



# **Installationsanleitung und technische Daten für Asterisk Interfacekarten**

(Version: 0.2)

## **Inhaltsverzeichnis:**

- 1 Allgemeines
- 2 Installation und Konfiguration von Zapfel (Digium) Karten
  - 2.1 Holen und kompilieren der zapfel-Quellen
  - 2.2 Die Datei zaptel.conf
  - 2.3 Einbau Konfiguration der verschiedenen Karten
    - 2.3.1 X100P
    - 2.3.2 TDMxxB (TDM400P)
    - 2.3.3 TE110P (E100P,T100P)
    - 2.3.4 TE4xxP (E400P,T400P)
    - 2.3.5 IAXy (S100I)
  - 2.4 Einbinden der Kanäle in Asterisk (zapata.conf)
    - 2.4.1 Einbinden von Analogkanälen (FXS/FXO)
    - 2.4.2 Einbinden von ISDN-Kanälen
- 3 Installation und Konfiguration von beroNet Karten
  - 3.1 Einbau der BN4S0
  - 3.2 Einbau der BN8S0
  - 3.3 Anschluss des BNPW1
  - 3.4 Anschlussdose für BN8S0
  - 3.5 mISDN Installieren
    - 3.5.1 mISDN manuell installieren
    - 3.5.2 Holen und kompilieren von chan\_mISDN
    - 3.5.3 mISDN automatisch mit beroNet Makefile installieren
    - 3.5.4 Laden des Treibermoduls
  - 3.6 Die Datei misdn.conf
  - 3.7 chan\_misdn CLI Kommandos
  - 3.8 mISDN Dial-Parameter
  - 3.9 Verschlüsselung einer ISDN-Verbindung
  - 3.10 Einige Worte zum Betrieb von BNxS0 Karten mit dem „bristuff“-Treiber von Junghanns.net
- 4 Installieren von Asterisk
- 5 Weitere Informationsquellen
- 6 Beispielkonfigurationen
- 7 E1 Kabelbelegungen für den Anschluss an PRI oder andere Telefonanlagen/Switches
- 8 Technische Daten
  - 8.1 BN4S0 Board
  - 8.2 BN8S0 Board
  - 8.3 TE410P / TE405P
  - 8.4 TDMxxP / TDM400P
  - 8.5 TE110P
  - 8.6 S100i (IAXy)

## **1. Allgemeines**

Vielen Dank, dass Sie sich für den Kauf einer Digium- oder einer beroNet-Interfacekarte oder einfach nur für das Anwenden der Asterisk open source PBX entschlossen haben.

Unsere Interfacekarten sind für den Einsatz in einem „Asterisk Open source PBX“ basierten System gedacht. Die SoftwarePBX „Asterisk“ und die Treiber für die Interfacekarten unterstehen der GNU public license (GPL). Bitte haben Sie Verständnis dafür, dass für Softwareseitige Mängel o.ä. keine Gewährleistung übernommen werden kann. Lediglich für die Digium- und beroNet-Interfacekarten wird eine Garantie von einem Jahr für Gewerbekunden, und von zwei Jahren für Privatkunden übernommen.

Dieses Handbuch ist als Hilfestellung für die Installation unserer Interfacekarten verfasst worden, und erhebt nicht den Anspruch auf Vollständigkeit, und auf Fehlerlosigkeit. Sollten Ihnen nicht korrekte Aussagen oder fachliche Fehler auffallen, wären wir Ihnen dankbar wenn Sie uns das mitteilen könnten, damit wir das so schnell wie möglich beheben können.

## **2. Installation und Konfiguration von Zapitel (Digium) Karten**

### **2.1. Holen und kompilieren der zapitel-Quellen**

Damit Sie Ihre Interfacekarte/n zum funktionieren bringen, müssen Sie zuerst einen entsprechenden Treiber in Form eines Linux-Kernel-Moduls installieren. Für die Digium-Interfacekarten heißt das, man muss die entspr. Treiber-Quelldateien aus dem CVS der Firma Digium herunterladen. ***Vorausgesetzt wird hier, dass die Kernelquellen Ihres aktuell laufenden Linuxkernels sich konfiguriert auf Ihrem System befinden, und eine Internetverbindung zur Verfügung steht.*** Üblicherweise sind die Kernelquellen im Verzeichnis „/usr/src/linux“ installiert, wobei das Verzeichnis „linux“ nur ein Verweis auf das tatsächliche Quellenverzeichnis darstellt. Das Gleiche gilt für die Quellen des sogenannten 2.6'er Kernels. Da wäre es nicht „linux“ sondern „linux-2.6“.

Damit Sie zapitel und Asterisk erfolgreich kompilieren können, sind folgende Voraussetzungen auf Ihrem System zu schaffen bzw. Pakete zu installieren:

- Installierter Compiler gcc
- Kernelquellen Ihres aktuellen Kernels Installiert und konfiguriert (make menuconfig dep)
- zlib, zlib-dev
- openssl, openssl-dev
- bison 1.875
- libreadline, libreadline-dev

Um die Treiberquellen aus dem CVS zu holen, öffnen Sie eine Konsole, und gehen Sie folgendermaßen vor:

Seite / Page Seitennummer von Statistik

Gehen Sie mit „cd /usr/src“ in das Verzeichnis „/usr/src“ und geben Sie folgende Kommandozeilen ein:

**export CVROOT=:pserver:anoncvs@cvs.digium.com:/usr/cvsroot <ENTER>**

**cvs login <ENTER>**

Nun erfolgt eine Eingabeaufforderung mit "password" :  
Geben Sie folgendes ein:

**anoncvs <ENTER>**

Nach erfolgreicher Anmeldung beim CVS-Server, können Sie die Quelldateien „Auschecken“. Da Sie wahrscheinlich nicht nur die Treiberquellen brauchen, sondern später vielleicht auch die „libpri“ und die „asterisk“- Quellen, geben Sie nun folgendes ein:

**cvs co zaptel libpri asterisk <ENTER>**

Es werden in „/usr/src“ drei Verzeichnisse erzeugt:

```
„/usr/src/zaptel“  
„/usr/src/libpri“  
„/usr/src/asterisk“
```

Nachdem der Download beendet ist, wechseln Sie in das Verzeichnis „/usr/src/zaptel“.  
Geben Sie folgendes ein:

**make install <ENTER>**

Wenn alles fehlerfrei durchgelaufen ist, haben Sie nun erfolgreich alle Kerneltreibermodule in Ihrem System kompiliert und installiert.

Für die ISDN Stack Unterstützung muss noch die „libpri“ installiert werden:  
Gehen Sie in das Verzeichnis „/usr/src/libpri/“, und geben Sie folgendes ein:“

**make install <ENTER>**

Seite / Page Seitennummer von Statistik

Standardmässig müssten nun folgende Kernelmodule im Verzeichnis „lib/modules/<Kernelversion>/misc“ installiert sein: relevante Treiber sind fett hervorgehoben.

- **zaptel.o** → **Zaptel-Modul-Layer, wird für alle Zapteltreiber benötigt**
- tor2.o → Treiber für die obsoletete E400P und T400P (4xPRI)
- torisa.o → Treiber für die sehr obsoletete ISA PRI Karte
- wcusb.o → Treiber für das USB-FXS Device S100U
- **wcfxo.o** → **Treiber für die obsoletete X100P (1xAnalog FXO)**
- (- wcfxs.o)* → *obsoleter Treiber für das TDM400P Board*
- **wctdm.o** → **Neuer Treiber für das TDM400P-Board**
- ztdynamic.o → Wird benötigt für TDMoE
- ztd-eth.o → Treiber für TDMoE
- **wct1xyp.o** → **Treiber für die obsoletete E100P**
- **wct4xyp.o** → **Treiber für die TE410P und TE405P**
- **wcte11yp.o** → **Treiber für die neue TE110P (Ersatz für E100P)**
- pciradio.o →
- ztd-loc →

Bevor das entsprechende Kerneltreibermodul geladen wird, müssen noch die Kanalinformationen in der Datei „zaptel.conf“ konfiguriert werden.

