DRP/BXP/RKP Series Computers Linux Software Manual

Version 2.0, April 2024

www.moxa.com/products

DRP/BXP/RKP Series Computers Linux Software Manual

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

Copyright Notice

Trademarks

Disclaimer

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

Technical Support Contact Information

www.moxa.com/support

Table of Contents

1. Introduction

This user manual is applicable to Moxa's new-generation of x86 computer listed below. Detailed instructions on getting started with your x86 computer, using the x86 Linux SDK wizard, configuring peripheral interfaces, and troubleshooting are covered in this manual.

Applicable Products

- BXP-A100 Series
- BXP-C100 Series
- DRP-A100 Series
- DRP-C100 Series
- RKP-A110 Series
- RKP-C110 Series

2. Getting Started

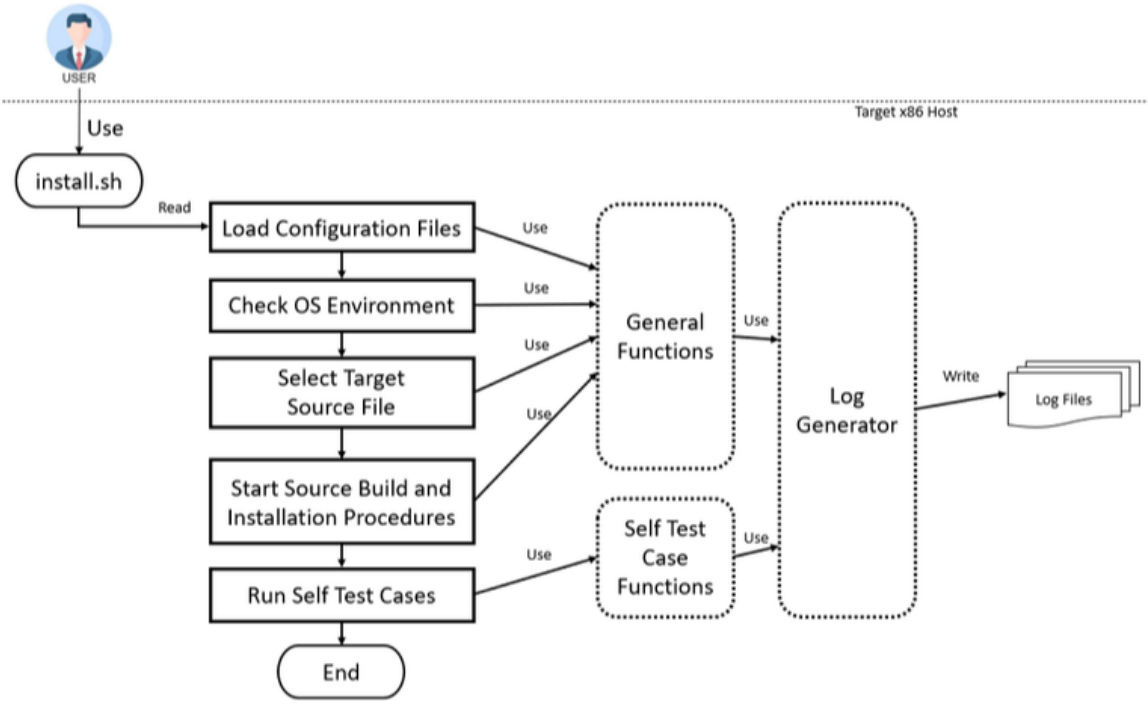
In this chapter, we introduce the Linux distributions supported and how to install and get started on your x86 computer.

Installing Linux on Your x86 Computer

Prerequisites

- Configure your **network settings** before installation.
- Prepare a **USB storage drive**
  - Download the ISO image file and restore it to a USB storage drive.
  - Youn can use the Rufus tool to create a bootable USB drive.
- Extract the **tgz tarball** file in a Linux environment (e.g. **`tar xvf *.tar.gz`**)
- Check the system status.
  - Run the **`--dry-run`** option before installation to check the availability of the target host device and environment.
  - Run the **`--selftest`** option after installation, to check the status of drivers and tools

