.1 Linux OS Installation Instructions 7

2.1.1 Prepare bootable USB drive 7

2.1.2 Current Supported Distributions 7

2.1.3 How to Enter BIOS Menu 7

3 3. x86 Linux SDK Wizard 8

3.1 Basic Information 8

3.2 Moxa x86 Linux SDK Applicable Table 8

3.3 Software Flow Diagram 9

3.3.1 Software Diagram 9

3.3.2 User Interface 9

3.3.3 Notice Before Starting Installation 10

4 4. Peripheral Interface Operations 11

4.1 Utilities 11

4.1.1 Utilities Applicable Table 11

4.1.2 Serial Port 12

4.1.3 Digital IO (DIO) Port 13

4.1.4 LTE (mPCIe slot) Module Control 13

4.1.5 Programmable LED Control 14

4.1.6 Relay Port State Control 15

4.1.7 Power Input Port State 15

4.1.8 USB Port Power State Control 16

4.1.9 Scaler Utility 17

4.1.10 MCU upgrade tool 17

4.1.11 Moxa MCU Manager 18

4.1.12 Moxa Disk Hotswap Daemon 21

4.1.13 HSR/PRP Utility 22

4.1.14 IRIG-B Utility 23

4.1.15 MCIM wrapper 25

4.2 Drivers 28

4.2.1 Drivers Applicable Table 28

4.2.2 moxa-it87-gpio-driver 29

4.2.3 moxa-it87-serial-driver 30

4.2.4 moxa-it87-wdt-driver 30

4.2.5 moxa-mxuport-driver 31

4.2.6 moxa-mxu11x0-driver 31

4.2.7 moxa-sdhci-pci-driver 32

4.2.8 moxa-gpio-pca953x-driver 33

4.2.9 moxa-hid-ft260-driver 33

4.2.10 moxa-irigb-driver 33

4.2.11 intel-gpu-i915-backports 34

4.3 Libraries 35

4.3.1 libgpiod 35

5 5. Basic Linux Concepts 36

5.1 5.1. Mounting SD Card Slot on BXP/DRP Series 36

5.2 5.2. Secure Boot 37

5.3 5.3. Linux PTP (IEEE 1588) 37

5.3.1 Example for Linux PTP setting up 38

6 6. Troubleshooting 41

6.1 6.1. How to Print Kernel Message from Linux Environment 41

6.2 6.2. How to Collect Systems Logs from Linux Environment 41

6.3 6.3. How to Get Installation Logs from Moxa x86 Linux SDK Install Wizard 41

6.4 6.4. How to Get Hardware Information on Host 42

7 7. Appendix 44

7.1 7.1. The License/Commercial-Use of Linux Distributions 44

7.1.1 Debian 44

7.1.2 Ubuntu 44

7.1.3 Red Hat Enterprise Linux (RHEL) 44

7.1.4 CentOS 45

8 8. Revision History 46

* [1. Introduction](#scroll-bookmark-3)
  + [Available Model Series](#scroll-bookmark-4)
* [2. Getting Started](#scroll-bookmark-5)
  + [Linux OS Installation Instructions](#scroll-bookmark-6)
    - [Prepare bootable USB drive](#scroll-bookmark-7)
    - [Current Supported Distributions](#scroll-bookmark-8)
    - [How to Enter BIOS Menu](#scroll-bookmark-9)
* [3. x86 Linux SDK Wizard](#scroll-bookmark-10)
  + [Basic Information](#scroll-bookmark-11)
  + [Moxa x86 Linux SDK Applicable Table](#scroll-bookmark-12)
  + [Software Flow Diagram](#scroll-bookmark-13)
    - [Software Diagram](#scroll-bookmark-14)
    - [User Interface](#scroll-bookmark-15)
    - [Notice Before Starting Installation](#scroll-bookmark-16)
* [4. Peripheral Interface Operations](#scroll-bookmark-17)
  + [Utilities](#scroll-bookmark-18)
    - [Utilities Applicable Table](#scroll-bookmark-19)
    - [Serial Port](#scroll-bookmark-20)
    - [Digital IO (DIO) Port](#scroll-bookmark-21)
    - [LTE (mPCIe slot) Module Control](#scroll-bookmark-22)
    - [Programmable LED Control](#scroll-bookmark-23)
    - [Relay Port State Control](#scroll-bookmark-24)
    - [Power Input Port State](#scroll-bookmark-25)
    - [USB Port Power State Control](#scroll-bookmark-26)
    - [Scaler Utility](#scroll-bookmark-27)
    - [MCU upgrade tool](#scroll-bookmark-28)
    - [Moxa MCU Manager](#scroll-bookmark-29)
    - [Moxa Disk Hotswap Daemon](#scroll-bookmark-30)
    - [HSR/PRP Utility](#scroll-bookmark-31)
    - [IRIG-B Utility](#scroll-bookmark-32)
    - [MCIM wrapper](#scroll-bookmark-33)
  + [Drivers](#scroll-bookmark-34)
    - [Drivers Applicable Table](#scroll-bookmark-35)
    - [moxa-it87-gpio-driver](#scroll-bookmark-36)
    - [moxa-it87-serial-driver](#scroll-bookmark-37)
    - [moxa-it87-wdt-driver](#scroll-bookmark-38)
    - [moxa-mxuport-driver](#scroll-bookmark-39)
    - [moxa-mxu11x0-driver](#scroll-bookmark-40)
    - [moxa-sdhci-pci-driver](#scroll-bookmark-41)
    - [moxa-gpio-pca953x-driver](#scroll-bookmark-42)
    - [moxa-hid-ft260-driver](#scroll-bookmark-43)
    - [moxa-irigb-driver](#scroll-bookmark-44)
    - [intel-gpu-i915-backports](#scroll-bookmark-45)
  + [Libraries](#scroll-bookmark-46)
    - [libgpiod](#scroll-bookmark-47)
* [5. Basic Linux Concepts](#scroll-bookmark-48)
  + [5.1. Mounting SD Card Slot on BXP/DRP Series](#scroll-bookmark-49)
  + [5.2. Secure Boot](#scroll-bookmark-50)
  + [5.3. Linux PTP (IEEE 1588)](#scroll-bookmark-51)
    - [Example for Linux PTP setting up](#scroll-bookmark-52)
      * [[Ordinary Clock (OC) mode]](#scroll-bookmark-53)
      * [[Boundary Clock (BC) mode]](#scroll-bookmark-54)
      * [[Transparent Clock (TC) mode]](#scroll-bookmark-55)
* [6. Troubleshooting](#scroll-bookmark-56)
  + [6.1. How to Print Kernel Message from Linux Environment](#scroll-bookmark-57)
  + [6.2. How to Collect Systems Logs from Linux Environment](#scroll-bookmark-58)
  + [6.3. How to Get Installation Logs from Moxa x86 Linux SDK Install Wizard](#scroll-bookmark-59)
  + [6.4. How to Get Hardware Information on Host](#scroll-bookmark-60)
* [7. Appendix](#scroll-bookmark-61)
  + [7.1. The License/Commercial-Use of Linux Distributions](#scroll-bookmark-62)
    - [Debian](#scroll-bookmark-63)
    - [Ubuntu](#scroll-bookmark-64)
    - [Red Hat Enterprise Linux (RHEL)](#scroll-bookmark-65)
    - [CentOS](#scroll-bookmark-66)
* [8. Revision History](#scroll-bookmark-67)

# 1. Introduction

This Moxa x86 Linux Software User Manual can help x86 Linux user to understand and navigate the usage of Moxa x86 Linux utilities and standard Linux operating system.

Below, we've provided comprehensive information for Getting Started, x86 Linux SDK wizard, Peripheral Interface Operations, Basic Linux Concepts, Troubleshooting and Appendix for x86 Linux user.