## 2.2 Die Datei „zaptel.conf“

Die Datei „zaptel.conf“ befindet sich direkt im Verzeichnis „/etc“. Ganz im Gegensatz aller anderen Asterisk-Konfigurationsdateien welche sich im Verzeichnis „/etc/asterisk“ befinden. Die „zaptel.conf“ beinhaltet alle relevanten Informationen welche die Treibermodule benötigen, um die entspr. Karte und sich selbst zu konfigurieren. D.h., wenn man ein Treibermodul lädt (modprobe <modulX>), wird nach dem Laden des Moduls ein „ztcfg“ ausgeführt. Dies geschieht in der Regel automatisch. Sollte das mal nicht automatisch geschehen, kann man es nach dem Laden einfach manuell ausführen mit der Eingabe „ztcfg <ENTER>“. Bei „ztcfg“ handelt es sich um eine ausführbare Datei die beim Kompilieren von zaptel mit generiert wird. „ztcfg“ konfiguriert alle im zaptel (vom jeweiligen Treiber) angemeldeten Kanäle nach den Vorgaben in der „zaptel.conf“.

Um das entsprechende Kerneltreibermodul zu laden, geben Sie folgendes ein, hier am Beispiel für eine TE1100P:

**modprobe wcte11yp.o <ENTER>**

## 2.3 Einbau Konfiguration der verschiedenen Karten

Im folgenden Abschnitt werden nun die verschiedenen Konfigurationen der verschiedenen Zaptel-Interfacekarten in der „zaptel.conf“ vorgestellt. Hierbei ist zu beachten, dass sich der jeweilige Abschnitt auf das Vorhandensein einer einzigen Karte bezieht. Sollten Sie mehrere (verschiedene oder gleiche) Karten in Ihrem System haben, gilt folgendes:

### ***Die Kanäle werden in der Reihenfolge des Ladens der Treiber angelegt.***

D.h., wenn Sie z.B. eine TDM Karte (z.B. TDM40B) und eine TE110P in Ihrem System haben, müssen Sie die Konfiguration in der „zaptel.conf“ in der Reihenfolge schreiben, in der Sie vorhaben die Treiber zu laden. Jeder Treiber legt im virtuellen Dateisystem „/proc/zaptel“ sogenannte „spans“ an, in denen sich die Kanäle befinden. Jede Nummer in diesem Verzeichnis repräsentiert ein „span“.

Dabei ist folgendes zu beachten:

- Für eine Digium TDM Karte wird ein „span“ mit vier Kanälen (vier Module=vier Kanäle) angelegt, auch wenn nicht alle Modulplätze besetzt sind.
- Für die Digium PRI Karten wird für jeden Port ein „span“ mit 31(E1) oder 24 (T1) Kanälen angelegt

Zuletzt sollte noch festgelegt werden, mit welcher „tonezone“ Sie arbeiten wollen. Die Standardeinstellung gibt folgendes for:

**loadzone = us**  
**defaultzone=us**

„loadzone“ gibt an welche Tonsignalisierung benutzt werden soll. D.h., welcher Klingelton, Besetztton usw. Diese Tonzonen sind konfigurabel, und für viele Länder vorgegeben in der Datei „/etc/asterisk/indications.conf“. Diese Datei steht erst zur Verfügung wenn Asterisk kompiliert und installiert ist.

Wenn Sie sich in Deutschland befinden, stellen Sie die Tonzone auf „de“ oder „nl“.

Empfehlenswert ist, wenn Sie die von „make“ installierte „zaptel.conf“ leeren, und folgendes eingeben:

**loadzone=de**  
**defaultzone=nl**

Danach fügen Sie die Konfiguration Ihrer Karte, wie im Folgenden beschrieben, hinzu.

**TIPP:** Fügen Sie am Ende einer jeden Konfigurationsdatei eine Leerzeile ein.

Seite / Page Seitennummer von Statistik

Weiter geht es nun mit der Kanalkonfiguration, welche Grundsätzlich folgende Form hat:

**[device]=[Kanalliste]**

Dabei kann [Kanalliste] eine oder mehrere Nummern in den folgenden Formen repräsentieren:

- 1
- 1-4
- 1-15,17-31

### 2.3.1 X100P (*obsolete*)



Bei der X100P handelt es sich um eine 1-Port FXO Karte, die es ermöglicht einen analogen Telefonanschluss in das Asterisk System zu führen. Nach dem Einbau der Karte fügen Sie folgendes in Ihre „zaptel.conf“ ein:

**fxsks=1**

Laden Sie das Treibermodul „wcfxo.o“ mit folgender Eingabe:

**modprobe wcfxo <ENTER>**

Gegebenenfalls geben Sie danach noch „ztcfg“ <ENTER> ein.

Wenn das Module korrekt geladen wurde müsste nun folgendes in der „Datei“ / proc/zaptel/1 stehen:

```
Span 1: WCFXO/0 "Wildcard X101P Board 1"  
      1 WCFXO/0/0 FXSKS (In use)
```

### 2.3.2 TDMxxP (TDM400P)



Bei der TDM400P handelt es sich um das Trägerboard für insgesamt vier analoge Module. Folgende Kombinationen werden vorgefertigt angeboten:

**TDM01B, TDM02B, TDM03B, TDM04B, TDM10B, TDM11B, TDM12B, TDM13B,  
TDM20B, TDM20B, TDM21B, TDM22B, TDM30B, TDM31, TDM40B**

Dabei steht die erste Zahl in der Kartenbezeichnung für die Anzahl der FXS-Module und die zweite Zahl für die Anzahl der FXO-Module.

**!!!! ACHTUNG !!!**

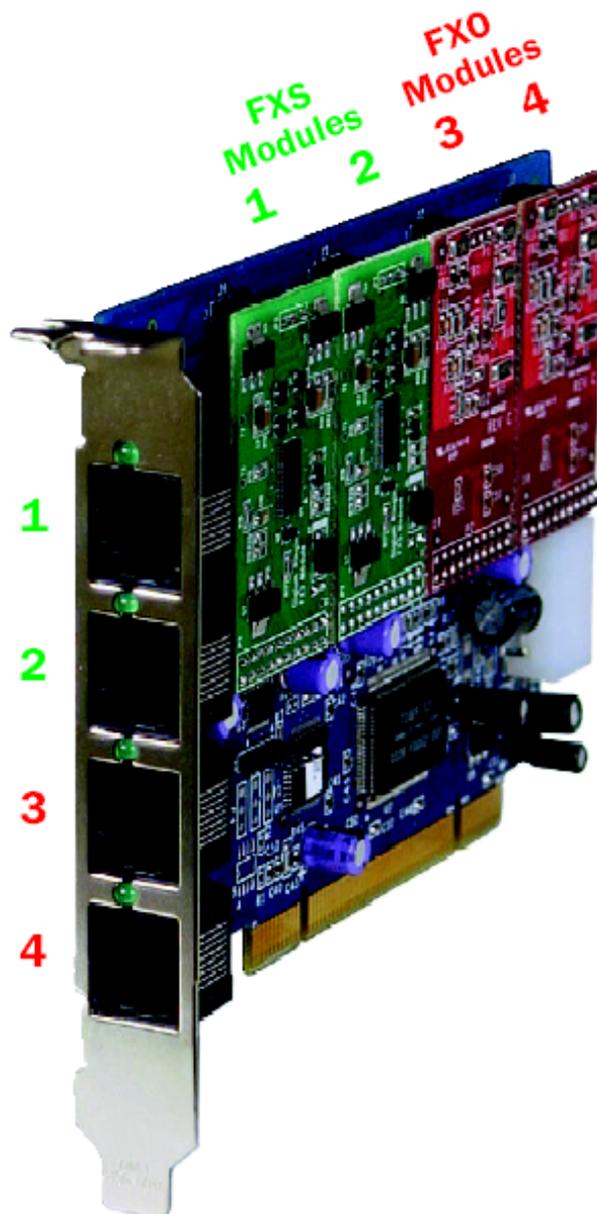
**BITTE HANDHABEN SIE DIE KARTE UND DIE MODULE MIT ÄUSSERSTER VORSICHT,  
DA DIESE TEILE SEHR EMPFINDLICH SIND. VEMEIDEN SIE WENN MÖGLICH  
STATISCHE ENTLADUNGEN.**

Seite / Page Seitennummer von Statistik

Bauen Sie die Karte folgendermaßen ein:

1. Schalten Sie den Computer aus.
2. Trennen Sie den Computer vom Netz, indem Sie den Netzstecker ziehen.
3. Stecken Sie die Karte in einen freien PCI-Slot
4. Stecken Sie vorsichtig einen freien 4-PIN 12V (abgehend vom PC-Netzteil) Stecker in den am Ende der Karte dafür vorgesehenen Anschluss.
5. Stecken Sie den Netzstecker wieder in die Steckdose, und schalten Sie den Computer ein.