User Interface	Main Command	Sub Command	Option	Description
`install.sh`				Start to install all procedures (default)
			-y, --yes	Automatic yes to prompts
	-h, --help			Display the help menu
	-v, --version			Display the version information
	-s, --selftest			Run the self test cases
	--uninstall			Uninstall driver and tool
	--dry-run			It won't perform the installation, list available driver and tool only
			--force	Install driver and tool even if the version is the same or older (default is to install newer version)



Linux Distributions Supported

- **Debian**
  - **Debian 11 (bullseye), Linux kernel 5.10**
    - Official Debian 11.8 netinst ISO download link
    - Official Debian installation guide
- **Ubuntu**
  - **Ubuntu 22.04.03 LTS (Jammy Jellyfish), Linux kernel 5.15**
    - Official Ubuntu 22.04.03 LTS desktop ISO download link
    - Official Ubuntu 22.04.03 LTS server ISO download link
    - Official Ubuntu installation guide
- **RedHat**
  - **RedHat 9, Linux kernel 5.14**
    - Official RedHat 9 download link
    - Official RedHat 9 installation guide
- **CentOS 7**
  - **CentOS 7.9, Linux kernel 3.10**
    - CentOS-7-x86_64-DVD-2009.iso download link

Installing the x86 Linux SDK Wizard

The **Moxa x86 Linux SDK** enables easy deployment of the Moxa x86 IPC platform functions. The SDK contains peripheral drivers, peripheral control tools, and configuration files. The SDK also provides deployment features, such as build & installation log, dry-run, and self test. The Moxa x86 Linux SDK zip file is available for download on the official Moxa website.

The SDK includes the following files:

- ***.tgz**: The tarball file for the 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

NOTE

Extract the **tgz** tarball file in a Linux OS environment to avoid file permission issues.

SDK Versions and Linux Distributions Supported

Model Name	Available SDK Version	Linux Distributions Supported
BXP-A100/BXP-C100	V1.1	Debian 11, Ubuntu 22.04.03 LTS (Jammy Jellyfish), RedHat 9, CentOS 7.9
DRP-A100/DRP-C100	V1.1	Debian 11, Ubuntu 22.04.03 LTS (Jammy Jellyfish), RedHat 9, CentOS 7.9
RKP-A110/RKP-C110	V1.1	Debian 11, Ubuntu 22.04.03 LTS (Jammy Jellyfish), RedHat 9, CentOS 7.9

Connecting to Your x86 Computer

To boot up your device and enter the BIOS menu, do the following:

1. Press the **F2** key until you enter the BIOS menu.
2. Select to **boot from USB** from **UEFI mode**.
3. Follow the distribution's official installation guide to finish the OS installation procedure.

3. Configuring Peripherals

In this chapter, we introduced the usage of the **Moxa peripheral interface control utility** package.

> **NOTE**
>
> The Moxa peripheral interface control utility package should be installed after the x86 Linux SDK Wizard installation procedure is completed.

The utility package consists of the serial port and DIO utilities. You can check the status of the utilities by running the following command:

```
./install.sh --selftest
```

The products supported by the Moxa peripheral interface control utility package are listed in the following table:

Series	Serial Port Utility	DIO Utility
BXP-A100/BXP-C100	✓	✓
DRP-A100/DRP-C100	✓	N/A
RKP-A110/RKP-C110	✓	✓

Configuring Serial Ports

The serial ports support RS-232, RS-422, RS-485 2-wire, and RS-485 2-wire operation modes with flexible baudrate settings. The default operation mode is RS-232. You can use the **`mx-uart-ctl`** command to change the operation mode.

Syntax

```
mx-uart-ctl -p <port_number> [-m <uart_mode>]
```

Port Number

0, 1, 2, 3…

UART Mode

Option	UART Mode
None	Set target port to UART mode
0	RS-232
1	RS-485 2-wire
2	RS-422
3	RS-485 4-wire

Drivers Dependency

- moxa-it87-gpio-driver
- moxa-it87-serial-driver
- moxa-mxuport-driver

Libraries Dependency

- libgpiod

For details,see **Installing Drivers for Interfaces** and **Installing Libraries.**

Example

```
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 0Current uart mode is RS422/RS485-4W
interface.
```

Configuring Digital I/Os (DIOs)

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

Syntax

```
mx-dio-ctl <-i|-o <#port number> [-s <#state>]>
```

State

- 0 Low
- 1 High

Drivers dependency

- moxa-it87-gpio-driver

Libraries dependency

- libgpiod

For details,see **Installing Drivers for Interfaces** and **Installing Libraries.**

Example

```
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
```

Installing Drivers for Interfaces

The x86 Linux SDK Install Wizard includes drivers for GPIO, RS-485 Automatic Direction Control (ADDC), Watchdog timer, and Moxa UPort.

