

PCI/PCIe Serial Card

Linux User Manual



Document Information

Information	Content
Abstract	This manual describes how to install and uninstall the Linux driver for PCI/PCIe Multi-port card series.
Version History	V1.8, Written by Je-hwan Yoo on April 4 th ,2013
	V1.9, Written by Je-hwan Yoo on May 14 th ,2013
	V2.0, Written by Je-hwan Yoo on Jan 15 th ,2014
	V2.1, Written by Je-hwan Yoo on Jan 30 th ,2014
	V2.2, Written by Je-hwan Yoo on Feb 7 th ,2014
	V2.3, Written by Won Lee on Jan 11 th ,2016
	V2.4, Written by Won Lee on Aug 2 nd ,2018
	V2.5, Written by Jae-hyuk Kim on Dec 12 th ,2018
	V2.6, Written by Yong-ho Moon on Feb 26 th ,2020
	V2.7, Written by Yong-ho Moon on May 26 th ,2025
	V2.8, Written by Yong-ho Moon on Jul 3 th ,2025

Contents

1. Introduction.....	4
2. Supported Linux Kernel Versions	4
3. Release Note.....	5
4. Required for driver installation	6
5. Installation.....	8
6. Remove	9
7. Testing.....	10

1. Introduction

Linux driver for PCI/PCIe Multi-port card series is provided in the form of a script, you can install the driver without other operations.

2. Supported Linux Kernel Versions

Supported Kernel: From version 2.4.18 to 6.16.5

Tested Linux		
Linux distribution	Kernel version	Architecture
CentOS Stream 9	5.14.0	x86_64
CentOS Stream 10	6.12.0	x86_64
Fedora 31	5.8.18	x86_64
Fedora 36	6.2.15	x86_64
Fedora 41	6.13.5	x86_64
Fedora 42	6.15.5	x86_64
OpenSuse 15.6	6.4.0	amd64
Ubuntu 16.04 LTS	4.15.0	amd64
Ubuntu 18.04 LTS	5.4.0	amd64
Ubuntu 20.04 LTS	5.15.0	amd64
Ubuntu 22.04 LTS	6.8.0	amd64
Ubuntu 24.04 LTS	6.11.0	amd64
Ubuntu 25.04	6.14.0	amd64

sysbas_mpdv.v23 is compatible with kernel versions 4.3 and below.

sysbas_mpdv.v21 is compatible with kernel versions 2.4 and below.

3. Release Note

- **Linux Driver Version 24.0**

- (Update) Support to kernel version 5.4.0~6.15.0 (Ubuntu, Fedora, CentOS, suseLinux)
- Not Support to kernel version 4.3.xx

- **Linux Driver Version 23.0**

- (Update) Support to kernel version 5.0.0~5.3.0 (Ubuntu, Fedora)

- **Linux Driver Version 22.0**

- (Update) Support to kernel version 4.4~4.16 (Ubuntu)
- Not Support to kernel version 2.4.xx
- Not Support to less than kernel version 3.10.0-862 (CentOS)

- **Linux Driver Version 21.0**

- (Update) Support to kernel version 3.13~4.2

- **Linux Driver Version 20.2**

- (Update) Support to Multi-32/PCIe (bridge type)

- **Linux Driver Version 20.1**

- (Bug Fix) Fix of "struct tty_port" link
- (Update) Support to kernel version 3.12

- **Linux Driver Version 20**

- (Update) Support to kernel version 3.8

4. Required for driver installation

1. Login as root user (Superuser).

You must login as root. You can check it from the prompt as shown below.

(Default: If you login without the root account, it will show \$, but when you did, # is displayed.)

2. GCC (GNU C Compiler)

Check if you have the GCC installed. When you type 'gcc -v' and the result shows "gcc: command not found", the GCC is not installed. You must install it before installing the driver.

```
sysbas@utu:/tmp$ gcc -v
Using built-in specs.
COLLECT_GCC=gcc
COLLECT_LTO_WRAPPER=/usr/lib/gcc/i686-linux-gnu/4.6.1/lto-wrapper
Target: i686-linux-gnu
Configured with: ../src/configure -v --with-pkgversion='Ubuntu/Linaro 4.6.1-9ubuntu3' --with-bugurl=file:///usr/share/doc/gcc-4.6/README.Bugs --enable-languages=c,c++,fortran,objc,obj-c++,go --prefix=/usr --program-suffix=-4.6 --enable-shared --enable-linker-build-id --with-system-zlib --libexecdir=/usr/lib --without-included-gettext --enable-threads=posix --with-gxx-include-dir=/usr/include/c++/4.6 --libdir=/usr/lib --enable-nls --with-sysroot=/ --enable-clocale=gnu --enable-libstdcxx-debug --enable-libstdcxx-time=yes --enable-plugin --enable-objc-gc --enable-targets=all --disable-werror --with-arch-32=i686 --with-tune=generic --enable-checking=release --build=i686-linux-gnu --host=i686-linux-gnu --target=i686-linux-gnu
Thread model: posix
gcc version 4.6.1 (Ubuntu/Linaro 4.6.1-9ubuntu3)
```

<GCC installed>

```
sysbas@utu:/tmp$ gcc -v
gcc: command not found
```

<GCC not installed>

Ubuntu 22.04 ships with GCC 11 by default, whereas the target GCC version for the build is 12. This version difference may cause a compiler mismatch.