Denken Sie daran, die Reihenfolge der Art der Kanäle richtet sich nach der Reihenfolge der aufgesteckten Module. Im folgenden Bild sieht man am Beispiel einer TDM22B wie sich die Reihenfolge der Ports mit der Reihenfolge der aufgesteckten Module ergibt:



Nach dem Einbau der Karte fügen Sie folgendes in Ihre „zaptel.conf“ für die verschiedenen Kombinationen ein:

<b>Kartenty p</b>	<b>Konfiguration</b>
TDM01B	<b>fxsks=1</b>
TDM02B	<b>fxsks=1-2</b>
TDM03B	<b>fxsks=1-3</b>
TDM04B	<b>fxsks=1-4</b>
TDM10B	<b>fxoks=1</b>
TDM11B	<b>fxoks=1 fxsks=4</b>
TDM12B	<b>fxoks=1 fxsks=2-3</b>
TDM13B	<b>fxoks=1 fxsks=2-4</b>
TDM20B	<b>fxoks=1-2</b>
TDM21B	<b>fxoks=1-2 fxsks=3</b>
TDM22B	<b>fxoks=1-2 fxsks=3-4</b>
TDM30B	<b>fxoks=1-3</b>
TDM31B	<b>fxoks=1-3 fxsks=4</b>
TDM40B	<b>fxoks=1-4</b>

Nachdem die Karte nun ordnungsgemäß eingebaut und konfiguriert wurde, kann der Treiber geladen werden. Zu beachten ist hierbei, dass für ältere zaptel-Versionen das Modul „wcfxs.o“ geladen werden muss. Für neuere Versionen gilt:

**modprobe wctdm <ENTER>**

Gegebenenfalls geben Sie danach noch „ztcfg <ENTER>“ ein.

Wenn das Modul korrekt geladen wurde müssten die Kontrolllampen der Modulbesetzten Ports grün leuchten, und folgendes in der „Datei“ /proc/zaptel/1 stehen, beispielhaft für eine TDM22B:

Span 1: WCTDM/0 "Wildcard TDM400P REV E/F Board 1"

- 1 WCTDM/0/0 FXOKS (In use)
- 2 WCTDM/0/1 FXOKS (In use)
- 3 WCTDM/0/2 FXSKS (In use)
- 4 WCTDM/0/3 FXSKS (In use)

### 2.3.3 TE110P



Bei der TE110P (E100P,T100P) handelt es sich um eine 1-Port T1/E1 Karte. Auf der Karte befindet sich ein Jumper, der angibt ob die Karte im T1 oder im E1 Modus arbeiten soll. Setzen Sie diesen Jumper Ihres Wunsches entsprechend. Diese Jumper gibt es bei den obsoleten E100P und T100P Karten nicht, weil hier für jede Art eine eigene Karte existierte. Bauen Sie die Karte in einen freien PCI-Steckplatz ein und Fahren Sie das System hoch.

Wenn Sie die Karte im **E1** Modus betreiben, fügen Sie Folgendes in Ihre „zaptel.conf“ ein:

```
span=1,1,0,ccs,hdb3  
bchan=1-15,17-31  
dchan=16
```

Wenn Sie die Karte im **T1** Modus betreiben, fügen Sie Folgendes in Ihre „zaptel.conf“ ein:

```
span=1,1,0,esf,b8zs  
bchan=1-23  
dchan=24
```

Laden Sie nun das Treibermodul „wcte11xp.o“ mit folgender Eingabe:

```
modprobe wcte11xp <ENTER>
```

Seite / Page Seitennummer von Statistik

Sollten Sie eine E100P oder eine T100P haben, laden Sie das Treibermodul „wct1xxp.o“:

**modprobe wct1xxp <ENTER>**

Gegebenenfalls geben Sie danach noch „ztcfg“ <ENTER> ein.

Wenn das Modul korrekt geladen wurde müsste nun folgendes in der „Datei“ / proc/zaptel/1 stehen (E1 Konfiguration):

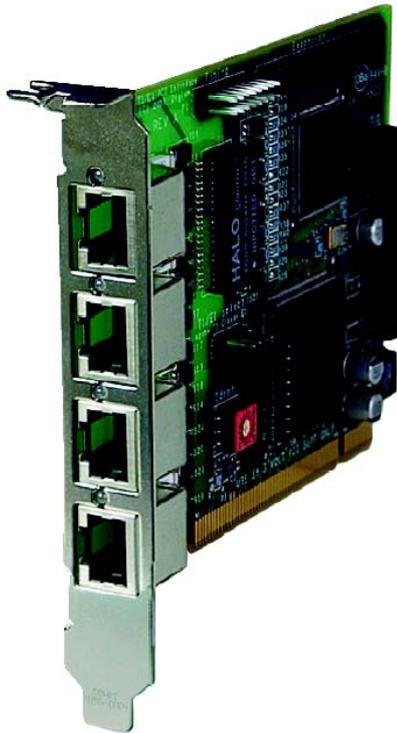
Span 1: TE1/0/1 "TE110P (PCI) Card 0 Span 1" HDB3/CCS/CRC4 ClockSource IRQ misses: 0

```
1 TE1/0/1/1 Clear (In use)
2 TE1/0/1/2 Clear (In use)
3 TE1/0/1/3 Clear (In use)
4 TE1/0/1/4 Clear (In use)
5 TE1/0/1/5 Clear (In use)
6 TE1/0/1/6 Clear (In use)
7 TE1/0/1/7 Clear (In use)
8 TE1/0/1/8 Clear (In use)
9 TE1/0/1/9 Clear (In use)
10 TE1/0/1/10 Clear (In use)
11 TE1/0/1/11 Clear (In use)
12 TE1/0/1/12 Clear (In use)
13 TE1/0/1/13 Clear (In use)
14 TE1/0/1/14 Clear (In use)
15 TE1/0/1/15 Clear (In use)
16 TE1/0/1/16 HDLCFCS (In use)
17 TE1/0/1/17 Clear (In use)
18 TE1/0/1/18 Clear (In use)
19 TE1/0/1/19 Clear (In use)
20 TE1/0/1/20 Clear (In use)
21 TE1/0/1/21 Clear (In use)
22 TE1/0/1/22 Clear (In use)
23 TE1/0/1/23 Clear (In use)
24 TE1/0/1/24 Clear (In use)
25 TE1/0/1/25 Clear (In use)
26 TE1/0/1/26 Clear (In use)
27 TE1/0/1/27 Clear (In use)
28 TE1/0/1/28 Clear (In use)
29 TE1/0/1/29 Clear (In use)
30 TE1/0/1/30 Clear (In use)
31 TE1/0/1/31 Clear (In use)
```

An der Karte befinden sich zwei Kontrolllampen, die bisher aus waren.