Available Models	it87_gpio	it87_serial	it87_wdt	mxuport
BXP-A100/BXP-C100	✓	✓	✓	✓
DRP-A100/DRP-C100	✓	✓	✓	✓
RKP-A110/RKP-C110	✓	✓	✓	✓

GPIO Interface

The Moxa **gpio_it87 driver** is used to manage the GPIO interface for **IT87xx Super I/O** chips and is based on the Linux kernel driver drivers/gpio/gpio-it87.c. The gpio_it87 driver in the Moxa utility package includes some additional changes to fix issues and remove the Moxa label.

The kernel module information is shown below:

```
root@moxa-ElkhartLake-U:/home/moxa# modinfo gpio_it87
filename:       /lib/modules/5.19.0-50-generic/kernel/drivers/gpio/gpio-it87.ko
version:        1.5.0
license:        GPL
description:    GPIO interface for IT87xx Super I/O chips
author:         Diego Elio Pettenò <flameeyes@flameeyes.eu>
srcversion:     BF1E1DA11ED46916F0525B3
depends:
retpoline:      Y
name:           gpio_it87
vermagic:       5.19.0-50-generic SMP preempt mod_unload modversions
parm:           force_id:Override the detected device ID (ushort)
```

Once the **gpio_it87** driver has been probed, the gpiochip interfaces **`/sys/class/gpio/gpiochip*`** and **`/sys/class/gpio/gpio*`** are created by the driver.

You can read/write the gpio value to get/set a super IO gpio interface. For details, see drivers/gpio/gpio-it87.c.

Example

```
# cat /sys/class/gpio/gpiochip698/label
gpio_it87
# cat /sys/class/gpio/gpio699/value
0
```

NOTE

If the Linux kernel version ≥ 5.x, the **libgpiod** library is used by default to set/get the gpio value.

For Linux kernel version ≤ 3.x, the **sys class gpio** is used by default to set/get the gpio value.

Serial: RS-485 Automatic Direction Control (ADDC)

The IT87xx Super I/O chip support 6 standard serial modes and the **RS485 automatic direction control (ADDC)** mode. This driver provides an interface under misc device for controlling the serial register.

The kernel module information is shown below:

```
root@moxa-ElkhartLake-U:/home/moxa# modinfo it87_serial
filename:       /lib/modules/5.19.0-50-generic/kernel/drivers/misc/it87_serial.ko
version:        1.4.1
license:        GPL
author:         Remus Wu <remusty.wu@moxa.com>
description:    Serial Port Register Control for IT8786 Super I/O chips
softdep:        pre: it87
srcversion:     DF70894844D938C398F1E94
depends:
retpoline:      Y
name:           it87_serial
vermagic:       5.19.0-50-generic SMP preempt mod_unload modversions
parm:           force_id:Override the detected device ID (ushort)
```

Once the **it87_serial** driver has been probed, the **`/sys/class/misc/it87_serial/serial[p]`** interface is created by the driver.

Example

```
# cat /sys/class/misc/it87_serial/serial1/serial1_rs485
0
```

If 0 is returned, the RS-485 automatic direction control (ADDC) is disabled. If 1 is returned, the ADDC mode is enabled. The **UART RS-485 ADDC state** selection is imported into the **mx-uart-ctl** utility.

Watchdog-timer Interface

The watchdog timer driver is used to set the watchdog timer in the ITE IT87xx environment. The moxa-it87-wdt-driver is based on the Linux kernel drivers/watchdog/it87_wdt.c driver. The it87_wdt.c driver in the Moxa utility package includes some additional changes to support the Moxa hardware platform.