Reinstall GCC version 12

3. Kernel Source

Check whether the kernel source files are installed. Type 'cd /usr/src' and check the result.

If you can't find kernel source under the directory, you must install the kernel source files before installing the driver.

```
sysbas@utu:/tmp$ cd /usr/src
sysbas@utu:/src$ ls
linux-headers-3.0.0-12          linux-headers-3.0.0-12-generic
```

<Kernel source installed>

```
sysbas@utu:/tmp$ cd /usr/src
sysbas@utu:/src$ ls
```

<Kernel source not installed>

4. Make

Check whether the "make" is installed. Type 'make -v' and check the result. When it shows "make: command not found", the "make" is not installed. You must install make. You must install the "make" before installing the driver.

```
sysbas@utu:/tmp$ make -v
GNU Make 3.81
Copyright (C) 2006 Free Software Foundation, Inc.
This is free software; see the source for copying conditions.
There is NO warranty; not even for MERCHANTABILITY or FITNESS FOR A
PARTICULAR PURPOSE.

This program built for i686-pc-linux-gnu
```

<Make installed>

```
sysbas@utu:/tmp$ make -v
make: command not found
```

<Make not installed>

5. Installation

1. Please check whether the power is off from your PC.
2. Install a PCI/PCIe Multi-port card in the PCI/PCIe slot of the PC.
3. If you have any external cables for each port, please connect the cables to the card.
4. Turn on the PC.
5. After the Linux boots, login with the administrator ID as root.
6. Run the device driver file named “sysbas_mpdv.v24.0.sh”.

The device driver file is an executable file. You could just type the name in shell followed by “.”. After you execute the file, you can see a sub directory named “sysbas_mpdv.v24.0” with installation information like a below image.

You will see the followings: model name of the multiport, type of serial interface-RS232/RS422/RS485, port name and version in the installation information

```
root@utu:/tmp# ./sysbas_mpdv.v24.0.sh
Verifying archive integrity... All good.
Uncompressing Enhanced Async Multi-Port(PCI/PCIe) Linux device driver installer.....
=====
          Enhanced Async Multi-Port(PCI/PCIe) Linux Device Driver
          Version : 24.0      revision: 2025-05-26
=====
1 board(s) installed
Board No.1 : Multi-4 PCI (rev b0)
      /dev/ttyMP0 (RS232 , 16C105X)
      /dev/ttyMP1 (RS232 , 16C105X)
      /dev/ttyMP2 (RS232 , 16C105X)
      /dev/ttyMP3 (RS232 , 16C105X)
root@utu:/tmp#
```


6. Remove

1. Go to the directory where the driver is installed.

If you installed the driver at “~/tmp”, please go to “~/tmp/sysbas_mpdv.v24.0” directory.

2. Run **Remove** command as follows.

```
#./Remove
```

```
root@utu:/tmp/sysbas_mpdv.v24.0# ls
Install  Remove  async_multiport  ioctl  multidrop_test
root@utu:/tmp/sysbas_mpdv.v24.0# ./Remove
```

3. All the installed files will be removed automatically after executing **Remove** command.

```
Remove Multiports PCI/PCle Driver..!!
remove device(/dev)....done
modify rc.local....done
root@utu:/tmp/sysbas_mpdv.v24.0#
```

7. Testing

```
#cd async_multiport
```

```
#./sb_test [Port Name] [Baudrate] [TestMode]
```

If you want to know how to use the sb_test, you just type the name without any argument. And then you can see the method of the usage.

```
root@utu:/tmp/sysbas_mpdv.v24.0/async_multiport# ./sb_test
Usage: ./sb_test [Port Name] [Baudrate] [TestMode]
Port Name : /dev/ttyMP0 ~ /dev/ttyMP32
Baudrate  : 9600, 19200, ...
TestMode  : 0(Loopback) 1(Send) 2(Recv)
root@utu:/tmp/sysbas_mpdv.v24.0/async_multiport#
```

```
Usage: ./sb_test [Port Name] [Baudrate] [TestMode]
```

```

Port Name      : /dev/ttyMP0 ~ /dev/ttyMP32
Baudrate       : 9600, 19200, 38400, 57600, 115200, 230400, 460800,
921600
TestMode       : 0 (Loopback)
                  1 (Send)
                  2 (Receive)
```

Example :

```

$./sb_test /dev/ttyMP0 9600 0
$./sb_test /dev/ttyMP5 921600 0
$./sb_test /dev/ttyMP3 115200 1
```

(Note) There is a blank space after “/sb_test” and before “/dev”.

After you connect the loopback connector to the port you wish to test, you can test it by using the loopback mode.

The test pattern is “abcdefghijklmnopqrstuvwxyz” and the program generates characters from “a” to “z” repeatedly increasing by one character.

```

root@utu:/tmp/sysbas_mpdv.v24.0/async_multiport# ./sb_test /dev/ttyMP0 9600 0
Loopback Test Mode  !
a
ab
abc
abcd
abcde
abcdef
abcdefg
abcdefgh
abcdefghi
abcdefghij
abcdefghijk
abcdefghijkl
abcdefghijklm
```

When the multi-port card driver is installed correctly, you will see the test result repeatedly as shown above.