Nach dem Laden des Treibers bzw. des Ausführens des „ztcfg“ muss eine Lampe rot blinken, sofern der Port noch nicht mit einer Gegenstelle verbunden ist. Sollte das nicht der Fall sein, ist etwas beim Laden des Treibers schiefgegangen, oder es fehlt das „ztcfg“. Wenn Sie nun den Port mit Ihrem NT-PRI (z.B. NTPA Deutsche Telekom AG) verbinden, dann sollte eine Lampe grün leuchten. Grün bedeutet, dass die Leitung physikalisch richtig verbunden ist und die Gegenstelle aktiv ist (Layer 1 ist oben).

### 2.3.4 TE4xxP



Bei der TE4xxPP (E400P,T400P) handelt es sich eine 4-Port T1/E1 Karte.

#### **ACHTUNG!**

**Die TE410P ist für einen 3.3V 64bit PCI Bus vorgesehen, und die TE405P für einen „normalen“ 5V 32bit PCI Bus.**

Auf der Karte befindet sich eine Reihe Jumper welche angeben ob die Ports im T1 oder im E1 Modus arbeiten sollen. Setzen Sie diese Jumper Ihres Wunsches entsprechend. Diese Jumper gibt es bei den obsoleten E400P und T400P Karten nicht, weil hier für jede Art eine eigene Karte existierte. Bauen Sie die Karte in einen freien PCI-Steckplatz ein und Fahren Sie das System hoch.

Wenn Sie die Karte im E1 Modus betreiben, fügen Sie Folgendes in Ihre „zaptel.conf“ ein:

**span=1,1,0,ccs,hdb3**  
**span=2,0,0,ccs,hdb3**  
**span=3,0,0,ccs,hdb3**  
**span=4,0,0,ccs,hdb3**

**bchan=1-15,17-31**  
**dchan=16**

**bchan=32-46,48-62**  
**dchan=47**

**bchan=63-77,79-93**

Seite / Page Seitennummer von Statistik

**dchan=78**

**bchan=94-108,110-124**

**dchan=109**

Wenn Sie die Karte im T1 Modus betreiben, fügen Sie Folgendes in Ihre „zaptel.conf“ ein:

**span=1,1,0,esf,b8zs**

**span=2,0,0,esf,b8zs**

**span=3,0,0,esf,b8zs**

**span=4,0,0,esf,b8zs**

**bchan=1-23**

**dchan=24**

**bchan=25-47**

**dchan=48**

**bchan=49-71**

**dchan= 72**

**bchan=73-95**

**dchan=96**

In beiden Konfigurationen ist span 1 als „Clocksource“, zu Deutsch, Taktquelle angegeben.

Laden Sie nun das Treibermodul „wct4xyp.o“ mit folgender Eingabe:

**modprobe wct4xyp <ENTER>**

Sollten Sie eine E400P oder eine T400P haben, laden Sie das Treibermodul „tor2.o“:

**modprobe tor2 <ENTER>**

Gegebenenfalls geben Sie danach noch „ztcfg“ <ENTER> ein.

Wenn das Modul korrekt geladen wurde müsste nun für den ersten „span“ folgendes in der „Datei“ /proc/zaptel/1 stehen (E1 Konfiguration):

Span 1: TE4/0/1 "TE410P (PCI) Card 0 Span 1" HDB3/CCS/CRC4 ClockSource IRQ misses: 0

1 TE4/0/1/1 Clear (In use)  
2 TE4/0/1/2 Clear (In use)  
3 TE4/0/1/3 Clear (In use)  
4 TE4/0/1/4 Clear (In use)  
5 TE4/0/1/5 Clear (In use)  
6 TE4/0/1/6 Clear (In use)  
7 TE4/0/1/7 Clear (In use)  
8 TE4/0/1/8 Clear (In use)  
9 TE4/0/1/9 Clear (In use)  
10 TE4/0/1/10 Clear (In use)

11 TE4/0/1/11 Clear (In use)  
12 TE4/0/1/12 Clear (In use)  
13 TE4/0/1/13 Clear (In use)  
14 TE4/0/1/14 Clear (In use)  
15 TE4/0/1/15 Clear (In use)  
16 TE4/0/1/16 HDLCFCS (In use)  
17 TE4/0/1/17 Clear (In use)  
18 TE4/0/1/18 Clear (In use)  
19 TE4/0/1/19 Clear (In use)  
20 TE4/0/1/20 Clear (In use)  
21 TE4/0/1/21 Clear (In use)  
22 TE4/0/1/22 Clear (In use)  
23 TE4/0/1/23 Clear (In use)  
24 TE4/0/1/24 Clear (In use)  
25 TE4/0/1/25 Clear (In use)  
26 TE4/0/1/26 Clear (In use)  
27 TE4/0/1/27 Clear (In use)  
28 TE4/0/1/28 Clear (In use)  
29 TE4/0/1/29 Clear (In use)  
30 TE4/0/1/30 Clear (In use)  
31 TE4/0/1/31 Clear (In use)

Das gleiche müssten Sie dann auch bei Span 2, 3 und 4 sehen.

An der Karte befinden sich an jedem Port Kontrolllampen, die bisher schnell nacheinander aufgeblinkt haben. Nach dem Laden des Treibers bzw. des Ausführens des „ztcfg“ müssen die Lampen rot, im Gleichtakt blinken, sofern die Ports noch nicht mit einer Gegenstelle verbunden sind. Sollte das nicht der Fall sein, ist etwas beim Laden des Treibers schiefgegangen, oder es fehlt das „ztcfg“. Wenn Sie nun den Port mit Ihrem NT-PRI verbinden, dann sollte die Lampe grün leuchten. Grün bedeutet, dass die Leitung physikalisch richtig verbunden ist und die Gegenstelle aktiv ist (Layer 1 ist oben).

### 2.3.5 IAXy (S100I)



Beim IAXy handelt es sich um einen s.g. VoIP-ATA (Voice over IP Analog Telephone Adaptor). D.h., mit diesem Gerät kann man ein handelsübliches Analogtelefon in ein IP-Telefon verwandeln.

**Für den Betrieb des IAXy's ist ein reguliertes Gleichstromnetzteil mit 9V und einem Anschlussstecker mit Außenmaß 5,5mm und Innenmaß 2,5mm notwendig. Die IAXy's werden von beroNet ohne Netzteil ausgeliefert, da Digium selbst nur eine amerikanische Variante eines solchen Netzteils mitliefert.**

Nehmen Sie den IAXy folgendermaßen in Betrieb:

1. Schließen Sie das Analogtelefon an indem Sie den RJ11 in die dafür vorgesehene Buchse stecken (auf dem Bild links)
2. Stecken Sie das Netzwerkkabel in den LAN-Port (Auf dem Bild in der Mitte)
3. Stecken Sie die Stromversorgung in den IAXy und das andere Ende in die Steckdose, und warten Sie einen Moment.

Der IAXy ist von Hause aus auf DHCP gestellt, und versucht nun über das Netzwerk einen DHCP Server zu erreichen um einen IP-Adresse zu bekommen. Wenn Sie keinen DHCP-Server im Netzwerk haben, müssen Sie sich einen Installieren, da Sie sonst keine Verbindung zum IAXy aufnehmen können. Öffnen Sie nun die Datei „/etc/asterisk/iaxprov.conf“ und konfigurieren Sie sie nach Ihren Wünschen. Danach verbinden Sie sich mit dem Asterisk Command Line Interface:

**asterisk -r <ENTER>**

und geben Sie folgendes Kommando ein:

**iax2 provision <IP des IAXy's> <name des Templates aus der iaxprov.conf>**

Damit müsste der IAXy nun konfiguriert sein. Starten Sie den IAXy neu, indem sie den Netzstecker ziehen und wieder hereinstecken.

## 2.4 Einbinden der Kanäle in Asterisk (zapata.conf)

An dieser Stelle sei zuerst erwähnt, dass die Konfiguration der zapata.conf eine umfangreiche Angelegenheit sein kann, und hier nicht auf alle möglichen Fälle oder Probleme eingegangen werden kann. Es wird lediglich die Einbindung der Kanäle und der damit in Zusammenhang stehenden Parameter behandelt.