## Available Model Series

* **BXP** series
  + [BXP-A100](https://www.moxa.com/products/industrial-computing/x86-computers/bxp-a100-series), [BXP-C100](https://www.moxa.com/en/products/industrial-computing/x86-computers/bxp-c100-series), BXP-A101
* **DRP** series
  + [DRP-A100,](https://www.moxa.com/products/industrial-computing/x86-computers/drp-a100-series) [DRP-C100](https://www.moxa.com/en/products/industrial-computing/x86-computers/drp-c100-series)
* **RKP** series
  + [RKP-A110](https://www.moxa.com/en/products/industrial-computing/x86-computers/rkp-a110-series), [RKP-C110](https://www.moxa.com/tw/products/industrial-computing/x86-computers/rkp-c110-series), RKP-C220
* **V** series
  + [V3200](https://www.moxa.com/tw/products/industrial-computing/x86-computers/v3200-series)
  + V3400
* **MPC** series
  + MPC-3000
* **DA** series
  + DA-820E
  + [DA-820C](https://www.moxa.com/tw/products/industrial-computing/x86-computers/da-820c-series)
  + [DA-682C](https://www.moxa.com/tw/products/industrial-computing/x86-computers/da-682c-series)
  + [DA-681C](https://www.moxa.com/tw/products/industrial-computing/x86-computers/da-681c-series)
  + [DA-680](https://www.moxa.com/tw/products/industrial-computing/x86-computers/da-680-series)

# 2. Getting Started

The Getting Started section will introduce the Linux OS distribution installation instructions.

## Linux OS Installation Instructions

### Prepare bootable USB drive

At first, prepare a **USB storage drive**, download the [**Rufus**](https://rufus.ie/en/) to create bootable USB drive. Download the ISO image file and restore ISO image into USB storage drive.

### Current Supported Distributions

* **Debian**
  + **Debian 11 (bullseye), Linux kernel 5.10**
  + **Debian 12 (bookworm), Linux kernel 6.1**
  + [Official Debian installation guide](https://www.debian.org/releases/bullseye/amd64/index.en.html)
* **Ubuntu**
  + **Ubuntu 20.04 LTS (Focal Fossa), Linux kernel 5.4 (20.04.1), Linux kernel 5.15 (20.04.5), HWE kernel 5.15 or later version**
  + **Ubuntu 22.04 LTS (Jammy Jellyfish), Linux kernel 5.15 (22.04.3), Linux kernel 6.5 (22.04.4), HWE kernel 6.5 or later version**
  + [Official Ubuntu installation guide](https://releases.ubuntu.com/22.04)
* **RedHat**
  + **RedHat 9, Linux kernel 5.14**
    - [Official RedHat 9 download link](https://access.redhat.com/downloads)
    - [Official RedHat 9 installation guide](https://www.redhat.com/sysadmin/install-linux-rhel-9)
* **CentOS 7**
  + **CentOS 7.9**, **Linux kernel 3.10**
    - [CentOS-7-x86\_64-DVD-2009.iso download link](http://isoredirect.centos.org/centos/7/isos/x86_64/)

### How to Enter BIOS Menu

Boot up device and press **F2** key from keyboard to enter BIOS menu, and select **boot from USB** from **UEFI mode.**

Then follow the distribution's official installation guide to finish OS installation procedure.

# 3. x86 Linux SDK Wizard

## Basic Information

The **Moxa x86 Linux SDK** enables the easy deployment on the Moxa x86 IPC platform. The SDK contains components for peripheral drivers, peripheral control tools and configuration files.

It also provides deployment features, such as build & installation log, dry-run, and self test on target model. User can download the Moxa x86 Linux SDK zip file from official product's website.

Below is the list of files:

* **\*.tgz** - the tarball file of x86 Linux SDK Install Wizard
* **README.docx/README.md** - the user manual of x86 Linux SDK Install Wizard
* **sources\_list** - the list of source code
* **build\_info** - build information

|  |
| --- |
| Please extract the **tgz** tarball file under Linux OS environment to avoid file permission issue. |

## Moxa x86 Linux SDK Applicable Table

|  |  |  |
| --- | --- | --- |
| **Model Name** | **Available SDK Version** | **Support Linux Distributions** |
| BXP-A100/BXP-C100 | V1.2 | Debian 11/12, Ubuntu 22.04/20.04 LTS & HWE, RedHat 9, CentOS 7.9 |
| DRP-A100/DRP-C100 | V1.2 | Debian 11/12, Ubuntu 22.04/20.04 LTS & HWE, RedHat 9, CentOS 7.9 |
| RKP-A110/RKP-C110 | V1.2 | Debian 11/12, Ubuntu 22.04/20.04 LTS & HWE, RedHat 9, CentOS 7.9 |
| V3200 | V1.2 | Debian 12, Ubuntu 22.04 LTS (HWE) |
| V3400 | V1.2 | Debian 12, Ubuntu 22.04 LTS (HWE) |
| MPC-3000 | V1.2 | Debian 11, Ubuntu 22.04 LTS, RedHat 9 |
| DA-820E | V1.2 | Debian 12, Ubuntu 22.04 LTS, RedHat 9 |
| DA-820C | V1.2 | Debian 12, Ubuntu 22.04 LTS & HWE, RedHat 9, CentOS 7.9 |
| DA-682C | V1.2 | Debian 12, Ubuntu 22.04 LTS & HWE, RedHat 9, CentOS 7.9 |
| DA-681C | V1.2 | Debian 12, Ubuntu 22.04 LTS & HWE, RedHat 9, CentOS 7.9 |
| DA-680 | V1.2 | Debian 12, Ubuntu 22.04 LTS & HWE, RedHat 9, CentOS 7.9 |
| RKP-C220 | V1.2 | Debian 12, Ubuntu 22.04 LTS (HWE), RedHat 9 |
| BXP-A101 | V1.2 | Debian 12, Ubuntu 22.04 LTS (HWE), RedHat 9 |

## Software Flow Diagram

### Software Diagram

****

### User Interface

****

### Notice Before Starting Installation

* Please configure your **network settings** before installation
* To extract the tgz tarball file under Linux environment (e.g. tar xvf \*.tar.gz)
* Run --dry-run option before installation, to check the target host device and environment are available
* Run --selftest option after installation, to check the status of drivers and tools

# 4. Peripheral Interface Operations

This guide is introduced the usage of **Moxa peripheral interface control utility**. These utilities should be installed after thex86 Linux SDK Wizard installation procedure.

User can check the status of utilities via running ./install.sh --selftest command.

## Utilities

### Utilities Applicable Table

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Available Models** | **Serial Port Utility** | **DIO** **Port** Utility | **PLED Utility** | **Relay Utility** | **Power Input State** | **USB Power Utility** |
| **BXP-A100/BXP-C100** | O | O | X | X | X | X |
| **DRP-A100/DRP-C100** | O | X | X | X | X | X |
| **RKP-A110/RKP-C110** | O | O | X | X | X | X |
| **V3200** | O | O | O | X | X | X |
| **V3400** | O | O | O | X | X | X |
| **MPC-3000** | O | O | X | X | X | X |
| **DA-820E** | O | O | O | O | O | O |
| **DA-820C** | O | O | O | O | O | O |
| **DA-682C** | O | O | O | O | O | O |
| **DA-681C** | O | O | O | O | O | O |
| **DA-680** | O | O | O | O | O | X |
| **RKP-C220** | O | O | X | X | X | X |
| **BXP-A101** | O | O | X | X | X | X |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Available Models** | **LTE (mPCIe slot) Module Utility** | **Scaler  Utility** | MCU Manager and upgrade tool | HSR/PRP Utility | IRIG-B Utility | MCIM wrapper | Disk Hotswap Daemon |
| **BXP-A100/BXP-C100** | X | X | X | X | X | O | X |
| **DRP-A100/DRP-C100** | X | X | X | X | X | O | X |
| **RKP-A110/RKP-C110** | X | X | X | X | X | O | X |
| **V3200** | O | X | O | X | X | O | X |
| **V3400** | O | X | O | X | X | O | O |
| **MPC-3000** | X | O | O | X | X | O | X |
| **DA-820E** | X | X | X | O | X | O | X |
| **DA-820C** | X | X | X | O | O | O | X |
| **DA-682C** | X | X | X | O | X | O | X |
| **DA-681C** | X | X | X | X | X | O | X |
| **DA-680** | X | X | X | X | O | O | X |
| **RKP-C220** | X | X | X | X | X | O | X |
| **BXP-A101** | O | X | X | X | X | O | X |

### Serial Port

The Moxa serial port mode control utility **mx-uart-ctl**, it is for getting and setting serial port's UART mode.

