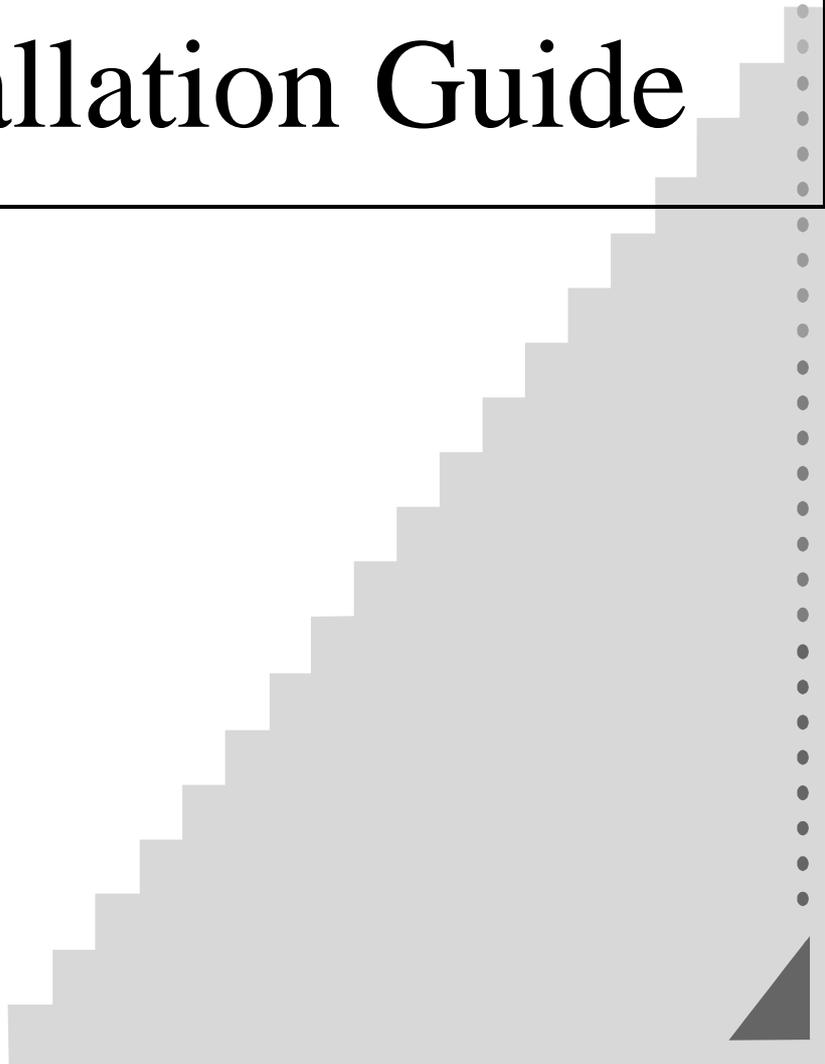


PCI 32 BIT Multi I/O Cards Linux Kernel 2.4 Installation Guide

January 6, 2003



INDEX

- ◆ INSTALLATION GUIDE FOR PCI SERIAL PORTS.....3
- ◆ INSTALLATION GUIDE FOR PCI PARALLEL PORTS8

◆ *Installation Guide for PCI Serial Ports*

This installation guide describes the procedures to install SUNIX PCI serial ports in Linux platform.

[Linux Platform]

- Operating System : RedHat V6.1/V6.0 (Kernel 2.2.x / 2.0.x)
- Terminal Emulation AP : minicom / xminicom
- Internet Dialer : Kppp

[Installation Steps]

(1) Find the available serial ports

Since Linux only support 4 serial ports (ttyS0, ttyS1, ttyS2, ttyS3) under the default condition. Most likely, ttyS0 & ttyS1 are supported by mother board's built-in 16550 controllers and ttyS2 & ttyS3 are free for additional I/O card. (Note that ttyS2: S is upper case)

It could be checked by the following commands.

```
#setserial /dev/ttyS0 -a      (COM1)
#setserial /dev/ttyS1 -a      (COM2)
#setserial /dev/ttyS2 -a      (COM3)
#setserial /dev/ttyS3 -a      (COM4)
```

If COM1 is used by mouse, the response is similar to

```
/dev/ttyS0 : Device or resource busy
```

If the COM1 does not attach any device, the response is similar to

```
/dev/ttyS0, Line 0, UART: 16550A, Port: 0x3f8, irq: 4
      Baud_base: 115200, clos_delay: 50, divisor: 0
      closing_wait: 3000, closing_wait2: infinite
      Flags: spd_normal skip_test
```

In case ttyS2 (COM3) is free, the response for command `# setserial /dev/ttyS2 -a` is shown below.

```
/dev/ttyS2, Line 2, UART: unknown, Port: 0x3e8, irq: 4  
Baud_base: 115200, clos_delay: 50, divisor: 0  
closing_wait: 3000, closing_wait2: infinite  
Flags: spd_normal skip_test  
(note that UART: unknown)
```

In case ttyS3 (COM4) is free, the response for command `# setserial /dev/ttyS3 -a` is shown below.

```
/dev/ttyS3, Line 3, UART: unknown, Port: 0x2e8, irq: 3  
Baud_base: 115200, clos_delay: 50, divisor: 0  
closing_wait: 3000, closing_wait2: infinite  
Flags: spd_normal skip_test  
(note that UART: unknown)
```

Finally, the `/dev/ttyS2` & `/dev/ttyS3` are free for PCI serial ports.

