How to install driver in Linux kernel 2.6 system

1. To get driver

a) Generally we can get Linux driver in CD or new driver from rayon@ms1.hinet.net service center.

b) It may be compressed file (ex, **alnx813.Z** or **alnx916.Z**). So you need to copy such file to your Linux system and uncompress it in **root directory** to get original driver diskette.

```
#cd /
#tar xvfz alnx813.Z          (we suggest that you copy file here)
```

c) Now we can have driver files in **/etc/rayon** or **/etc/rayone** directory. So we need to install driver in such location.

```
#cd /etc/rayon
```

NOTE: we have alnx813.Z driver for IOP3927, PCIPORT, P640NU cards and driver files in /etc/rayon directory.

NOTE: we have alnx916.Z driver for APCIe, OPe95x, PCIePort, SPCIPORT cards and driver files in /etc/rayone directory.

2. To install driver

a) Because there are different distribution version for Linux kernel 2.6 system. They may have different file to be run in boot procedure. So we need to use different Install script for different distribution Linux system.

b) Currently we have "**Install.redhat**" for REDHAT type Linux system. We have "**Install.susu1**" for Suse type Linux system. We have "**Install.slackware**" for Slackware type Linux system. We have "**Install.Ubuntu**" for Ubuntu type Linux system.

c) Please use one install file to meet your target system to install Linux driver. In following procedure we will install driver in **Fedora 6** system for **P588U** card.

d) Because Fedora 6 system is **REHAT** type system, so we will use "**Install.redhat**" to install driver.

#./Install.redhat

e) Because P588U card is **PCIPORT** type, so we will choose PCIPORT driver to install.

f) Because Fedora 6 system is **kernel 2.6.18**, so we will choose kernel version **before 2.6.20** .

g) After the driver installation procedure we can have **pciport.ko** file to use. In next boot procedure we can have extra TTY device (serial port) to use.

3. How to confirm driver installed

a) In boot procedure you can see the **driver's installed message** in console. If you did not see such console out message in your screen, then you can use "**dmesg**" command to check console out message for boot procedure.

#dmesg > ttt

Now you can check "ttt" file to confirm our driver's installed message.

b) Generally we may have three conditions to happen.

1) You do not have "**pciport.ko**" file generated in driver installation procedure. Or you use the wrong "Install" file(For Fedora system you do not use Install.redhat and use wrong Install.susel file) to install driver. So we do not install MODULE driver in boot procedure. ===> In this condition we may not see any message about "rayon" in boot procedure.

2) You do not install our driver in **same kernel version** (your original Fedoar 6 system) as your boot kernel version (you had upgraded your kernel). ====> You can see message about "rayon" and tell you MODULE version is wrong.

3) You may choose **wrong card model** to install driver. ====> You can see message about "rayon" and driver information. But you may see "**CARD NOT FOUND**" information.

c) In normal condition you may see message about "**rayon**" and driver information for **PCIPORT** card (you have **P588U** card installed).

d) Now we can run data transmission function in TTY device.

1) We have one "**etty**" utility to send or receive data via TTY device.

2) In **/etc/rayon/etty** directory we can run "make" command to generate "etty" execution file for your Linux system.

3) We can connect one **NULL MODEM cable** between two serial ports.

4) If we had NULL MODEM cable to connect between port A and port B of P588U card, we may have **/dev/tty81a** and **/dev/tty81b** device name to use.

5) We may use "etty" command to receive data in port A.

```
#./etty -rl 9600 64K 1 tty81a
```

6) In this command line we will use 9600bps to receive 64K byte data in serial port A.

7) We may use "etty" command to transmit data in port B.

```
#./etty -t 9600 64K 1 tty81b
```

8) From above procedure we can check data transmission function in P588U card.

e) Of course you can use "**minicom**" terminal emulation program to send/receive data in /dev/tyy81a--/dev/tty81h serial port.

NOTE: We can have **alnx813.Z** driver to support **IOP3927, PCIPORT, P640NU** card in **/etc/rayon** directory. We can have following device name for serial port in different card type.

a) **IOP3927 card: /dev/tty[6-7][1-8][a-h].**
b) **PCIPORT card: /dev/tty8[1-4][a-h].**
c) **P640NU card: /dev/tty[9-A][1-8][a-h].**

Each Linux system can install **one card type** only.

NOTE: We can have **alnx916.Z** driver to support **APCIe, OPe95x** card in **/etc/rayone** directory. We can have following device name for serial port in different card type.

a) **APCIe card: /dev/tty2[1-4][a-h].**
b) **OPe95x card: /dev/tty3[1-4][a-h].**

Each Linux system can install **one card type** only.

NOTE: We can install **alnx813.Z** driver and **alnx916.Z** driver (card type) in same Linux system. So we have P588U card (**PCI card slot**) and A168 card (**PCIe card slot**) in same PC.

4. Known problem and solution

a) We can not generate target ***.ko** file in driver installation procedure.

1) Firstly we need to confirm that your Linux system must support development environment. For example, you do not have C compiler in your system. Then you can not go "**build**" procedure.

2) Please confirm that your LINUX system do have "**build**" environment. Generally you can use "**uname -r**" to know your kernel version. Then we must find corresponding directory under **/lib/modules/** to have "build" environment. For example, we may have kernel version 2.6.30.5, then we may have **/lib/modules/2.6.30.5/build** exist. Such file may redirect to /usr/src/linux-2.6.30.5 location.

3) In such directory we may have "**Makefile**" to use. So we can use such environment to generate driver **.ko** file from source file.

4) If we could have correct development environment, then we may need to update our driver to meet your target kernel version. Please send mail to **rayon@ms1.hinet.net** for support.

b) We do have ***.ko** file and find our card and driver information in boot procedure. But we can not use "etty" to open our /dev/tty81a device.

1) Generally we may generate /dev/tty81[a--h] device name in driver installation procedure.

2) But your LINUX system may delete such device name in next boot. So application software can not know such device name.

3) For such system we may need to create such device name in boot procedure. Please confirm that you use the correct Install file to install driver in your system.

c) We do have *.ko file. But we can not see message for rayon driver in boot procedure.

1) Generally Linux system will run some file under /etc/rc.d directory in boot procedure. Our "Install" file will use different file name in different Linux system to install our module driver.

2) Your system may not run our target file (we add our module driver install command here) in boot procedure.

3) Please check your /etc/rc.d directory to find one file must be run in boot procedure. Please modify "Install.redhat" to have target name to match your system. Then you can have dedicated "Install" file for your system. Then we can confirm the module driver must be installed in boot procedure.