* Drivers dependency
  + moxa-it87-gpio-driver
  + moxa-it87-serial-driver
  + moxa-mxuport-driver
* Libraries dependency
  + libgpiod

**Usage of UART mode control**

|  |
| --- |
| Usage:  mx-uart-ctl -p <port\_number> [-m <uart\_mode>]    OPTIONS:  -p <port\_number>  Set target port.  -m <uart\_mode>  Set target port to uart\_mode  0 --> set to RS-232 mode  1 --> set to RS-485-2W mode  2 --> set to RS-422 mode  3 --> set to RS-485-4W mode    Example:  Get mode from port 0  # mx-uart-ctl -p 0    Set port 1 to RS232 mode  # mx-uart-ctl -p 1 -m 0 |

### Digital IO (DIO) Port

Moxa DIO port control tool **mx-dio-ctl** is for getting DI/DO and setting DO ports status (low/high).

* Drivers dependency
  + moxa-it87-gpio-driver
* Libraries dependency
  + libgpiod

**Usage of DIO state control**

|  |
| --- |
| Usage:  mx-dio-ctl <-i|-o <#port number> [-s <#state>]>    OPTIONS:  -i <#DIN port number>  -o <#DOUT port number>  -s <#state>  Set state **for** target DOUT port  0 --> LOW  1 --> HIGH    Example:  Get value from DIN port 0  # mx-dio-ctl -i 0  Get value from DOUT port 0  # mx-dio-ctl -o 0    Set DOUT port 0 value to LOW  # mx-dio-ctl -o 0 -s 0  Set DOUT port 0 value to HIGH  # mx-dio-ctl -o 0 -s 1 |

### LTE (mPCIe slot) Module Control

Moxa LTE (mPCIe slot) module control tool **mx-module-ctl** is provided to control LTE power on/off state and SIM card select functions

* Drivers dependency
  + moxa-it87-gpio-driver
  + moxa-gpio-pca953x-driver
* Libraries dependency
  + libgpiod

**Usage of LTE (mPCIe slot) module control tool**

|  |
| --- |
| Usage:  mx-module-ctl [Options]    Operations:  -s, --slot <module\_slot\_id>  Select module slot  -p, --power [on|off]  Get/Set power on/off module  -r, --reset [on|off]  Get/Set reset pin to high(on)/low(off) to slot  -i, --sim 1|2  Get/Set sim card slot    Example:  Power on module 1  # mx-module-ctl -s 1 -p on    Set module 2 reset pin to high  # mx-module-ctl -s 2 -r on    Select SIM 2 **for** module 1  # mx-module-ctl -s 1 -i 2    Get power status of module 1  # mx-module-ctl -s 1 -p    Get current SIM slot of module 1  # mx-module-ctl -s 1 -i |

### Programmable LED Control

Moxa LED control tool **mx-led-ctl** is provided to control programmable LEDs light on/off

* Drivers dependency
  + moxa-it87-gpio-driver
  + moxa-gpio-pca953x-driver
* Libraries dependency
  + libgpiod

**Usage of programmable LED control tool**

|  |
| --- |
| Usage:  mx-led-ctl -i <led\_index> [on|off]    OPTIONS:  -i <led\_index>  Set LED index.    Example:  Get state from index 1  # mx-led-ctl -i 1    Set index 1 to on  # mx-led-ctl -i 1 on |

### Relay Port State Control

Moxa relay port state control tool **mx-relay-ctl** is for getting and setting relay ports status (NO: Normal Open/NC: Normal Closed).

* Drivers dependency
  + moxa-it87-gpio-driver
* Libraries dependency
  + libgpiod

**Usage of relay state control**

|  |
| --- |
| Usage:  mx-relay-ctl -p <port\_number> [-m <relay\_mode>]    OPTIONS:  -p <port\_number>  Set target port.  -m <relay\_mode>  Set target port to relay\_mode  0 --> set to NC (Normal Closed) mode  1 --> set to NO (Normal Open) mode    Example:  Get mode from port 0  # mx-relay-ctl -p 0    Set port 0 to mode NC  # mx-relay-ctl -p 0 -m 0    Set port 0 to mode NO  # mx-relay-ctl -p 0 -m 1 |

### Power Input Port State

Moxa power input port state tool **mx-input-power-state** is for getting power input ports status (connected/disconnected):



* Drivers dependency
  + moxa-it87-gpio-driver
* Libraries dependency
  + libgpiod

**Usage of power input port state tool**

|  |
| --- |
| USAGE:  mx-input-power-state -i <power\_port>    OPTIONS:  -i <power\_port>  Get power input port state <connected/disconnected>    EXAMPLE:  Get power input port 0 state  mx-input-power-state -i 0 |

### USB Port Power State Control

Moxa USB port power state control tool **mx-usb-power-ctl** is for setting/getting USB ports (front/rear/internal) power state (off/on) control:

* Drivers dependency
  + moxa-it87-gpio-driver
* Libraries dependency
  + libgpiod

**Usage of USB power port state control tool**

|  |
| --- |
| USAGE:  mx-usb-power-ctl -i <usb\_port> [-s <state>]    OPTIONS:  -i <usb\_port>  Get USB port power state  0: front  1: rear  2: internal    -s <state>  Set USB port power state  0: off  1: on    EXAMPLE:  Get USB front port power state  mx-usb-power-ctl -i 0    Get USB rear port power state  mx-usb-power-ctl -i 1    Set USB front port power state to off  mx-usb-power-ctl -i 0 -s 0    Set USB internal port power state to on  mx-usb-power-ctl -i 2 -s 1 |

### Scaler Utility

Moxa scaler utility is designed to configure basic settings of display devices, such as brightness, touch panel status, and OSD settings

**Usage**

|  |
| --- |
| Usage:  mx-scaler-util [Options]...  Options:  -v, --version  Get scaler firmware version  -s, --status  Get system status  -m, --model  Get model name  -b, --brightness [0~10]  Set brightness, query without arg  -p, --touch-panel [0~1]  Set touch panel on(1)/off(0), query without arg  -o, --osd-config [0~1]  Set OSD on(1)/off(0), query without arg |

### MCU upgrade tool

The mx-lpc-mcu-upgrade-tool is a command-line utility designed for upgrading the firmware of MCU.

|  |
| --- |
| **Warning** |
| Before using MCU firmware upgrade tool, please stop Moxa MCU related serivce to avoid communication conflict issue. |

**Usage**

|  |
| --- |
| Usage:  mx-lpc-mcu-upgrade-tool [Options]...  Options:  -f, --file  Start MCU upgrade from file  -v, --version  Get current MCU version  Example:  mx-lpc-mcu-upgrade-tool -f FB\_MCU\_V3000\_V1.00S03\_22060219.bin  mx-lpc-mcu-upgrade-tool -v |

### Moxa MCU Manager

Moxa MCU Manager (MMM) is used to control MCU(microcontroller) on Moxa x86 computer products. Including the LAN bypass, panel display, panel programmable LEDs, and update MCU ROM firmware.

Please ensure that mx-mcud is running on background as daemon.

**Usage**

|  |
| --- |
| MOXA MCU Management Command-line Utility    Usage:  mx-mcu-mgmt [command]    Flags:  -h, --help Prints help information  -v, --version Prints utility version    Commands:  mcu\_version Get MCU firmware version  relay Control relay mode  wdt\_reset Control watchdog reset mode  wdt\_relay Control watchdog relay mode  poweroff\_relay Control power off (S5) relay mode  app\_wdt\_reset Control app watchdog reset mode  app\_wdt\_relay Control app watchdog relay mode  app\_wdt\_timout Control app watchdog timeout |

**LAN Bypass Modes**

****

The LAN Bypass feature support the following three modes:

* **[Connect]**
  + LAN A and LAN B ports are connected to the NICs and is data transmitted through  
    system normally.
* **[Disconnect]**
  + LAN A and LAN B ports are neither connected to the NICs nor to each other,  
    which means that data packets are blocked.
* **[Bypass]**
  + LAN A and LAN B ports are connected to each other to keep transmitting data  
    without interruption even when a system device crashes or encounters a cyber attack.