(2) Find the PCI card resource (IO port address & IRQ) for the serial ports

Please enter the command `"#more /proc/pci"`.

The response is similar to the following

```
.....  
.....  
Bus 0, Device 11, function 0:  
  ^^  
Serial controller : Unknown vendor Unknown device (rev 1).  
Vendor id=1409, Device id=7168  
Medium devsel. Fast back-to-back capable. IRQ 10  
  ^^  
I/O at 0xef80 [0xef81]  
  ^^^^  ^^^^  
.....
```

(note : ^^ means it could be different from the above.
They are varied with the different PC.)

From the /proc/pci file, it is possible to find the PCI card's IO port address and IRQ. Especially, the SUNIX card always shows

"Vendor id=**1409**, Device id=**7168**".

(3) Configure the parameters for ttyS2 & ttyS3

for SUNIX 4025A card (PCI 1S), please enter (if ttyS2 is free)

```
# setserial /dev/ttyS2 port 0xef80 UART 16550A
      irq 10 Baud_base 921600
```

for SUNIX 4036A card (PCI 2S), please enter (if ttyS2 & ttyS3 are free)

```
# setserial /dev/ttyS2 port 0xef80 UART 16550A
      irq 10 Baud_base 921600
# setserial /dev/ttyS3 port 0xef88 UART 16550A
      irq 10 Baud_base 921600
```

(4) Check the setting for ttyS2 & ttyS3

Please enter `# setserial /dev/ttyS2 -a`

The response look likes below

```
/dev/ttyS2, Line 2, UART: 16550A, Port: 0xef80, irq: 10
      Baud_base: 921600, clos_delay: 50, divisor: 0
      closing_wait: 3000, closing_wait2: infinite
      Flags: spd_normal skip_test
```

(5) Then the ttyS2 & ttyS3 are ready for application

(eg. `minicom -s` or `xminicom -s` or `Kppp ...`)

(6) In case more than 4 serial ports are needed

If there are more than 4 serial ports to be supported by Linux system, (e.g. Sunix 4055A/4056A/4065A/4066A/4095A/4096A cards) the first step is to add more tty device nodes into system.

Inquire the system tty device nodes,

`#ls -al /dev/ttyS*`

```
crw----- 1 root tty 4, 64 Jan 8 11:40 /dev/ttyS0
crw----- 1 root tty 4, 65 Jan 8 11:40 /dev/ttyS1
crw----- 1 root tty 4, 66 Jan 8 11:40 /dev/ttyS2
crw----- 1 root tty 4, 67 Jan 8 11:40 /dev/ttyS3
```

Add tty device node one by one

```
#mknod /dev/ttyS4 c 4 68 (for ttyS4)
#mknod /dev/ttyS5 c 4 69 (for ttyS5)
#mknod /dev/ttyS6 c 4 70 (for ttyS6)
#mknod /dev/ttyS7 c 4 71 (for ttyS7)
```

.....

Please add all tty device nodes accordingly

Configure the parameters for all new ttyS*

Please repeat step (2) (3) (4) to inquire and change the I/O address for each tty device. Because all the new added tty device nodes are still invalid by default.

For PCI 4S card (4055A/4056A), it allocate **2** I/O resources. You could inquire it according to step(2). E.g.

`# more /proc/pci`

.....

Vendor id=1409, Device id=7168
Medium devsel. Fast back-to-back capable. IRQ 10

I/O at 0xd000 [0xd001] → 1st port=0xd000, 2nd port=0xd008
I/O at 0xb800 [0xb801] → 3rd port=0xb800, 4th port=0xb808

For PCI 8S card (4065A/4066A), it allocate **6** I/O resources. You also need to inquire its allocated address

`# more /proc/pci`

.....

Vendor id=1409, Device id=7168
Medium devsel. Fast back-to-back capable. IRQ 10

^^

- I/O at 0xd000 [0xd001] → 1st port=0xd000, 2nd port=0xd008
- I/O at 0xb800 [0xb801] → 3rd port=0xb800, 4th port=0xb808
- I/O at 0xb400 [0xb401] → 5th port=0xb400
- I/O at 0xb000 [0xb001] → 6th port=0xb000
- I/O at 0xa800 [0xa801] → 7th port=0xa800
- I/O at 0xa400 [0xa401] → 8th port=0xa400

Re-Inquire the system tty device nodes,

#ls -al /dev/ttyS*

```
crw----- 1 root tty 4, 64 Jan 8 11:40 /dev/ttyS0
crw----- 1 root tty 4, 65 Jan 8 11:40 /dev/ttyS1
crw----- 1 root tty 4, 66 Jan 8 11:40 /dev/ttyS2
crw----- 1 root tty 4, 67 Jan 8 11:40 /dev/ttyS3
crw-r--r-- 1 root root 4, 68 Jan 18 11:40 /dev/ttyS4
crw-r--r-- 1 root root 4, 69 Jan 18 11:40 /dev/ttyS5
crw-r--r-- 1 root root 4, 70 Jan 18 11:40 /dev/ttyS6
crw-r--r-- 1 root root 4, 71 Jan 18 11:40 /dev/ttyS7
```

.....

