

USER MANUAL
G.703 PCI Extension Cards
TAHOE 971/972

TAHOE
FREEDOM OF COMMUNICATION

TABLE OF CONTENTS

1. Introduction	1
2. Card installation	2
3. Drivers installation	3
3.1. Linux drivers	3
3.2. FreeBSD drivers	5
3.3. Using sethdlc	6
4. Frame Relay setup	9
5. Technical data	10
6. Declaration of Conformity	11

Tahoe® 971 (one G.703 port)

Tahoe® 972 (two G.703 ports)

User Manual

<http://www.tahoe-group.com/>

©2004 Tahoe®. All rights reserved.

Other trademarks of other companies are used only for explanation and to
the owner's benefit, without intent to infringe.

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® 970 series PCI interface cards are equipped with one or two G.703 ports working with throughputs up to 2048 kbps. The use of E1 streams never was that easy - now the data stream may be plugged directly into the PC and the operating system treats it as a regular network interface.

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.

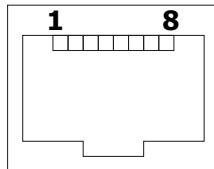
Following steps should be taken to install the card:

- 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 connect the G.703 port to the external G.703 device (e.g. a HDSL modem).

The G.703 port has a RJ-45 connector. The G.703 standard doesn't specify the connector pinout, but the pinout used is the most common in the industry, thus in most cases a regular straight patch-cord may be used to connect the card to the external device.

Anyway the pinout should be carefully checked before connecting the DCE device. The RX+ pin of the card should be connected to the TX+ pin of the other device, the RX- pin to the TX- pin, TX+ to RX+ and TX- to RX-.



Pin	Signal
1	Rx+ (input)
2	Rx- (input)
3	-
4	Tx+ (output)
5	Tx- (output)
6	-
7	-
8	-

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. New Linux kernel installation

3.1.2

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/hd1  
read-only
```

Instead of `/dev/hd1` 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: Tahoe972: 256 KB RAM at 0xd9000000, IRQ5,
using 10 TX + 75 RX packets rings
kernel: hdlc0: Tahoe972 node 0
kernel: hdlc1: Tahoe972 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

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 -zxvf 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 could be configured using the **sethdlc** utility.

3.3. Using sethdlc

3.3

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
```

3.3.1. WAN protocol choice

3.3.1

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
```

3.3.2

3.3.2. Timeslots selection

For a proper operation of the G.703 port you have to specify, which timeslots are used for transmission. Timeslots can be given in two ways - as a list of numbers (from 1 to 32) and ranges separated by commas or as a string of zeros and ones (where '0' means an unused slot and '1' - an used one, with the first slot at the beginning of the string):

- sethdlc hdlc0 slotmap 2-8,20-21,25
- sethdlc hdlc0 slotmap 011111100000000001100010000000

Use of the first timeslot automatically enables unframed mode, where whole 2048 kbps are used for data transfer. Otherwise the first timeslot is used for framing synchronization. You can also enable the unframed mode by typing:

```
sethdlc hdlc0 slotmap all
```

3.3.3. CRC4 checksum

3.3.3

The command below enables and disables, respectively, the CRC4 checksum transmission:

```
sethdlc hdlc0 crc4 on  
sethdlc hdlc0 crc4 off
```

3.3.4. Port range and sensivity selection

3.3.4

The command below switches between short haul (50m range) and long haul (2000m range) mode:

```
sethdlc hdlc0 short  
sethdlc hdlc0 long
```

4. Configuring the Frame Relay

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

4.1. Linux

```
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

4.2. FreeBSD

```
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. Technical data

- serial interface controller:
Hitachi HD64570, 10MHz
- maximum throughput:
2048 kbps per port
- G.703 interface
 - balanced, 120 , RJ-45 connector
 - G.704 framing
 - CRC4 support
 - receiver sensitivity: -12 dB or -43 dB
 - reach: 50 m or 2000 m
- PCI connector:
32-bit, 5V, 33MHz, PCI v2.1 compliant
- maximum power consumption:
Tahoe 971 630 mA @ 5V (3,15W)
Tahoe 972 700 mA @ 5V (3,50W)
- environmental conditions:

storage:	temperature	-20°C to 65°C
	humidity	5 to 95%
operation:	temperature	0°C to 40°C
	humidity	0 to 85%

6. Declaration of Conformity



TAHOE
Piotr Kaczmarzyk
ul. Uniwersytecka 1
50-951 Wroclaw, Poland

We declare that the products Tahoe 971 and Tahoe 972 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 971 and Tahoe 972 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: A handwritten signature in black ink, appearing to read "Piotr Kaczmarzyk".

Date: 30 Mar 2004
Place: Wroclaw, Poland

©2004 Tahoe®. All rights reserved.

Other trademarks of other companies are used only for explanation and to
the owner's benefit, without intent to infringe.

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.

TAHOE®
Uniwersytecka 1
50951 Wrocław, Poland
phone +48 71 344 2637
fax +48 71 344 2642
<http://www.tahoe-group.com/>