**Options of LAN Bypass Modes Control Utility**

* **Relay Mode**
  + Description
    - Directly to get or set LAN Bypass mode from relay states.
  + Example
    - Get LAN Bypass Mode: **mx-mcu-mgmt relay get\_mode**
    - Set LAN Bypass Mode:
      * Set **Connect** Mode: **mx-mcu-mgmt relay set\_mode connect**
      * Set **Disconnect**Mode: **mx-mcu-mgmt relay set\_mode disconnect**
      * Set **Bypass** Mode: **mx-mcu-mgmt relay set\_mode bypass**
* **Watchdog Reset Mode**
  + Description
    - Use the Watchdog Reset Mode to indicate if the **system needs to be reset** after the MCU RTC watchdog (ds1374) is triggered
  + Example
    - Get Watchdog Reset Mode: **mx-mcu-mgmt wdt\_reset get\_mode**
    - Set Watchdog Reset Mode
      * Watchdog Reset OFF:  **mx-mcu-mgmt wdt\_reset set\_mode off**
      * Watchdog Reset ON:  **mx-mcu-mgmt wdt\_reset set\_mode on**
* **Watchdog Relay Mode**
  + Description
    - Use this Watchdog Relay Mode to indicate the **relay mode** to switch to after the MCU RTC watchdog (ds1374) is triggered.
  + Example
    - Get Watchdog Relay Mode: **mx-mcu-mgmt wdt\_relay get\_mode**
    - Set Watchdog Relay Mode
      * Set Watchdog Relay **Connect** Mode: **mx-mcu-mgmt wdt\_relay set\_mode connect**
      * Set Watchdog Relay **Disconnect**Mode: **mx-mcu-mgmt wdt\_relay set\_mode disconnect**
      * Set Watchdog Relay **Bypass** Mode: **mx-mcu-mgmt wdt\_relay set\_mode bypass**
* **Power Off Relay Mode**
  + Description
    - Use this Power Off Relay Mode to indicate the **relay mode** to switch to after the system is **powered off (S5 state)**.
  + Example
    - Get Power Off Relay Mode: **mx-mcu-mgmt poweroff\_relay get\_mode**
    - Set Power Off Relay Mode
      * Set Power Off Relay as **Disconnect**Mode: **mx-mcu-mgmt poweroff\_relay set\_mode disconnect**
      * Set Power Off Relay as **Bypass** Mode: **mx-mcu-mgmt poweroff\_relay set\_mode bypass**

**App Watchdog Modes Control Utility**

The App Watchdog Modes Control Utility is to configure MCU’s behavior.

It provides set MCU timeout value, timeout-reset function, timeout-relay mode, and kicking service and daemon.

* **APP WDT Mode**
  + Description
    - Use this mode to enable or disable the MCU watchdog application
    - Activating the watchdog function is key to creating a trigger to activate LAN bypass when your application encounters issues or is unresponsive.
  + Example
    - Get APP WDT Mode and Timeout value: **mx-mcu-mgmt app\_wdt\_timout get\_timeout**
    - Set APP WDT Mode and Timeout value:
      * Enable APP WDT Mode and set 10 sec timeout: **mx-mcu-mgmt app\_wdt\_timout set\_timeout 10**
      * Disable APP WDT Mode: **mx-mcu-mgmt app\_wdt\_timout set\_timeout 0**
* **APP WDT Reset Mode**
  + Description
    - Use this mode to indicate if the **system needs to be reset** after the MCU app watchdog is timeout triggered.
  + Example
    - Get App Watchdog Reset Mode: **mx-mcu-mgmt app\_wdt\_reset get\_mode**
    - Set App Watchdog Reset Mode when app watchdog is triggered
      * Disable the reset system function: **mx-mcu-mgmt app\_wdt\_reset set\_mode off**
      * Enable the reset system function: **mx-mcu-mgmt app\_wdt\_reset set\_mode on**
* **APP WDT Relay Mode**
  + Description
    - Use this mode to indicate the relay mode to switch to after the MCU app watchdog is timeout triggered.
  + Example
    - Get App Watchdog Relay Mode: **mx-mcu-mgmt app\_wdt\_relay get\_mode**
    - Set App Watchdog Relay Mode when app watchdog is timeout triggered
      * Set the App Relay mode
        + Set **Connect** Mode: **mx-mcu-mgmt app\_wdt\_relay set\_mode connect**
        + Set **Disconnect**Mode: **mx-mcu-mgmt app\_wdt\_relay set\_mode disconnect**
        + Set **Bypass** Mode: **mx-mcu-mgmt app\_wdt\_relay set\_mode bypass**

### Moxa Disk Hotswap Daemon

Moxa Disk Hotswap Daemon is used to monitor the disk plug/unplug status with push buttons and programmable LEDs. User can remove disk via pressed push button to umount and remove disk safely.

This feature support on V3400 series.

* Libraries dependency
  + libgpiod

**Add systemd service to use (if needed)**

edit /lib/systemd/system/mx\_disk\_hotswapd.service

|  |
| --- |
| [Unit]  Description=Moxa disk hotswap daemon service    [Service]  Type=oneshot  ExecStart=/usr/sbin/mx-disk-hotswapd  RemainAfterExit=yes    [Install]  WantedBy=multi-user.target |

then enable service.

|  |
| --- |
| systemctl enable mx\_disk\_hotswapd.service |

If user pressed push button over 3 seconds, the programmable LED will blinks 3 times and turn off light, and daemon will start to umount and remove target disk.

Alternatively, if pressed push button less than 3 seconds, the daemon will scan disk and mount target disk, the programmable LED will turn on to notify user the disk has been mounted, for example:

|  |
| --- |
| /dev/sda2 on /media/disk1p2 type squashfs (ro,relatime,errors=**continue**)  /dev/sda1 on /media/disk1p1 type vfat (rw,relatime,fmask=0022,dmask=0022,codepage=437,iocharset=ascii,shortname=mixed,utf8,errors=remount-ro)  /dev/sda3 on /media/disk1p3 type ext4 (rw,relatime) |

### HSR/PRP Utility

Moxa HSR/PRP card utility is based on SMBUS to query FPGA related register.

**Usage**

|  |
| --- |
| [root@localhost moxa]# mxhsrprpd -h  Usage:  -h: Show **this** information.  -B: Run daemon in the background  -b: SMBUS device, **default** is /dev/i2c-0  -t: HSR/PRP Status update period. Default is 3 second.  -m: configure to prp or hsr mode, **default** is prp mode.  The argurement is [index]:[mode]  [index] range from 0~7.  [mode] 0 is prp, mode 1 is hsr.  Ex: Set card 0 to hsr mode, card 1 to prp mode.  root@Moxa:~# mxhsrprpd -t 2 -m 0:1,1:0  -s: configure fiber speed, **default** is auto detect mode.  The argurement is [index]:[speed]  [index] range from 0~7.  [speed] 0 is 100M, 1 is 1000M. (**default** fiber speed is 1000M)  Ex: Set card 0 fiber speed to 100M, card 1 fiber speed to 1000M.  root@Moxa:~# mxhsrprpd -t 2 -s 0:0,1:1 |

**Add systemd service to use (if needed)**

edit /lib/systemd/system/mx\_hsrprp.service

|  |
| --- |
| [Unit]  Description=Moxa HSR-PRP daemon service    [Service]  Type=oneshot  ExecStart=/usr/sbin/mx\_hsrprp start  ExecStop=/usr/sbin/mx\_hsrprp stop  RemainAfterExit=yes    [Install]  WantedBy=multi-user.target |

then enable service.

|  |
| --- |
| systemctl enable mx\_hsrprp.service |

### IRIG-B Utility

Utility for controlling DA-IRIG-B expansion module Compile and install the IRIG-B time sync daemon.