Welche Digium-Karte (zaptel) Sie auch immer eingebaut haben, alle stellen Ihre Kanäle in Form von „ZapChannels“ der Asterisk zur Verfügung. Leider holt sich die Asterisk die verfügbaren Kanäle nicht automatisch, d.h., man muss Ihr also noch mitteilen welche Kanäle Sie einfügen soll, und welche Signalisierung damit verbunden werden soll. Wenn Sie sich die mitinstallierte Beispielkonfigurationsdatei „/etc/asterisk/zapata.conf“ ansehen, werden Sie merken, dass es hier eine Vielzahl von Parametern gibt die aktiviert oder deaktiviert werden können. Auch ist nicht immer klar welcher Parameter mit welchem Typ Kanal in Verbindung steht. Meist sind als Erklärung noch vorgesetzte, auskommentierte Textzeilen eingefügt. Wie gesagt die zapata.conf kann ein Umfangreiches Thema sein, daher wird hier im Folgenden nur auf das Wichtigste eingegangen.

Grundsätzlich werden Zaptel-Kanäle folgendermaßen eingebunden:

```
option1= value1
option2= value2
option n = value xy
channel => 1-4
```

**An dieser Stelle sei nocheinmal angemerkt, dass die Reihenfolge und Art der Kanäle sich nach der Reihenfolge des Ladens der Treibermodule richtet.**

Mit "option" ist eine Eigenschaft (Einstellung) an einem "Channelobjekt" gemeint. D.h., alle vor dem Schlüsselwort „channel“ gesetzten Optionen gelten für die angegebenen Kanäle. Es können so viele Kanaleinträge vorhanden sein wie Kanäle die von den jeweiligen Treibern bzw. Karten zur Verfügung gestellt werden. Will man jetzt für weitere folgende Kanäle Optionen verändern, so setzt man sie einfach nochmal, nur mit anderem Inhalt:

```
option1= value1
option2= value2
channel => 1-4
```

```
option1=value1.5
channel => 5-8
```

So setzt man für Kanal 5-8 in Bezug auf Option1 eine andere Einstellung wie bei Kanal 1-4. Der Wert für Option2 bleibt jedoch gleich, weil er vor dem zweiten Kanaleintrag nicht noch mal gesetzt worden ist.

### 2.4.1 Einbinden von Analogkanälen (FXS/FXO)

Um eine TDMxxP einzubinden müssen Sie nun entsprechend der Reihenfolge in Ihrer „zaptel.conf“ die angegebenen Kanäle einbinden.

Für das Einbinden eines FXS Kanals würde das folgendermaßen aussehen:

```

signalling=fxo_ks
context=dialout
group=1
callerid="TestName" <123456789>
channel => 1

```

Damit hätten Sie jetzt einen FXS-Port z.B. von einer (TDM10B) eingebunden.

Für die Signalisierung („signalling“) müssen Sie bei analogen Kanälen immer genau das Gegenteil einstellen, wie das wofür er gedacht ist. D.h. für FXS-Ports muss eine FXO-Signalisierung, „**signalling=fxo\_ks**“, und für FXO-Ports eine FXS-Signalisierung, „**signalling=fxs\_ks**“, angegeben werden.

Die Option „context“ gibt an, in welchem Konfigurationsabschnitt (Kontext) des „Dialplans“ eingehende Rufe geleitet werden sollen. Hier wird ein auf dem Kanal 1 eingehender Ruf in den Kontext „dialout“ geleitet.

Die Option „group“ bestimmt eine aus einem oder mehreren Kanälen bestehende Gruppe. Stehen mehrere Kanäle zur Verfügung, ist es möglich diese in Gruppen zusammenzufassen. D.h., wenn ich z.B. ausgehende Rufe absetzen möchte, und automatisch der erste freie Kanal verwendet werden soll, kann ich über diese Gruppen rufen. Der „Dialstring“ bezieht sich dann nicht auf einen einzelnen Kanal (*Dial(Zap/1/\${EXTEN})*) sondern eben auf eine Gruppe von Kanälen (*Dial(Zap/g1/\${EXTEN})*). Wie oben schon gezeigt, werden Gruppen folgendermaßen definiert:

```

option1 ....
group=1
channel=>1-4
channel=>5-8

group=2
channel=9-12

```

So wären Kanal 1-8 in Gruppe 1 und 9-12 in Gruppe zwei.

Die Option „callerid“ bestimmt den Absendernamen und die Absendernummer des Kanals bzw. Gerätes. Diese Daten werden wenn möglich mitsignalisiert.

Bei dem Schlüsselwort „channel“ ist die Konfiguration für die darin angegebenen Kanäle abgeschlossen. Nachfolgende Optionen werden nur noch für nachfolgende Kanäle berücksichtigt.

## 2.4.2 Einbinden von ISDN-Kanälen

Das Einbinden von ISDN-Kanälen erfolgt grundsätzlich genauso wie in Punkt 2.4.1 beschrieben. Es gibt jedoch noch einige Besonderheiten, die beachtet werden müssen.

Zusätzlich zu den in Punkt 2.4.1 beschriebenen Optionen müssen folgende Optionen zusätzlich oder anders gesetzt werden:

### **busydetect=no**

Sollte auf "no" stehen, da diese Option für analoge Kanäle gedacht ist, und im Zusammenhang mit ISDN zu Problemen führen kann.

### **callprogress=no**

Hier gilt dasselbe wie für "busydetect".

### **switchtype=euroisdn**

Gibt die Art des "Switchings" an. Diese Option muss in Deutschland und den meisten EU Staaten auf den Wert „euroisdn“ gesetzt werden. In den USA müsste er auf „national“ gesetzt werden.

### **pridialplan=unknown**

Setzt den ausgehenden ISDN Nummernplan.

Möglich Werte:

<b>unknown:</b>	<b>Unknown</b>
<b>private:</b>	<b>Private ISDN</b>
<b>local:</b>	<b>Local ISDN</b>
<b>national:</b>	<b>National ISDN</b>
<b>international:</b>	<b>International ISDN</b>

### **localpridialplan=national**

Setzt den eingehenden ISDN Nummernplan.

Möglich werte: siehe „pridialplan“

### **immediate=no**

Bestimmt, ob der Ruf sofort nach dem Abheben in den entspr. Kontext bzw. in den Dialplan geleitet wird, oder ob erst ein Freizeichen kommt, und während des Wählens auf ein Timeout gewartet wird.

### **overlapdial=yes**

Legt fest, ob Ziffern nachgewählt werden können. Wenn „Overlapdial“ möglich sein soll, muss „immediate“ deaktiviert werden.

### **echocancel=no**

Software Echocanceling an oder aus. Sollte hier aus sein, weil im ISDN normalerweise kein Echo entsteht, und das Echocanceling sehr rechenintensiv ist.

### **echocancelwhenbridged=no**

### **echotrainig=no**

**signalling=pri\_cpe**

Bestimmt die Art der Schicht-3-Siganlisierung. Normalerweise möchten Sie den PRI Port an einen NTPM Ihres Providers anbinden, und dafür muss der Port als „TE“ (Termination Endpiont) agieren. Wenn Sie den Port im „NT“ Modus betreiben wollen, geben Sie hier den Wert „pri\_net“ an.

Außerdem muss bei der Einbindung der ISDNkanäle darauf geachtet werden, dass der d-Kanal nicht mit eingebunden wird:

**channel => 1-15,17-31**

Damit ist ein PRI-ISDN Port in der Asterisk angemeldet.

Weitere Konfigurationsbeispiele finden Sie im Anhang.

