USER MANUAL

V.35 PCI Extension Cards

TAHOE 931/932

TAHOE

FREEDOM OF COMMUNICATION

TABLE OF CONTENTS

Tahoe® 931 (one V.35 port)
Tahoe® 932 (two V.35 ports)
User Manual
http://www.tahoe-group.com/

Tahoe® assumes no responsibility for any errors or omissions that may appear in this document. Tahoe® makes no commitment to update the information contained here, and may make changes at any time without notice.

1. Introduction

The intelligent Tahoe® 930 series PCI interface cards are equipped with one or two V.35 ports capable of transmitting data at speeds up to 5 Mbps. The V.35 devices (e.g. the HDSL modems) can be connected directly to a PC without any expensive external routers.

The built-in hardware HDLC controller fully services the data transmission, thus the PC processor isn't unnecessarily loaded.

Both Linux and FreeBSD drivers are available, supporting Frame Relay, HDLC, Cisco® HDLC, X.25 and synchronous PPP protocols.

2. Card installation

Following steps should be taken to install the card:

- disconnect the power cable from the power supply
- remove the case
- find an unoccupied PCI slot
- unscrew the masking plate in the rear of the free PCI slot
- insert the card into the slot (the card connectors should be accessible through the rear opening in the case)
- screw on the card firmly to the case
- close the case and switch the power on

After installing the card inside the PC use the V.35 cable to connect an external device (e.g. a V.35 modem) to the card.

3

3. Drivers installation

Tahoe® WAN PCI cards are supplied with Linux and FreeBSD drivers. The Linux drivers consist of two parts:

- a patch for the Linux kernel
- a sethdlc utility used to configure the card

The FreeBSD drivers consist of:

- a kernel module
- a sethdlc utility

The CD-ROM enclosed contains separate directories for each of the operating systems.

3.1

3.1. Linux drivers

The “linux” directory contains:

- `hdlc-2.4.24-1.15t.patch` - a kernel patch
- `sethdlc-1.15t.tar.gz` - sethdlc source code
- `linux-2.4.24.tar.gz` - original Linux kernel source code
- `linux-2.4.24-tahoe.tar.gz` - Linux kernel source code with driver patch applied

3.1.1

3.1.1. Linux kernel compilation

Follow these steps to compile the Linux kernel with drivers included:

- copy the kernel sourced from the CD-ROM (`linux-2.4.24-tahoe.tar.gz` - with patch applied)

- uncompress it in the /usr/src directory:

```
root@linux ~ # cd /usr/src
root@linux /usr/src # tar -zxvf linux-2.4.24.tar.gz
root@linux /usr/src # ln -sf linux linux-2.4.24
```

- configure kernel:

```
root@linux /usr/src # cd linux
root@linux /usr/src/linux # make menuconfig
```

Enter the "Network device support" menu, then choose the "WAN interfaces" submenu and mark the "WAN interfaces support" option. A list of options will appear, on which you should select "Generic HDLC layer", appropriate protocols (like "Frame Relay support") and Tahoe card drivers ("Tahoe 9xx support"). You can insert them permanently into the kernel ("<*>") or compile as a module ("<M>").

○ besides Tahoe drivers you may need to enable other options like the Ethernet card support, SCSI drivers, multimedia devices, etc.

○ after configuring a kernel it should compiled:

```
root@linux /usr/src/linux # make dep
root@linux /usr/src/linux # make clean
root@linux /usr/src/linux # make bzImage
```

A new kernel should be now installed.

3.1.2

3.1.2. New Linux kernel installation

After the compilation, the kernel is placed in a file named /usr/src/linux/arch/i386/boot/bzImage. Now you should copy it to the /boot directory changing its name to a more meaningful one:

```
root@linux /usr/src/linux # cp arch/i386/boot/bzImage
/boot/vmlinuz-2.4.24-tahoe
```

Then you have to install it. The most widely used kernel loader is LILO (Linux Loader). In its configuration file - /etc/lilo.conf - at the end you should add new section:

```
image = /boot/vmlinuz-2.4.24-tahoe
        label = Tahoe
        root = /dev/hda1
        read-only
```

Instead of /dev/hda1 you may need to enter the name of the partition where the Linux root filesystem resides.