**Usage**

|  |
| --- |
| [root@localhost moxa]# ServiceSyncTime -h  Found the IRIG-B module, Hardware ID = 7  IRIG-B time sync daemon.  Usage: ServiceSyncTime -t [signal type] -I -i [Time sync interval] -s [Time Source] -p [Parity check mode] -B  -t - [signal type]  0 - TTL  1 - DIFF  **default** value is 1  -I - Inverse the input signal  -s - [Time Source] The sync source from FREERUN(Internal RTC), Fiber or IRIG-B port  0 - FREERUN(Internal RTC) module  1 - Fiber port  2 - IRIG-B port  **default** value is 2  -i - [Time sync interval] The time interval in seconds to sync the IRIG-B time into system time.  1 ~ 86400 Time sync interval. Default is 10 second.  -p - [Parity check mode] Set the parity bit  0: EVEN  1: ODD  2: NONE  **default** value is 0  -B - Run daemon in the background  Usage example: Enable to sync time from IRIG-B Port 1, in TTL signal type every 10 seconds. The input signals is not inverse.  root@Moxa:~# ServiceSyncTime -t 0 -i 10 |

**Use systemd service step by step**

1. Disable NTP service

|  |
| --- |
| **Warning** |
| NTP service affects IRIG-B service time syncing. |

* Disable service

|  |
| --- |
| timedatectl set-ntp **false** |

* Make sure NTP service is inactive

|  |
| --- |
| timedatectl status  Local time: Mon 2023-02-13 02:27:54 PST  Universal time: Mon 2023-02-13 10:27:54 UTC  RTC time: Mon 2023-02-13 10:27:54  Time zone: America/Los\_Angeles (PST, -0800)  System clock **synchronized**: yes  NTP service: inactive  RTC in local TZ: no |

2. Config IRIG-B time sync service

* Edit /usr/sbin/mx\_irigb.sh to config service options MX\_IRIGB\_SERVICESYNCTIME\_OPTS.

|  |
| --- |
| **Info** |
| For more detial about options, please refer to ServiceSyncTime -h |

|  |
| --- |
| ...  # The time sync daemon **default** configure wtih  # -t 1 - Sync time in DIFF signal format  # -i 10 - The time interval in 10 seconds to sync the IRIG-B time into system time.  # -B - Run daemon in the background  #  MX\_IRIGB\_SERVICESYNCTIME\_OPTS="-t 1 -i 10 -B"  ... |

3. Start IRIG-B time sync service

* Create and edit systemd service file /lib/systemd/system/mx\_irigb.service

|  |
| --- |
| [Unit]  Description=Moxa DA-IRIG-B daemon service    [Service]  Type=oneshot  ExecStart=/usr/sbin/mx\_irigb.sh start  ExecStop=/usr/sbin/mx\_irigb.sh stop  RemainAfterExit=yes    [Install]  WantedBy=multi-user.target |

* Launch service

|  |
| --- |
| $ systemctl daemon-reload  $ systemctl enable mx\_irigb.service  Created symlink /etc/systemd/system/multi-user.target.wants/mx\_irigb.service → /lib/systemd/system/mx\_irigb.service.  $ systemctl start mx\_irigb.service  $ systemctl status mx\_irigb.service  ● mx\_irigb.service - Moxa DA-IRIG-B daemon service  Loaded: loaded (/lib/systemd/system/mx\_irigb.service; enabled; vendor preset: enabled)  Active: active (exited) since Tue 2023-02-14 01:48:29 PST; 5s ago  Process: 8322 ExecStart=/usr/sbin/mx\_irigb.sh start (code=exited, status=0/SUCCESS)  Main PID: 8322 (code=exited, status=0/SUCCESS)  CPU: 9ms    Feb 14 01:48:29 moxa systemd[1]: Starting Moxa DA-IRIG-B daemon service...  Feb 14 01:48:29 moxa systemd[1]: Finished Moxa DA-IRIG-B daemon service. |

### MCIM wrapper

MCIM wrapper means Moxa Computer Interface Manager (MCIM) shell script based wrapper. It's provide users with commands similar to MCIM when operating peripherals.

**Usage**

|  |
| --- |
| The Moxa Computer Interface Manager (MCIM) is a tool designed to simplify  user control of peripherals. The design of MCIM aims to enhance  operational efficiency, enabling users to conveniently handle tasks  related to peripheral devices.  Usage:  mx-**interface**-mgmt [command]    Available Commands:  cellular Manages the cellular modem  dio Manages digital inputs and outputs **for** external devices  led Manages LED indicators  relay Manages the relay mode  serialport Manages the serial port  input\_power Manages the power input state  usb\_power Manages the usb power state    Flags:  -h, --help help **for** mx-**interface**-mgmt    Use "mx-interface-mgmt [command] --help" **for** more information about a command. |

**Usage (cellular wrapper)**

|  |
| --- |
| Usage:  mx-**interface**-mgmt cellular <NAME> <COMMAND> [ARG]    Available Commands:  Get the power state of a cellular  $ mx-**interface**-mgmt cellular <cellular\_name> get\_power  Set the power state of a cellular  $ mx-**interface**-mgmt cellular <cellular\_name> set\_power <power\_state>  Get the SIM slot of a cellular  $ mx-**interface**-mgmt cellular <cellular\_name> get\_sim\_slot  Set the SIM slot of a cellular  $ mx-**interface**-mgmt cellular <cellular\_name> set\_sim\_slot <sim\_slot>    Arguments:  cellular\_name: The slot number of cellular (e.g. 1|2)  power\_state: on|off  sim\_slot: 1|2 |

**Usage (dio wrapper)**

|  |
| --- |
| Usage:  mx-**interface**-mgmt dio <NAME> <COMMAND> [ARG]    Available Commands:  Get the state of a dio  $ mx-**interface**-mgmt dio <dio\_name> get\_state  Set the state of a dio  $ mx-**interface**-mgmt dio <dio\_name> set\_state <dio\_state>    Arguments:  dio\_name: The name of dio (e.g. DI0、DO0)  dio\_state: 0(low)|1(high) |

**Usage (led wrapper)**

|  |
| --- |
| Usage:  mx-**interface**-mgmt led <NAME> <COMMAND> [ARG]    Available Commands:  Get the state of a LED  $ mx-**interface**-mgmt led <led\_name> get\_state  Set the state of a LED  $ mx-**interface**-mgmt led <led\_name> set\_state <led\_state>    Arguments:  led\_name: The number of LED (e.g. 0, 1, 2, ......)  led\_state: on|off |

**Usage (relay wrapper)**

|  |
| --- |
| Usage:  mx-**interface**-mgmt relay <NAME> <COMMAND> [ARG]    Available Commands:  Get the mode of a relay  $ mx-**interface**-mgmt relay <relay\_name> get\_mode  Set the mode of a relay  $ mx-**interface**-mgmt relay <relay\_name> set\_mode <relay\_mode>    Arguments:  relay\_name: The number of relay (e.g. 0, 1, 2, ......)  relay\_mode: 0|1  0 --> set to NC (Normal Closed) mode  1 --> set to NO (Normal Open) mode |

**Usage (input\_power wrapper)**

|  |
| --- |
| Usage:  mx-**interface**-mgmt input\_power <NAME> <COMMAND> [ARG]    Available Commands:  Get the state of a input\_power  $ mx-**interface**-mgmt input\_power <input\_power\_name> get\_state    Arguments:  input\_power\_name: The number of input\_power (e.g. 0, 1, 2, ......) |

**Usage (usb\_power wrapper)**

|  |
| --- |
| Usage:  mx-**interface**-mgmt usb\_power <NAME> <COMMAND> [ARG]    Available Commands:  Get the usb power state of a port  $ mx-**interface**-mgmt usb\_power <usb\_port> get\_state  Set the usb power state of a port  $ mx-**interface**-mgmt usb\_power <usb\_port> set\_state <state>    Arguments:  usb\_port: Get USB port power state  0: front  1: rear  2: internal  state: Set USB port power state  0: off  1: on |

**Usage (serialport wrapper)**

|  |
| --- |
| Usage:  mx-**interface**-mgmt serialport <NAME> <COMMAND> [ARG]    Available Commands:  Get the **interface** of a serial port  $ mx-**interface**-mgmt serialport <serialport\_name> get\_interface  Set the **interface** of a serial port  $ mx-**interface**-mgmt serialport <serialport\_name> set\_interface <serial\_interface>    Arguments:  serialport\_name: The number of serial port (e.g. 0, 1, 2, ......)  serial\_interface:  0 --> set to RS-232 mode  1 --> set to RS-485-2W mode  2 --> set to RS-422 mode  3 --> set to RS-485-4W mode |