### 3. Installation und Konfiguration von beroNet Karten

Bei den BNxS0 Karten handelt es sich um s.g. Basic Rate Interface Karten (BRI). Diese Karten sind mit 4 (BN4S0) oder 8 (BN8S0) ISDN-Ports ausgestattet. Die Ports können einzeln als TE (Endgerät) oder NT (Netzanschluss) konfiguriert werden. Wenn an den NT-Ports ISDN-Telefone betrieben werden sollen die nicht selbst gespeist sind, kann zusätzlich eine Spannungsversorgung (BNPW1) an die Karte angeschlossen werden. Auch die Art der Signalisierung kann pro Port eingestellt werden. D.h. die Ports können jeweils entweder als „Punkt zu Mehrpunkt“ (Mehrgeräteanschluss) oder „Punkt zu Punkt“ (Anlagenanschluss) betrieben werden.

#### 3.1. Einbau der BN4S0

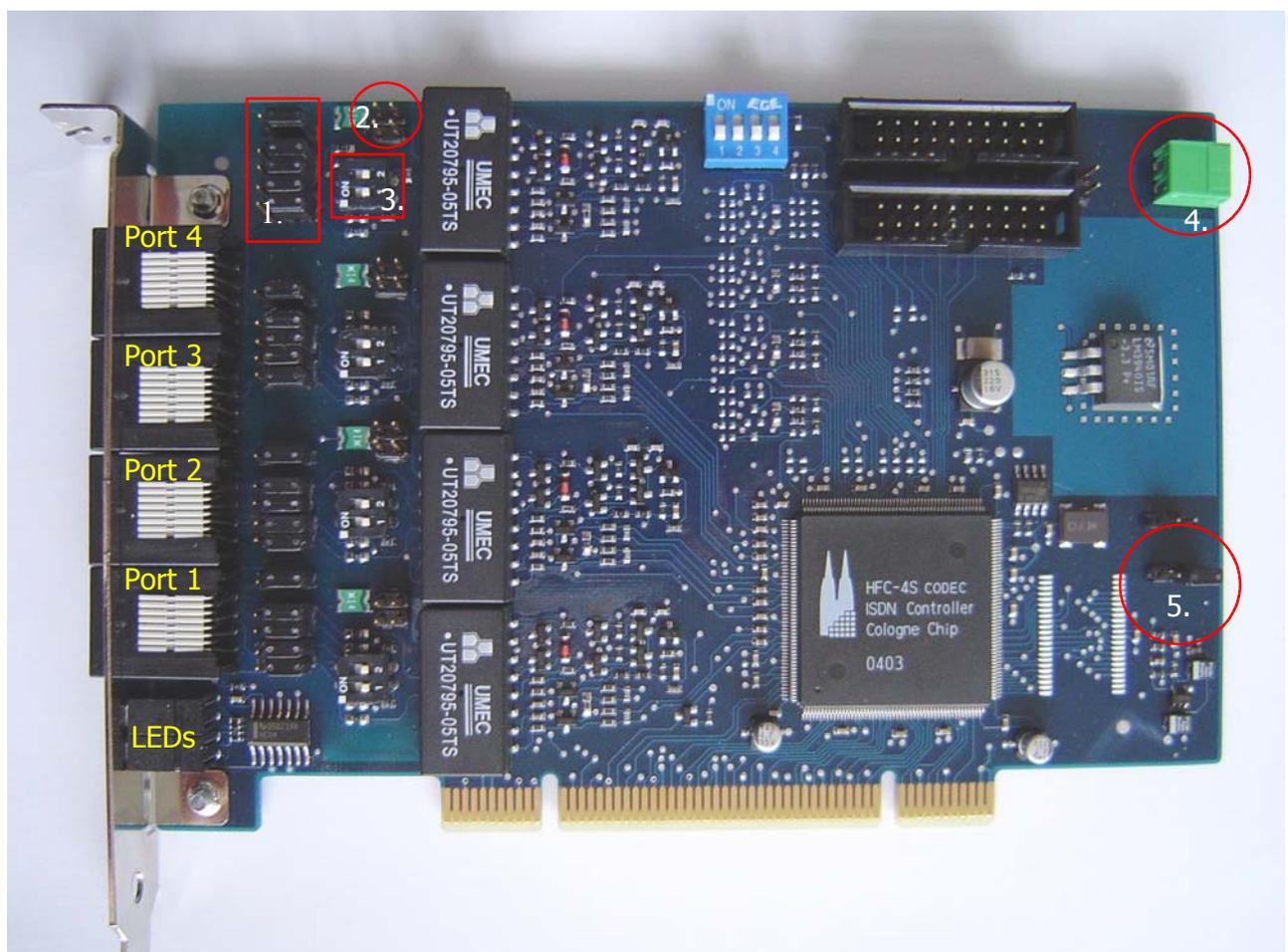


Abb. 3.1

Bevor Sie die BN4S0 einbauen, sollten Sie sich überlegen wie Sie Ihre vier zur Verfügung stehenden Ports betreiben wollen. D.h., Sie müssen sich entscheiden, an welchen Ports Sie „Endgeräte“ anschließen wollen, und welche Endgeräte sein sollen.

Seite / Page Seitennummer von Statistik

In Abbildung 3.1 sehen Sie eine Übersicht der Karte. Sie sehen vier Ports mit jeweils einer rechts danebenliegenden Gruppe von Jumpern und DIP-Schaltern.

Die rot markierten Bereiche haben folgende Bedeutungen:

1. Gruppe, bestehend aus 5 Jumpern zum Einstellen der Portart NT oder TE. Verbinden die Jumper die rechten beiden Pins (Werkseinstellung), ist der Port als „TE“ konfiguriert. Verbinden die Jumper die linken beiden Pins, ist der Port als „NT“ konfiguriert. Zu beachten ist, dass immer die gesamte Reihe der Jumper eines Ports gesteckt werden muss.
2. Diese Gruppe von Pins muss in gleicher Richtung wie die Pins in Punkt 1. geschlossen werden, wenn der Port von einem an Punkt 4. angeschlossenen BNPW1 gespeist werden soll.
3. Diese DIP-Schalter sind Werkseitig immer auf Position „on“. Das bedeutet, dass der Endwiderstand von 100 Ohm aktiviert ist. Soll das nicht der Fall sein, schalten Sie hier einfach auf „off“.
4. Anschluss für die Stromversorgung BNPW1.
5. Jumper für die PCI Versorgungsspannung (3.3V / 5V)

**Wichtig: Bei älteren Motherboards kann es vorkommen, daß die Spannung am PCI Bus nicht automatisch erkannt wird. In diesem Fall sollten Sie den Jumper (Nr. 5 in Abb 3.1) auf die Stellung „3V3 reg.“ (die rechten beiden PINs müssen verbunden sein) stellen.**

**Tipp:** Nehmen Sie beim Umstecken der Jumper eine kleine Telefonzange zur Hilfe, und greifen Sie damit die Jumper von oben.

Wenn Sie nun alles richtig und nach Ihren Vorstellungen eingerichtet haben, kann die Karte in einen freien PCI-Slot gesteckt werden. Gehen Sie dabei folgendermaßen vor:

1. Schalten Sie den Computer aus, und trennen Sie ihn vom Netz, indem Sie den Netzstecker ziehen.
2. Stecken Sie die Karte in ein freien PCI Steckplatz, und schrauben Sie die Karte fest.  
**Gehen Sie beim Einstecken der Karte mit äußerster Vorsicht zu Werke, da einige Bauteile auf der Karte beim Einstecken beschädigt werden können, indem Sie vielleicht an einer anderen Karte oder am Gehäuse hängen bleiben.**
3. Schrauben Sie den Computer zu, stecken Sie den Netzstecker wieder ein, und schalten Sie den Computer wieder an.

Beim Anschliessen von ISDN Anschlüssen oder ISDN Endgeräten, müssen Sie nichts weiter beachten, da Kreuzen der Anschlusspins bereits auf der Karte durch die Jumper vorgenommen wird.

