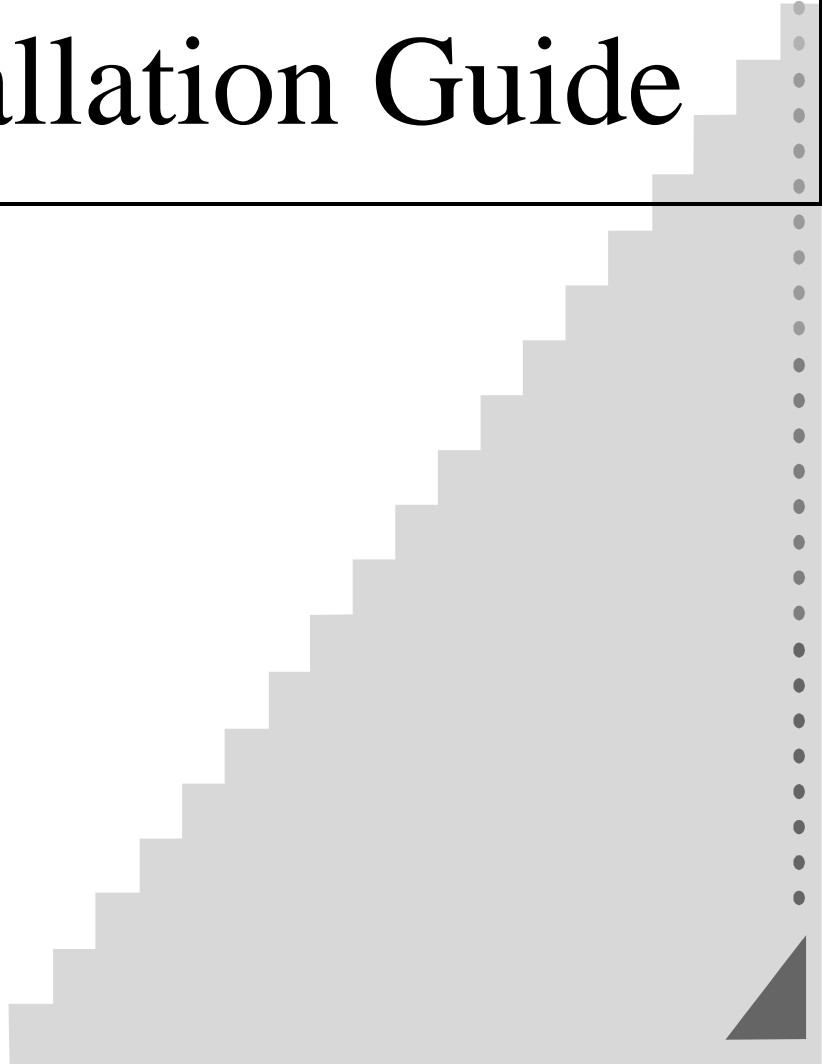


PCI 32 BIT
Multi I/O Series Card
Linux
Installation Guide



INDEX

INSTALLATION GUIDE FOR PCI SERIAL PORTS.....	3
INSTALLATION GUIDE FOR PCI PARALLEL PORTS	6

◆ *Installation Guide for PCI Serial Ports*

This installation guide describes how to install PCI serial ports in Linux platform.

[Linux Platform]

- Operating System : RedHat V6.0/V5.2 (Kernel 2.2.5 / 2.0.36)
- Terminal Emulation AP : minicom
- 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 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 mouse uses the COM1, 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: 3
      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** is shown below.

```
/dev/ttyS2, Line 2, UART: unknown, Port: 0x3f8, 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** is shown below.

```
/dev/ttyS3, Line 3, UART: unknown, Port: 0x3f8, 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)
```

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 with above.
It is case by case on the different PC.)

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

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

(3) Configure the parameters for ttyS2 & ttyS3

for 1 serial port card (PCI 1S), please enter (if ttyS2 is free)

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

for 2 serial ports 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 Linux's 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 or Kppp ...)

◆ *Installation Guide for PCI Parallel Ports*

This installation guide describes how to install PCI parallel ports in Linux platform.

[Linux Platform]

- Operation System: RedHat V6.0/V5.2 (Kernel 2.2.5 / 2.0.36)
- 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 a 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 with above.
It is case by case on 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 PCI parallel card always shows

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

(3) Install 'parport' module

to load the generic 'parport' code

insmod parport.o

to load the architecture-dependent code to tell 'parport' code

**# insmod parport_pc.o io=0x378,0xe000,0xd401
 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.

(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

```
# insmode 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

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