## Drivers

### Drivers Applicable Table

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Available Models** | **it87\_gpio** | **it87\_serial** | **it87\_wdt** | **mxuport** | **mxu11x0** | **sdhci-pci** | **gpio-pca953x** | **hid-ft260** | **irigb** | **i915 (backport)** |
| **BXP-A100/BXP-C100** | O | O | O | O | X | \*[1] | X | X | X | X |
| **DRP-A100/DRP-C100** | O | O | O | O | X | X | X | X | X | X |
| **RKP-A110/RKP-C110** | O | O | O | O | X | X | X | X | X | X |
| **V3200** | O | O | O | X | X | X | O | X | X | X |
| **V3400** | O | O | O | X | X | X | O | X | X | X |
| **MPC-3000** | O | O | O | X | X | \*[2] | X | X | X | X |
| **DA-820E** | O | O | O | X | X | X | \*[3] | \*[4] | X | \*[5] |
| **DA-820C** | O | O | O | X | X | X | \*[3] | \*[4] | O | X |
| **DA-682C** | O | O | O | X | X | X | \*[3] | \*[4] | X | X |
| **DA-681C** | O | X | O | X | X | X | \*[3] | X | X | X |
| **DA-680** | O | X | O | X | O | X | X | X | O | X |
| **RKP-C220** | O | O | O | O | X | X | X | X | X | X |
| **BXP-A101** | O | O | O | X | X | \*[2] | X | X | X | X |

\*[1]: \*A100 on Debian 11

\*[2]: Debian 11

\*[3]: Debian 11, Debian 12, RHEL 9, CentOS 7.9

\*[4]: RHEL 9, CentOS 7.9

\*[5]: Ubuntu 22.04 LTS

### moxa-it87-gpio-driver

The purpose of **moxa-it87-gpio-driver** is controlling GPIO interface for **IT87xx Super I/O** chips, based on Linux kernel [drivers/gpio/gpio-it87.c](https://github.com/torvalds/linux/blob/v5.2/drivers/gpio/gpio-it87.c), removed label for Moxa utilities' compatibility and fix-up some issues.

**Kernel module information**



Once the **gpio\_it87** driver has been probed, the gpiochip interface /sys/class/gpio/gpiochip\* and /sys/class/gpio/gpio\* are created by driver.

E.g.

|  |
| --- |
| # cat /sys/**class**/gpio/gpiochip698/label  gpio\_it87  # cat /sys/**class**/gpio/gpio699/value  0 |

Thus, by read/write the gpio value, user can get/set the super IO gpio value.

|  |
| --- |
| If Linux kernel version >= 5.x, default uses the **libgpiod** to set/get set/get gpio value.  Alternatively, for Linux kernel version <= 3.x, default uses the **sys class gpio** to set/get gpio. |

### moxa-it87-serial-driver

IT87xx Super I/O chips support six standard serial ports and **RS485 automatic direction control (ADDC)**. This driver provide an interface under misc device for controlling serial register.

**Kernel module information**

****

Once the **it87\_serial** driver has been probed, the /sys/class/misc/it87\_serial/serial[p] interface are created by driver.

E.g.

|  |
| --- |
| # cat /sys/**class**/misc/it87\_serial/serial1/serial1\_rs485  0 |

If returns 0, the RS485 automatic direction control (ADDC) is disabled. Alternatively, if returns 1, the ADDC is enabled.

The **UART RS485 ADDC state** selection has been imported into **mx-uart-ctl** utility.

### moxa-it87-wdt-driver

Watchdog timer driver for ITE IT87xx environment control. The moxa-it87-wdt-driver is based on Linux kernel [drivers/watchdog/it87\_wdt.c](https://github.com/torvalds/linux/blob/v5.2/drivers/watchdog/it87_wdt.c) driver, and add kernel parameters to support Moxa platform’s hardware design.

**Kernel module information**

****

The watchdog device node /dev/watchdog0 is created by it87\_wdt driver.

Thex86 Linux SDK Wizard will default setup the watchdog daemon configuration file /etc/watchdog.conf and enable service for specific Linux distributions.

Default timeout of watchdog device is 60 seconds (maximum is 65535 seconds). If you want to change timeout value, you need to edit watchdog daemon config file /etc/watchdog.conf

e.g. watchdog timeout for 300 second:

|  |
| --- |
| watchdog-timeout = 300 |

### moxa-mxuport-driver

The purpose of moxa-mxuport-driver is MOXA UPort series driver. This driver remains traditional serial device properties and only dial-in ports will be created.

**Kernel module information**

****

The device name for each serial port is /dev/ttyUSBxx which xx is a sequence number maintained by USB subsystem.

The mxuport UART mode selection has been imported into **mx-uart-ctl** utility.

### moxa-mxu11x0-driver

The purpose of moxa-mxu11x0-driver is Moxa UPort 11x0 USB to Serial Hub driver. The driver can be used in the Linux kernel with the usbcore and usbserial modules.

**Kernel module information**

****

The device name for each serial port is /dev/ttyUSBxx which xx is a sequence number maintained by USB subsystem.

The mxu11x0 UART mode selection has been imported into **mx-uart-ctl** utility.

### moxa-sdhci-pci-driver

|  |
| --- |
| Only on BXP-A100 on Debian 11, for resolving the SD card detection issue. |

The purpose of **moxa-sdhci-pci-driver** is SDHCI on PCI bus interface driver.

Due to the SD host controller communicates with the CPU via SDIO, it would not initialize successfully on **Debian 11**.

Thus to resolve this issue, this driver add module parameter (enable\_probe\_cd\_gpio) to determine probe card detect gpio or not.

|  |
| --- |
| modprobe sdhci\_pci enable\_probe\_cd\_gpio=0 |

Or add modprobe configuration file: /lib/modprobe.d/sdhci-pci-option.conf

Kernel message and SD card interface:



### moxa-gpio-pca953x-driver

This driver is for PCA953x 4/8/16/24/40 bit I/O ports control.

**Kernel module information**

****

Once the gpio-pca953x driver has been probed, and bind with USB to i2c bridge (e.g. FT260 or CP2112), the gpiochip interface /sys/class/gpio/gpiochip\* and /sys/class/gpio/gpio\* are created by driver.

The example refers to **moxa-it87-gpio-driver** section.

### moxa-hid-ft260-driver

This driver is for USB to SMBus master bridge driver on FT260.

**Kernel module information**

****

**Add Udev Rules to Rebind FT260 Device**

To avoid the ft260 hid device is pre-bind to hid-generic subsystem, add udev rules to re-bind to ft260 driver.

Edit /etc/udev/rules.d/11-ft260-pca9535.rules

|  |
| --- |
| ACTION=="add", KERNEL=="0003:0403:6030.\*", SUBSYSTEM=="hid", DRIVERS=="hid-generic", \  RUN+="/bin/bash -c 'echo $kernel > /sys/bus/hid/drivers/hid-generic/unbind'", \  RUN+="/bin/bash -c 'echo $kernel > /sys/bus/hid/drivers/ft260/bind'" |

### moxa-irigb-driver

The IRIG-B driver is for Moxa embedded compute for controlling the IRIG-B device.

**Kernel module information**

****

The IRIG-B driver is depends on IRIG-B Utility.

### intel-gpu-i915-backports

Intel® Graphics Driver Backports for Linux® OS (intel-gpu-i915-backports)

Contains the backported kernel module source code of intel GPUs on various OS distributions and LTS Kernels.

**Kernel module information**

****

Use lspci -v to check i915 driver is in use



## Libraries

### libgpiod

libgpiod - C library and tools for interacting with the **Linux GPIO character device** (gpiod stands for GPIO device).

Since **Linux kernel 4.8** the GPIO sysfs interface is deprecated. User space should use the character device instead. This library encapsulates the ioctl calls and data structures behind a straightforward API.

The new character device interface guarantees all allocated resources are freed after closing the device file descriptor and adds several new features that are not present in the obsolete sysfs interface.

One device file per gpiochip

* /dev/gpiochip0, /dev/gpiochip1, …, /dev/gpiochipX

Usage:

|  |
| --- |
| There are currently six command-line tools available:    \* gpiodetect - list all gpiochips present on the system, their names, labels  and number of GPIO lines    \* gpioinfo - list all lines of specified gpiochips, their names, consumers,  direction, active state and additional flags    \* gpioget - read values of specified GPIO lines    \* gpioset - set values of specified GPIO lines, potentially keep the lines  exported and wait until timeout, user input or signal    \* gpiofind - find the gpiochip name and line offset given the line name    \* gpiomon - wait **for** events on GPIO lines, specify which events to watch,  how many events to process before exiting or **if** the events  should be reported to the console |

E.g.

|  |
| --- |
| # Read the value of a single GPIO line.  $ gpioget gpiochip1 23  0    # Read two values at the same time. Set the active state of the lines to low.  $ gpioget --active-low gpiochip1 23 24  1 1    # Set the value of a single line, then exit immediately.  # This is useful **for** floating pins.  $ gpioset gpiochip1 23=1 |

# 5. Basic Linux Concepts

The section introduces basic Linux concepts, like x86 secure boot, IO interfaces, TPM2 module, SD card slot mounting, Linux PTP (IEEE 1588), etc.