### 3.2. Einbau der BN8S0

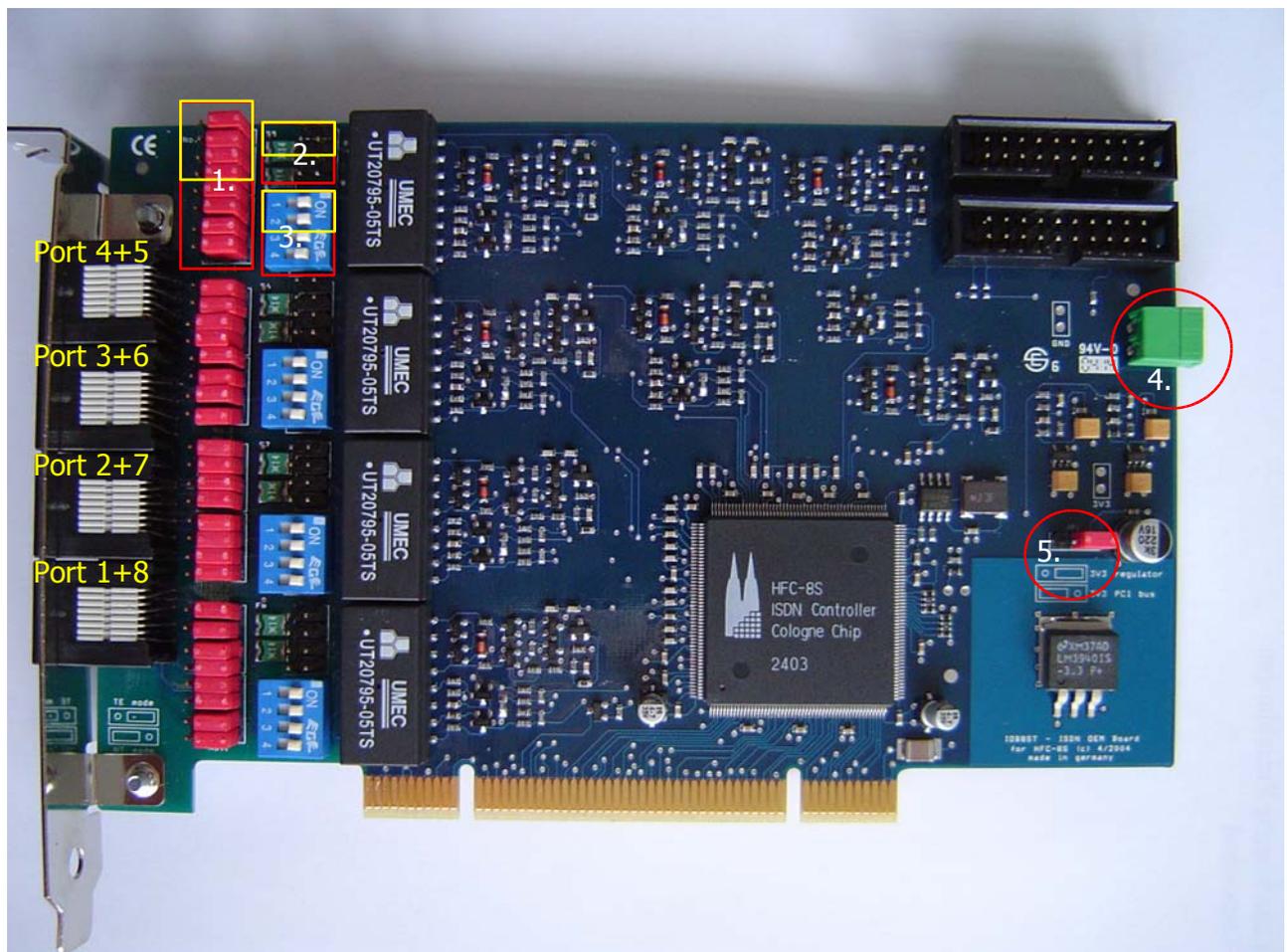


Abb. 3.2.

Wie schon in Punkt 3.1. erwähnt, muss auch hier erst mal überlegt werden wie die einzelnen Ports betrieben werden sollen. Was hier beachtet werden muss, ist, dass hier auf einem Anschluss jeweils zwei ISDN Ports geschaltet sind. Die genaue Belegung der Anschlusspins entnehmen Sie bitte dem Anschlussplan im Anhang (7.2). Die Jumper-Gruppen und DIP-Schalter beziehen sich also immer auf zwei ISDN-Ports. Deshalb sind die für einen ISDN Port zuständigen Jumper und DIP-Schalter farbig voneinander abgehoben. Für die Stellung der Jumper und DIP-Schalter gilt dasselbe wie für die BN4S0 in Punkt 3.1.

**Achten Sie darauf, das die Portreihenfolge gegenläufig zählt !**

Gehen Sie beim Einstecken der Karte genau wie in Punkt 3.1 beschrieben vor.

**An dieser Stelle sei noch mal darauf hingewiesen, dass äußerste Umsicht beim Einstecken der Karte notwendig ist.**

### 3.3. Anschluss des BNPW1

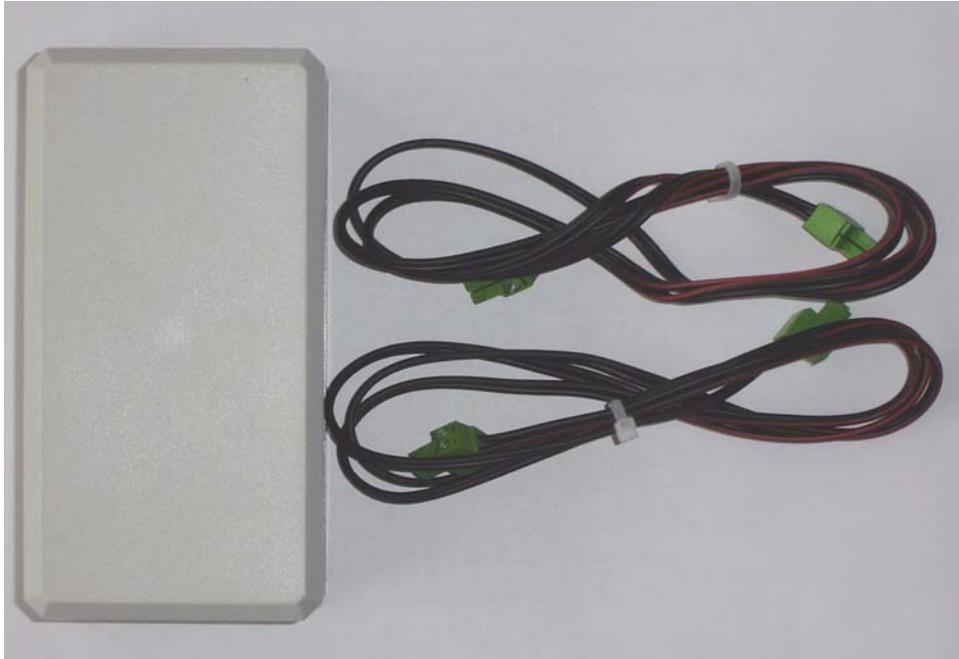


Abb 3.3.1

Bei der BNPW1 handelt es sich um ein Speisernetzteil, welches es ermöglicht die ISDN Ports auf der Karte mit Strom zu versorgen, falls sie als NT konfiguriert sind, und daran nicht selbstgespeiste ISDN Endgeräte betrieben werden sollen.

Wenn Sie eine BNPW1 mitbestellt haben, haben Sie wie in Abb. 3.3.1 ein graues Gehäuse (Netzteil) und zwei Anschlusskabel mitgeliefert bekommen. An den Kurzen Seiten des Gehäuses befindet sich jeweils eine Buchse.



Abb. 3.3.2



Abb. 3.3.3

In Abb. 3.3.2 sehen Sie die Buchsen für den Anschluss an die BNxS0. Auf der BnxS0 befindet sich ebenfalls eine solche Buchse. Stecken Sie eines der mitgelieferten Anschlusskabel in eine der beiden Buchsen am Netzteil und das andere Ende in die Buchse an der Karte. Damit die Ports auch wirklich gespeist werden, setzen Sie auch die Jumper entsprechend wie in Punkt 3.1 beschrieben.