Important Notes :

(1) Since all serial ports on Sunix PCI card are using only one interrupt pin, you must set them the same IRQ number with **setserial** command.

(2) Un-installation,
e.g.#rm /dev/ttyS4 (remove ttyS4 device)

◆ *Installation Guide for PCI Parallel Ports*

This installation guide describes the procedure to install SUNIX PCI parallel ports in Linux platform.

[Linux Platform]

- Operation System: RedHat V6.1/V6.0 (Kernel 2.2.x / 2.0.x)
- "AnotherLevel menu -> administration -> printtool" in X windows.

[Installation steps]

Linux kernel provide a 'parport' code to support parallel port (/dev/lp0, /dev/lp1, /dev/lp2). This code provides the ability to share one port between multiple devices. And it is loadable when kernel is running.

The 'parport' code is split into two parts : generic (which deals with port sharing) and architecture-dependent eg. X86, SPARC. (which deals with actually using the port). Thus, please take the following steps to install 'parport' module!

(1) Check the on-board parallel port's hardware resources

Most likely each motherboard has one built-in parallel port. And its hardware resources are settable with BIOS utility.

(for example, on-board parallel is in 0x378 port, IRQ=7)

(2) Find the PCI resource (IO port address & IRQ) for the parallel ports

please enter the command "# more /proc/pci"

The response will look like the following

```
.....  
.....  
Bus 0, Device 11, function 0:  
  ^^
```

**Parallel controller : Unknown vendor Unknown device (rev 1).
Vendor id=1409, Device id=7268 (or 7168)
Medium devsel. Fast back-to-back capable. IRQ 10**

^^

**I/O at 0xe000 [0xe001]
I/O at 0xd800 [0xd801]
I/O at 0xd400 [0xd401]
I/O at 0xd000 [0xd001]**

^^^^ ^^^^^

.....
.....

(Note : ^^ means it could be different from the above.
They are varied with the different PC.)

From the /proc/pci file, it is possible to find the PCI card's I/O port address and IRQ. Especially, the SUNIX card always shows

"Vendor id=1409, Device id=7268"
or
"Vendor id=1409, Device id=7168"

(3) Install 'parport' module

<step I> to load the generic 'parport' code

insmod parport.o

<step II> to load the architecture-dependent code to tell 'parport' code

**# insmod parport_pc.o io=0x378,0xe000,0xd400
 irq=7,none,none**

that you have three PC-style ports, one at 0x378 with IRQ 7, one at 0xe000 with no IRQ, one at 0xd400 with no IRQ.

Note :

1. Please enter "**#find /lib -name parport.o**" to find out the correct

- file path. (it is varied with the different kernel version)
2. For the detailed description, please refer to /Documentation /parport.txt in Linux kernel

(4) Check the attached printer device [optional]

once the architecture-dependent part of the parport code is loaded into the kernel. You could enter the following command

```
# insmod parport_proble.o
```

to check any attached devices and log a message similar to

```
parport0: Printer, BJC-210 (Canon)
```

(5) Then /dev/lp0, /dev/lp1 and /dev/lp2 are ready for service

Now parallel port is available, please connect a correct type of printer for your printing applications.

(eg. AnotherLevel menu -> administration -> printtool in X windows)

Note :

1. Type "**# cat /dev/printcap**" to inquire the printer connection.
2. It's possible to print a text file to the printer for verification
(**#lpr -Pprinter_name textfile_name**)

(6) In case more than 3 parallel ports (eg. /dev/lp3, /dev/lp4 ...) are needed for service

Please get the root privilege and enter the following command

```
#mknod /dev/lp3 6 3  
#chmod 660 /dev/lp3  
#chgrp daemon /dev/lp3  
→to add /dev/lp3 into kernel
```

```
#mknod /dev/lp4 6 4
#chmod 660 /dev/lp4
#chgrp daemon /dev/lp4
→to add /dev/lp4 into kernel
```

```
.....
.....
```

then please enter the following command to check /dev/lp devices

```
#ls -al /dev/lp*
crwxrwxrwx 1 root daemon 6, 0 may 5 1998 /dev/lp0
crwxrwxrwx 1 root daemon 6, 1 may 5 1998 /dev/lp1
crwxrwxrwx 1 root daemon 6, 2 may 5 1998 /dev/lp2
crwxrwxrwx 1 root daemon 6, 3 dec 9 1999 /dev/lp3
crwxrwxrwx 1 root daemon 6, 4 dec 9 1999 /dev/lp4
```

and then repeat step (3) to load parport module for /dev/lp*.

Note : #rm /dev/lp3 to remove it