To provide skills and basic information for newcomers to learn more about Linux.

## 5.1. Mounting SD Card Slot on BXP/DRP Series

The **BXP** and **DRP** series support one SD card slot (SD 3.0 interface (SDHC/SDXC)).

The differences of hardware design between these model are:

* **BXP-A100**: SDIO interface
* **BXP-C100/DRP-A100/DRP-C100**: USB to SD Bridge IC (USB2244)

Make sure your SD card is inserted into the SD card slot on your computer, the kernel message should be shown:

For **BXP-A100**:



To **mount** the SD Card:

The block devices /dev/mmcblk1, the block device is created from sdhci driver.

Then, user can create a mount point on directory (e.g. /mnt): sudo mount /dev/mmcblk1p1 /mnt

For **BXP-C100/DRP-A100/DRP-C100**:



The block devices /dev/sdX, where "X" is a letter indicating the specific device (e.g., /dev/sdb, /dev/sdc, etc.).

Then, user can create a mount point on directory (e.g. /mnt): sudo mount /dev/sdX /mnt

## 5.2. Secure Boot

The **UEFI Secure Boot** is a security feature that has been widely adopted in modern computer systems, especially those running Windows and some Linux distributions.

Its primary purpose is to ensure the integrity and authenticity of the operating system and bootloader during the system boot process, protecting the system against boot-time malware and other unauthorized software.

**Secure Boot Purpose**

Secure Boot is designed to prevent the loading of malicious software, such as rootkits and bootkits, during the boot process.

It does this by ensuring that only trusted and digitally **signed** bootloaders and OS kernels are executed.

Thus, if user loads **unsigned** bootloaders and OS kernels on target Linux distributions when UEFI secure boot has been enabled on BIOS menu, the boot process or kernel modules should be failed due to unauthorized policy.

**Operating System Support**

User can be considered to refer to the following website links for more UEFI secure boot information.

* [Debian Secure Boot](https://wiki.debian.org/SecureBoot)
* [Ubuntu Secure Boot](https://wiki.ubuntu.com/UEFI/SecureBoot)
* [RedHat Secure Boot](https://access.redhat.com/documentation/zh-tw/red_hat_enterprise_linux/8/html/managing_monitoring_and_updating_the_kernel/signing-a-kernel-and-modules-for-secure-boot_managing-monitoring-and-updating-the-kernel)

## 5.3. Linux PTP (IEEE 1588)

The **Precision Time Protocol (PTP)** is a protocol used to synchronize clocks throughout a computer network. PTP provides higher precision and faster synchronization than NTP even without hardware support. With hardware support, sub-microsecond accuracy can be expected.

Whereas NTP is intended for WAN use, PTP is designed for LAN environments and makes use of UDP multicast.

**Available LAN chip**

* Intel I210 (driver: igb)
* Intel I219 (driver: e1000e)

**Debian Linuxptp package**

**Linuxptp package** is an implementation of the Precision Time Protocol (PTP) according to IEEE standard 1588 for Debian Linux. Features include:

1. support for hardware and software time stamping via the Linux SO\_TIMESTAMPING socket option.
2. support for the Linux PTP Hardware Clock (PHC) subsystem by using the clock\_gettime family of calls, including the new clock\_adjtimex system call
3. implementation of **Boundary Clock (BC)** and **Ordinary Clock (OC)**
4. transport over UDP/IPv4, UDP/IPv6, and raw Ethernet (Layer 2)
5. support for IEEE 802.1AS-2011 in the role of end station

**Debian phc2sys program**

phc2sysis a program which synchronizes two or more clocks in the system. Typically, it is used to synchronize the system clock to a PTP hardware clock (PHC), which itself is synchronized by the ptp4l(8) program. See [manpage](https://manpages.debian.org/bullseye/linuxptp/phc2sys.8.en.html) for more information.

* **Prerequisite**
  + Install **Debian 11** or later version
  + Install **Linuxptp** package: apt update && apt install linuxptp
  + Stop and disable systemd time sync daemon service to avoid some unexpected operations: systemctl stop systemd-timesyncd && systemctl disable systemd-timesyncd

### Example for Linux PTP setting up

#### [Ordinary Clock (OC) mode]

Set as **OC master** mode: Layer 2, P2P mode, peer delay mechanism

|  |
| --- |
| # Assume A side **interface** device is 'enp4s0'  ip link set dev enp4s0 up  ptp4l -m -2 -P -i enp4s0 |

Set as **OC slave** mode: Layer 2, P2P mode, peer delay mechanism

|  |
| --- |
| # Assume B side **interface** device is 'enp5s0'  ip link set dev enp5s0 up  ptp4l -m -2 -P -s -i enp5s0  # or with log: ptp4l -m -2 -s -P -i enp5s0 2>&1 | tee $(date +%Y%m%d%H%M%S.log)    # use phc2sys to sync sys clock **for** 10Hz  phc2sys -a -m -r -R 10 |

#### [Boundary Clock (BC) mode]

Set as **BC mode** host

* clock\_type Specifies the kind of PTP clock. Valid values are "OC" for ordinary clock, "BC" for boundary clock, "P2P\_TC" for peer to peer transparent clock, and "E2E\_TC" for end to end transparent clock. An multi-port ordinary clock will automatically be configured as a boundary clock. The default is "OC".
* boundary\_clock\_jbod When running as a **boundary clock** (that is, when more than one network interface is configured), ptp4l performs a sanity check to make sure that all of the ports share the same hardware clock device. This option allows ptp4l to work as a boundary clock using "just a bunch of devices" that are not synchronized to each other. For this mode, the collection of clocks must be synchronized by an external program, for example phc2sys(8) in "automatic" mode. The default is 0 (disabled).

**Example for BC mode**

|  |
| --- |
| # For example, edit config file 'bc.cfg'  # and assume 'enp12s0' and 'enp4s0' are connected network **interface**  [global]  sanity\_freq\_limit 0  step\_threshold 0.000002  tx\_timestamp\_timeout 10  logMinPdelayReqInterval 0  logSyncInterval 0  logAnnounceInterval 0  announceReceiptTimeout 3  syncReceiptTimeout 2  twoStepFlag 1  summary\_interval 0  clock\_type BC  priority1 128  priority2 127  delay\_mechanism P2P    [enp12s0]  boundary\_clock\_jbod 1  network\_transport L2  fault\_reset\_interval 0    [enp4s0]  boundary\_clock\_jbod 1  network\_transport L2  fault\_reset\_interval 0    # run the ptp4l procedure  ip link set dev enp12s0 up  ip link set dev enp4s0 up  ptp4l -m -f bc.cfg    # use phc2sys to sync sys clock **for** 10Hz  phc2sys -a -m -r -R 10 |

On OC **Grandmaster**

|  |
| --- |
| # assume **interface** is enp5s0  ip link set dev enp5s0 up  ptp4l -2 -m -P -i enp5s0 |

On OC **Slave**

|  |
| --- |
| # assume **interface** is enp4s0  ip link set dev enp4s0 up  ptp4l -2 -m -s -P -i enp4s0  # with log: ptp4l -2 -m -s -P -i enp4s0 2>&1 | tee $(date +%Y%m%d%H%M%S.log) |

#### [Transparent Clock (TC) mode]

**Set TC mode host**

|  |
| --- |
| # For example, edit config file 'tc.cfg'  # and assume 'enp12s0' and 'enp4s0' are connected network **interface**  [global]  priority1 254  priority2 253  free\_running 1  freq\_est\_interval 3  tc\_spanning\_tree 1  clock\_type P2P\_TC  network\_transport L2  delay\_mechanism P2P    [enp12s0]  egressLatency 0  ingressLatency 0  delay\_mechanism P2P  network\_transport L2    [enp4s0]  egressLatency 0  ingressLatency 0  delay\_mechanism P2P  network\_transport L2    # run the ptp4l procedure  ip link set dev enp12s0 up  ip link set dev enp4s0 up  ptp4l -m -f tc.cfg    # use phc2sys to sync sys clock between master & slave **for** 10Hz  # -c Specify the slave clock by device (e.g. /dev/ptp1) or **interface** (e.g. eth1)  # -s Specify the master clock by device (e.g. /dev/ptp0) or **interface** (e.g. eth0)  phc2sys -s enp12s0 -c enp4s0 -O 0 -R 10 -m |

**As OC Grandmaster**

|  |
| --- |
| # assume **interface** is enp5s0  ip link set dev enp5s0 up  ptp4l -2 -m -P -i enp5s0 |

**As OC Slave**

|  |
| --- |
| # assume **interface** is enp4s0  ip link set dev enp4s0 up  ptp4l -2 -m -s -P -i enp4s0    # use phc2sys to sync sys clock **for** 10Hz on slve  phc2sys -a -m -r -R 10 |

# 6. Troubleshooting

The troubleshooting section provides fundamental skills for system logging, debugging, the debug of Moxa x86 SDK Wizard and issues tracing.