To install LILO with new settings simply type 'lilo'. Then you can force loading of a new kernel at the next system boot:

```
root@linux ~ # lilo
root@linux ~ # lilo -R Tahoe
```

After entering these commands you should restart your system - it will boot with the new kernel. If everything works fine, you can add a line ‘default = Tahoe' to the /etc/lilo.conf file and type 'lilo', so the kernel labeled 'Tahoe' will be the default one used at the system boot.

If for some reason you cannot use this kernel (e.g. drivers for other system components are missing), you may simply reboot your system and the old kernel will be loaded again.

If the drivers were compiled as a module, then after booting of the new kernel you should load them:

```
root@linux ~ # modprobe tahoe9xx
```

In the /var/log/messages file you should see messages confirming successful loading of the drivers and detecting a card:

```
kernel: Tahoe 9xx driver version: 1.15t
kernel: Tahoe932: 256 KB RAM at 0xd9000000, IRQ5,
using 10 TX + 75 RX packets rings
kernel: hdlc0: Tahoe932 node 0
kernel: hdlc1: Tahoe932 node 1
```

If the modules were permanently compiled into the kernel, then such messages will appear during the system boot. You can review them using the **dmesg** command.

Since now there are new interfaces available - hdlc0 and hdlc1 (or hdlc0 alone in case of one-port cards).

The next step is to configure the card using the **sethdlc** utility (described after the FreeBSD installation section)

3.2

3.2. FreeBSD drivers

The "freebsd" directory contains:

- `if_th.ko` - a kernel module
- `sethdlc-1.15t.tar.gz` - sethdlc source code

Copy the drivers from the CD-ROM and unpack them using following command:

```
tar -zxvlf tahoe9xx-freebsd5.1.tar.gz
```

(instead 5.1 there may be 4.8 in case of FreeBSD 4.8 drivers)

Directory tahoe9xx-freebsd5.1 will be created, containing kernel module and source code of sethdlc utility. To compile and install drivers enter:

```
make install
```

Driver module **if_th.ko** will be installed in modules directory - **/boot/kernel** in case of FreeBSD 5.1 or **/modules** in case of FreeBSD

4.8. Now it can be loaded using **kldload if_th** command or by bringing the interface up: **ifconfig th0 up**. One interface - th0 - is added when using one-port card and two interfaces - th0 and th1 - when using two-port card.

Now the card sould be configured using the **sethdlc** utility.

3.3. Using sethdlc

The **sethdlc** utility is used to configure the WAN card. In case of FreeBSD drivers it is compiled and installed along with the kernel modules. In case of Linux drivers it is supplied in a separate file and additional steps must be followed to install it:

```
root@linux /usr/src # tar -zxvf sethdlc-1.15t.tar.gz
root@linux /usr/src # cd sethdlc-1.15t
root@linux /usr/src/sethdlc-1.15t # make
root@linux /usr/src/sethdlc-1.15t # make install
```

The **sethdlc** utility may be used to select the protocol used on the WAN link. Some of the options are available in the Linux driver only. Under Linux configure the hdlc0 and hdlc1 interfaces and under FreeBSD - th0 and th1.

- **transparent HDLC** (Linux driver only)**:**

```
sethdlc hdlc0 hdlc [encoding] [parity]
```

where:

- encoding is one of: nrz, nrzi, fm-mark, fm-space, manchester
- parity: no-parity, crc16, crc16-pr0, crc16-itu, crc16-itu-pr0, crc32-itu

- **HDLC treated as an Ethernet interface** (Linux driver only)**:**

```
sethdlc hdlc0 hdlc-eth [encoding] [parity]
```

parameters same as above

- **Cisco® HDLC:**

```
sethdlc hdlc0 cisco [interval value] [timeout value]
```

where:

- interval value is the time between subsequent keepalive packets sent through the interface
- timeout value is the time, after which in case of no reception

of keepalive frames the link is treated as unusable

- **Frame Relay:**

```
sethdlc hdlc0 fr [lmi type [parameter]]
```

where:

- `type` is the LMI type: `none`, `ansi`, `ccitt`
- `parameter` is a Frame Relay protocol parameter: `dce` (Linux only), `t391 value`, `t392 value`, `n391 value`, `n392 value`, `n393 value`

Moreover in case of Frame Relay protocol you can create and delete PVCs (Permanent Virtual Channels) with given DLCI values:

```
sethdlc hdlc0 create 99
sethdlc hdlc0 delete 99
```

After creating a PVC, a new interface is created. Under Linux these interfaces have names pvc0, pvc1 and so on. Under FreeBSD the name depends on the DLCI number. In the example above a pvc99 interface would be created

- **Synchronous PPP:**

```
sethdlc hdlc0 ppp
```

4. Configuring the Frame Relay

4

After setting up a connection to the Frame Relay network, user is usually supplied with following information:

LMI signaling:	ANSI
DLCI number:	99
Remote router address:	194.204.100.129
Client router address:	194.204.100.130
WAN link subnet:	194.204.100.128
Subnet mask:	255.255.255.252

After compiling and installing the drivers described below commands should be issued to configure the Frame Relay access - set the LMI signaling and create the PVC interface.

4.1. Linux

4.1

```
sethdlc hdlc0 fr lmi ansi
sethdlc hdlc0 create 99
ifconfig hdlc0 up
```

A new **pvc0** interface would appear. It should be configured using the **ifconfig** command to set supplied IP addresses:

```
      ifconfig pvc0 194.204.100.130 netmask 255.255.255.252
pointopoint 194.204.100.129
```

4.2. FreeBSD

4.2

```
ifconfig th0 up
sethdlc th0 fr lmi ansi
sethdlc th0 create 99
```

A new **pvc99** interface would appear. It should be configured using the **ifconfig** command:

```
ifconfig pvc99 194.204.100.130 194.204.100.129
```

5. The V.35 port

The V.35 port is equipped with a DB-25 connector. A single or double V.35 cable is connected there. The V.35 port has following pinout:

Pin	Port	Name	V.35 pin
1	1	GND	B
2	1	SD(A)	P
3	1	RD(A)	R
4	1	RTS	C
5	1	CTS	D
6	2	DCD	F
7	2	GND	B
8	1	DCD	F
9	1	SCR(B)	X
10	2	SD(A)	P
11	2	SCT(B)	AA
12	1	SCT(B)	AA
13	2	RD(B)	T
14	1	SD(B)	S
15	1	SCT(A)	Y
16	1	RD(B)	T
17	1	SCR(A)	V
18	-	-	-
19	2	SCR(A)	V
20	2	SCR(B)	X
21	2	RTS	C
22	2	SD(B)	S
23	2	SCT(A)	Y
24	2	CTS	D
25	2	RD(A)	R

The “V.35 pin” is the 34-pin Winchester connector pin. The Tahoe 931 card uses the first port only, while the Tahoe 932 card uses both.

6. Technical data

- serial interface controller:
  Hitachi HD64570, 10MHz

- maximum throughput:
  5 Mbps per port

- V.35 interface
  DB-25 connector
  single or double V.35 cable terminated with 34-pin Winchester connector available separately

- PCI connector:
  32-bit, 5V, 33MHz, PCI v2.1 compliant

7. Declaration of Conformity

CE

We declare that the products Tahoe 931 and Tahoe 932 comply with the regulations of the following European Directives:

○ **73/23/EEC**	low voltage safety requirements
○ **89/336/EEC**	EMC requirements
○ **99/5/EEC**	radio & telecommunication terminal equipment requirements

The compliance of Tahoe 931 and Tahoe 932 with the requirements of the above mentioned directives is ensured by complete application of the following harmonized European Standards:

- **EN 60950:2000**
- **EN 55022:1998**
- **EN 61000-6-1:2002**
- **EN 61000-6-3:2002**

Signed: Piotr Kaczmarzyk
Position: Director

Signature: Piotr Kaczmarzyk

Date: 2 Feb 2004
Place: Wroclaw, Poland

TAHOE®
Uniwersytecka 1
50951 Wrocław, Poland
phone +48 50 100 7362
fax +48 71 344 2642
http://www.tahoe-group.com/