The kernel module information is shown below:

```
root@moxa-ElkhartLake-U:/home/moxa# modinfo it87_wdt
filename:       /lib/modules/5.19.0-50-generic/kernel/drivers/watchdog/it87_wdt.ko
version:        1.5.0
license:        GPL
description:    Hardware Watchdog Device Driver for IT87xx EC-LPC I/O
author:         Oliver Schuster
srcversion:     539E4978F03512C150A3753
depends:
retpoline:      Y
name:           it87_wdt
vermagic:       5.19.0-50-generic SMP preempt mod_unload modversions
parm:           timeout:Watchdog timeout in seconds, default=60 (int)
parm:           testmode:Watchdog test mode (1 = no reboot), default=0 (int)
parm:           nowayout:Watchdog cannot be stopped once started, default=0 (bool)
parm:           krst:Watchdog enable KRST reset output, default=1 (bool)
parm:           ldn_reset:Set SIO LDN back to 01h when init and update_timeout, default=0 (bool)
parm:           force_id:Override the detected device ID (ushort)
```

The watchdog device node **`/dev/watchdog0`** is created by the **it87_wdt** driver.

The x86 Linux SDK Wizard will by default setup the watchdog daemon configuration file **/etc/watchdog.conf** and enable the service for specific Linux distributions. The default timeout of watchdog device is 60 seconds (maximum is 65535 seconds). If you want to change timeout value, edit the watchdog daemon config file **/etc/watchdog.conf.**

Example

Watchdog timeout after 300 seconds

```
watchdog-timeout = 300
```

Moxa UPort

The purpose of **mxuport**-driver is to support the Moxa UPort series driver. This driver retains traditional serial device properties and only dial-in ports will be created.

The kernel module information is shown below:

```
root@moxa-ElkhartLake-U:/home/moxa# modinfo mxuport
filename:       /lib/modules/5.19.0-50-generic/misc/mxuport.ko
license:        GPL
description:    MOXA UPort series driver
author:         Danny Lin <danny.lin@moxa.com>
srcversion:     95402A0905F4FBBACF95A11
alias:          usb:v110Ap7003d*dc*dsc*dp*ic*isc*ip*in*
alias:          usb:v110Ap7002d*dc*dsc*dp*ic*isc*ip*in*
alias:          usb:v110Ap0850d*dc*dsc*dp*ic*isc*ip*in*
alias:          usb:v110Ap0450d*dc*dsc*dp*ic*isc*ip*in*
alias:          usb:v110Ap0250d*dc*dsc*dp*ic*isc*ip*in*
```

The device name for each serial port is **/dev/ttyUSB***xx*, where xx is a sequence number maintained by the USB subsystem. The mxuport UART mode selection has been imported into the **mx-uart-ctl** utility.

PCI Interface

> **NOTE**
>
> This driver is only available on BXP-A100 with 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:

```
# dmesg
[83967.247209] sdhci: Secure Digital Host Controller Interface driver
[83967.247212] sdhci: Copyright(c) Pierre Ossman
[83967.249643] sdhci-pci 0000:00:1a.0: SDHCI controller found [8086:4b47] (rev 11)
[83967.250181] sdhci-pci 0000:00:1a.0: disable card detect gpio from setup
[83967.250229] mmc0: CQHCI version 5.10
[83967.250363] mmc0: SDHCI controller on PCI [0000:00:1a.0] using ADMA 64-bit
[83967.250390] sdhci-pci 0000:00:1a.1: SDHCI controller found [8086:4b48] (rev 11)
[83967.251508] sdhci-pci 0000:00:1a.1: disable card detect gpio from setup
```

```
# ls -l  /sys/class/mmc_host/mmc*
lrwxrwxrwx 1 root root 0 Nov 30 11:08 /sys/class/mmc_host/mmc0 -> ../../devices/pci0000:00/0000:00:1a.0/mmc_host/mmc0
lrwxrwxrwx 1 root root 0 Nov 30 11:08 /sys/class/mmc_host/mmc1 -> ../../devices/pci0000:00/0000:00:1a.1/mmc_host/mmc1
```

Installing Libraries

Linux GPIO Character Device

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
```

Example

```
# 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
```

Mounting the SD Card Slot (BXP and 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**:

```
root@moxa:~# dmesg | grep sdhci
[    1.569095] sdhci: Secure Digital Host Controller Interface driver
[    1.569098] sdhci: Copyright(c) Pierre Ossman
[    1.570901] sdhci_pci: loading out-of-tree module taints kernel.
[    1.570945] sdhci_pci: module verification failed: signature and/or required key missing - tainting kernel
[    1.571276] sdhci-pci 0000:00:1a.0: SDHCI controller found [8086:4b47] (rev 11)
[    1.571807] sdhci-pci 0000:00:1a.0: disable card detect gpio from setup
[    1.572551] sdhci-pci 0000:00:1a.1: SDHCI controller found [8086:4b48] (rev 11)
[    1.576861] sdhci-pci 0000:00:1a.1: disable card detect gpio from setup
```

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**:

```
[ 2507.486612] usb 1-4: new high-speed USB device number 5 using xhci_hcd
[ 2507.614763] usb 1-4: New USB device found, idVendor=05e3, idProduct=0761, bcdDevice=24.04
[ 2507.614769] usb 1-4: New USB device strings: Mfr=0, Product=1, SerialNumber=2
[ 2507.614772] usb 1-4: Product: USB Storage
[ 2507.614775] usb 1-4: SerialNumber: 000000002404
[ 2507.651199] usb-storage 1-4:1.0: USB Mass Storage device detected
[ 2507.651428] scsi host2: usb-storage 1-4:1.0
[ 2507.651496] usbcore: registered new interface driver usb-storage
[ 2507.653051] usbcore: registered new interface driver uas
[ 2508.655796] scsi 2:0:0:0: Direct-Access     Generic  MassStorageClass 2404 PQ: 0 ANSI: 6
[ 2508.656130] sd 2:0:0:0: Attached scsi generic sg1 type 0
[ 2509.593552] sd 2:0:0:0: [sdb] 31260672 512-byte logical blocks: (16.0 GB/14.9 GiB)
[ 2509.594597] sd 2:0:0:0: [sdb] Write Protect is off
[ 2509.594602] sd 2:0:0:0: [sdb] Mode Sense: 21 00 00 00
[ 2509.595470] sd 2:0:0:0: [sdb] Write cache: disabled, read cache: enabled, doesn't support DPO or FUA
[ 2509.601096]  sdb: sdb1
[ 2509.603857] sd 2:0:0:0: [sdb] Attached SCSI removable disk
```

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`**

4. Configuring 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
- Ubuntu Secure Boot
- RedHat Secure Boot

5. Precision Time Protocol (PTP) Based on 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: ibg)
- 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 implementation of **Boundary Clock (BC)** and **Ordinary Clock (OC)** transport over UDP/IPv4, UDP/IPv6, and raw Ethernet (Layer 2)
3. Support for IEEE 802.1AS-2011 in the role of end station

Debian phc2sys program

`phc2sys` is 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 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.

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
```

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.

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:

```
[info] Product Name: RKPA110
[info] OS Name: Ubuntu
[info] OS Version: 22.04
[info] Kernel Info: Linux moxa-ElkhartLake-U 5.19.0-50-generic #50-Ubuntu SMP PREEMPT_DYNAMIC Mon Jul 10 18:24:29 UTC 2023 x86_64 x86_64 x86_64 GNU/Linux
[info] >>> Execute hook script "self-test.sh".
[info] ------------------------------------------------------------------------
[info]  Name                          Installed      Status         Version
[info] ========================================================================
[info]  moxa-it87-gpio-driver                                       5.2+1.5.0-1
[info]  - gpio_it87                   Yes            Loaded
[info]  moxa-it87-wdt-driver                                        5.2+1.5.0-1
[info]  - it87_wdt                    Yes            Loaded
[info]  - watchdog service            Yes            Active
[info]  moxa-it87-serial-driver                                     1.4.1+u2
[info]  - it87_serial                 Yes            Loaded
[info]  moxa-mxuport-driver                                         5.1.1_build_23080316
[info]  - mxuport                     Yes            Loaded
[info]  moxa-x86-control-tools                                      1.8.1
[info]  - mx-uart-ctl                 Yes            10 ports
[info]  - mx-dio-ctl                  Yes            8 DI / 8 DO
[info] ------------------------------------------------------------------------
[info] <<< Execute hook script "self-test.sh" done.
```

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.

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:
...
```

BYTE	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Define	PCBA Nmae (Eng					PCBA Name (Number					Serial	Type		PCBA version		
Example : UC-8580 Main board PCBA : 1.0a	UC					8580					0	00		10a		

[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

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/