## 6.1. How to Print Kernel Message from Linux Environment

The dmesg command is used to display the kernel ring buffer, which contains messages related to the kernel and hardware events.

It's a useful tool for troubleshooting hardware-related issues, monitoring system-level events and diagnosing hardware issues.

To simply view the kernel ring buffer, run the following command: *dmesg*

You can save the output of dmesg to a file for further analysis. For instance, to save the log to a file named kernel.log, use the following command:

|  |
| --- |
| # save kernel message to log  dmesg >kernel.log    # or simply to save the error and warninglevel log:  dmesg --level=err,warn > kernel\_err\_warn.log |

## 6.2. How to Collect Systems Logs from Linux Environment

The following procedure describes the collecting of log files. Log files in the /var/log directory.

Archive and compress all log files and put them in /tmp

|  |
| --- |
| tar czvf /tmp/varlog.tar.gz /var/log/\*.log.\* |

The output file /tmp/varlog.tar.gz can be transferred for debugging usage.

## 6.3. How to Get Installation Logs from Moxa x86 Linux SDK Install Wizard

**Moxa x86 Linux SDK** provides **self-test** for diagnosing the status of drivers and tools after installation. To simply see the log, run the following command:

|  |
| --- |
| ./install.sh --selftest |

Then the self test cases will check the SDK status and print on terminal, for example:



For further, the log of installation is also created on Moxa\_x86\_Linux\_Install\_Wizard\_<version>\_Build\_<build\_date>/install.log

User can consider to view the log file and check issues.

## 6.4. How to Get Hardware Information on Host

IOS exports the hardware information on **DMI** (Desktop Management Interface) table.

Linux **dmidecode** is a tool for dumping a computer DMI (some say **SMBIOS**) table contents in a human-readable format. This table contains a description of the system’s hardware components, as well as other useful pieces of information such as serial numbers and BIOS revision.

**Install dmidecode Package**

* Ubuntu/Debian: sudo apt-get install dmidecode
* RHEL: sudo yum install dmidecode

**Example**

**[Get model name and hardware version]**

The Option 1 (or Option 2) displays the 16 bytes information, for example: RKP A110000091

RKP A110000091 means

* PCBA name = RKP
* PCBA number = A110
* PCBA serial = 0
* PCBA type = 00
* PCBA hw version = 091 (v0.91)

How to get information from dmitable

|  |
| --- |
| # dmidecode -t 12  Handle 0x0021, DMI type 12, 5 bytes  System Configuration Options  Option 1: RKP A110000091  Option 2:  Option 3:  ... |



**[Get current BIOS version]**

|  |
| --- |
| # dmidecode -t bios  BIOS Information  Vendor: INSYDE Corp.  Version: V1.0.0S04  Release Date: 05/15/2023  Address: 0xE0000  Runtime Size: 128 kB  ROM Size: 10 MB  ... |

**[Get memory and processor hardware information]**

|  |
| --- |
| # dmidecode -t memory  Physical Memory Array  Location: System Board Or Motherboard  Use: System Memory  Error Correction Type: None  Maximum Capacity: 16 GB  Error Information Handle: Not Provided  Number Of Devices: 2  ...  # sudo dmidecode -t processor  Processor Information  Socket Designation: U3E1  Type: Central Processor  Family: Other  Manufacturer: Intel(R) Corporation  ID: 61 06 09 00 FF FB EB BF  Version: Intel Atom(R) x6425E Processor @ 2.00GHz  Voltage: 1.1 V  External Clock: 100 MHz  ... |

# 7. Appendix

## 7.1. The License/Commercial-Use of Linux Distributions

A Linux distribution is a version of the Linux operating system that includes the Linux kernel, system utilities, libraries, and additional software and applications. Linux distributions are created by various organizations, communities, and individuals, each tailoring the operating system to meet specific needs and preferences.

Linux distribution include:

### Debian

Debian is a free and open-source operating system, and its intellectual property rights policy is based on a commitment to free software principles. Debian adheres to a set of guidelines and policies outlined in the Debian Free Software Guidelines (DFSG). The DFSG defines the criteria that software must meet to be considered "free" in the context of Debian.

Commercial use:

Free redistribution.

The license of a Debian component **may not restrict** any party from selling or giving away the software as a component of an aggregate software distribution containing programs from several different sources. The license may not require a royalty or other fee for such sale.

<https://wiki.debian.org/DebianFreeSoftwareGuidelines>

<https://www.debian.org/trademark>

<https://wiki.debian.org/ProposedTrademarkPolicy>

### Ubuntu

Ubuntu is built by Canonical and the Ubuntu community. We share access rights owned by Canonical with the Ubuntu community for the purposes of discussion, development and advocacy. We recognise that most of the open source discussion and development areas are for non-commercial purposes and we therefore allow the use of Canonical IP in this context, as long as there is no commercial use and that the Canonical IP is used in accordance with this IPRights Policy.

You can modify Ubuntu for **personal** or **internal commercial** use.

You can **redistribute** Ubuntu, but only where there has been **no modification** to it.

For more Canonical’s intellectual property rights policy:

<https://ubuntu.com/legal/intellectual-property-policy>

### Red Hat Enterprise Linux (RHEL)

Red Hat Enterprise Linux (RHEL) is a **commercial** Linux distribution provided by Red Hat, Inc. It is designed for enterprise environments and comes with a subscription-based pricing model.

<https://www.redhat.com/en/store/linux-platforms>

<https://www.redhat.com/en/about/trademark-guidelines-and-policies>

<https://www.redhat.com/en/about/terms-use>

### CentOS

The CentOS Linux and CentOS Stream distributions are compilations of software packages. Each package is governed by its own license. The CentOS Linux and CentOS Stream compilation copyright is licensed under GPLv2. To the extent you hold any copyright in the selection, coordination, or arrangement of packages making up the CentOS Linux or CentOS Stream distributions, you license that copyright under GPLv2.

<https://www.centos.org/legal/licensing-policy/>

# 8. Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Version** | **Date** | **Author** | **Revision Notes** |
| 2.0 | 2024/03/12 | Elvis CW Yao | First created. |
| 2.1 | 2024/09/05 | Elvis CW Yao  Wilson YS Huang  Henry LC Chen | * Add the following model support   + V3200 series   + MPC-3000 series   + DA-820E series   + DA-820C series   + DA-682C series   + DA-681C series   + DA-680 series * Add Debian 12, Ubuntu 20.04 into Linux OS list * Add LTE (mPCIe slot) Module Control section * Add HSR/PRP Utility section * Add IRIG-B Utility section * Add MCIM wrapper section * Add Scaler Utility section * Add MCU upgrade tool section * Add moxa-gpio-pca953x-driver section * Add moxa-mxu11x0-driver section * Add moxa-hid-ft260-driver section * Add moxa-irigb-driver section * Add intel-gpu-i915-backports section |
| 2.2 | 2024/11/12 | Elvis CW Yao  Wilson YS Huang | * Add the following model support   + V3400 series   + BXP-A101 series * Add disk hotswap daemon * Add Moxa MCU Manager |