Seite / *Page* Seitennummer von Statistik

In Abb. 3.3.3 sehen Sie eine Buchse für den Anschluss eines 12V Steckers. Stecken Sie dort einen freien 12V Stecker von Ihrem PC Netzteil rein.

Verstauen Sie das Netzteil innerhalb Ihres Computergehäuses so, daß es sich möglichst nicht bewegen kann. Am besten fixieren Sie das Netzteil am Boden des PC-Gehäuses mit Schrauben von aussen, oder Sie kleben es mit einer Heißklebepistole fest.

### 3.4. Anschlussdose für BN8S0



Abb. 3.4.1

Wenn Sie eine BN8S0 gekauft haben, liefern wir eine Anschlussdose mit. Da immer zwei ISDN Interfaces auf einem physikalischen Port liegen, müssen die Pins entsprechend rausgeführt werden. In folgender Abbildung sehen Sie die offene Anschlussdose:

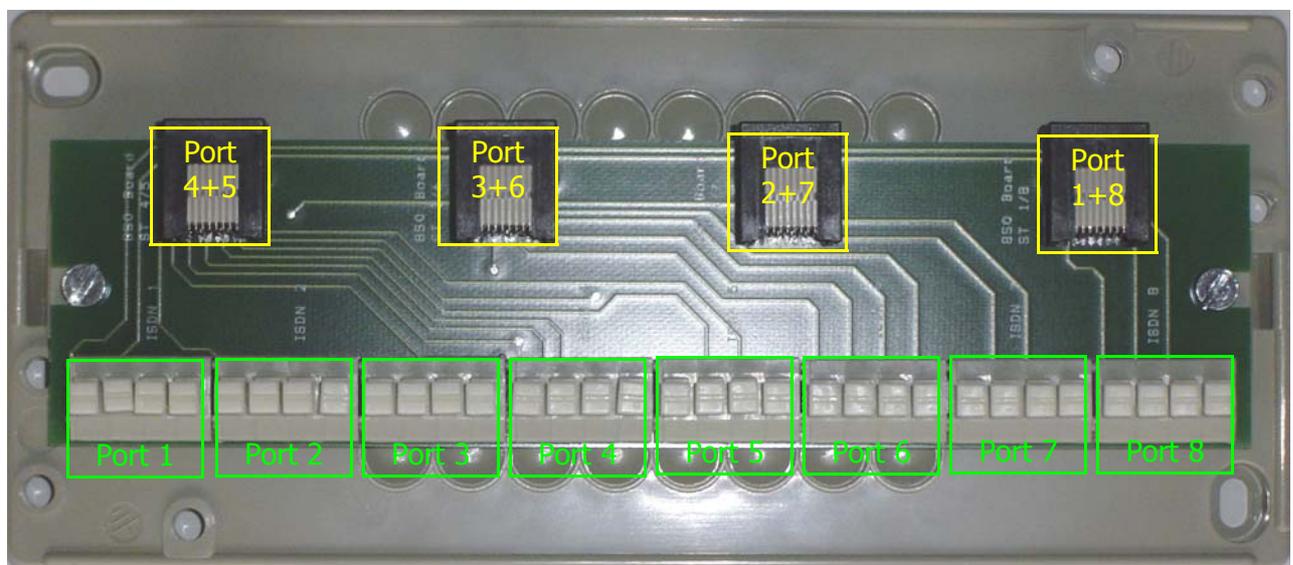


Abb. 3.4.2.

Oberhalb sind vier RJ45 Buchsen (gelb) angebracht. Jede dieser vier Buchsen ist für eine Buchse auf der Karte vorgesehen, und sollte mit je einem der 8-adrigen Kabel verbunden werden. Danach stehen unterhalb für jeden Port eine 4-polige Anschlussklemme zur Verfügung, um daran weiterführende Installationen anzuschliessen.

Seite / Page Seitennummer von Statistik

Wenn Sie eine weiterführende ISDN-Dose an den Anschlussklemmen anschliessen möchten, achten Sie auf die Belegung der Adern:

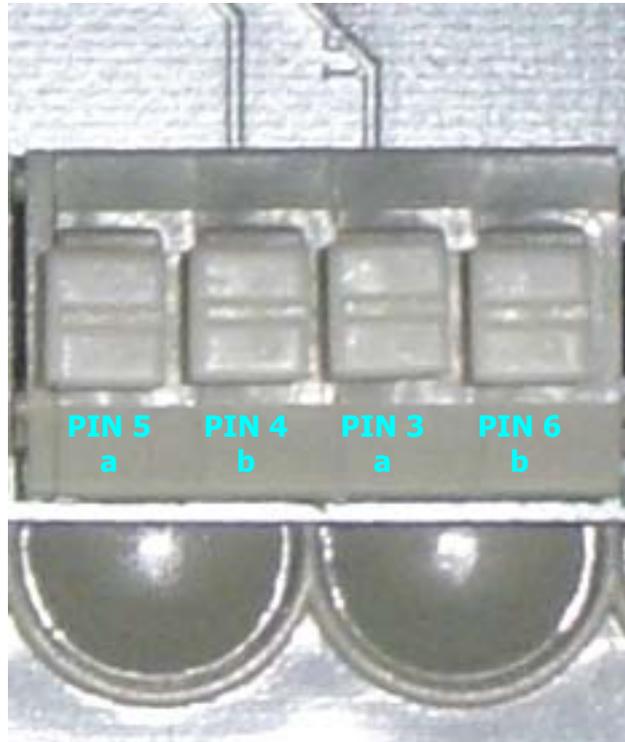


Abb 3.4.3

In Abb. 3.4.3 sehen Sie eine vergrößerte Darstellung einer Anschlussklemme. Darin sind die Nummern der PINs eingezeichnet, die Sie mit den entsprechenden PINs an der ISDN-Dose verbinden müssen. Unterhalb der PIN-Nummern steht auch welcher Teil eines ISDN-Adernpaares angeschlossen werden muss (a oder b).

### 3.5. mISDN Installieren

Wie Sie vielleicht schon wissen, ist mISDN ([www.isdn4linux.de/](http://www.isdn4linux.de/)) die zukünftige ISDN-Schicht für Linux. mISDN ist das Nachfolgeprojekt von „hisax“, und wird von der SuSE AG ([www.suse.com/](http://www.suse.com/)) und einigen freien Entwicklern wie Andreas Eversberg ([www.jolly.de](http://www.jolly.de)) gepflegt und ständig weiterentwickelt. Durch den höchst flexiblen und modularen Aufbau von mISDN ist es nun möglich verschiedenste Applikationen und Zwischenschichten an ein von mISDN unterstütztes ISDN Device anzubinden. Ein gutes Beispiel dafür ist natürlich CAPI. Vormalig gab es ja ein CAPI4Linux Projekt, was jetzt auch in mISDN aufgeht, da fast alle vormalig unter „hisax“ unterstützten Karten auf mISDN portiert werden.

Auch die beroNet Karten bauen hier auf die Zukunft, und nutzen das mISDN als Basis für einen Channel-Treiber der es ermöglicht die beroNet Karten und nebenbei alle anderen in mISDN unterstützten Karten in der open Source PBX Asterisk zu verwenden. Einige Leute werden nun sagen, „ja aber es gibt doch schon ein chan\_capi...“, womit man ja ebenfalls die mISDN Karten in der Asterisk nutzen könnte. Ja, das stimmt. Das Problem ist jedoch, dass CAPI kein NT Modus unterstützt, und somit die Vorteile einer HFC-Chip basierten Karte nicht voll genutzt werden können. Denn die beroNet Karten bieten einen solchen Modus, und dieser soll natürlich auch genutzt werden.

Um den von beroNet zur Verfügung gestellten Asterisk Channel-Treiber nutzen zu können, muss zuerst mISDN in Ihr Linuxsystem integriert werden.

**Ab hier wird davon ausgegangen, dass Sie einen 2.6'er Kernel haben, und das die Asterisk kompiliert und installiert auf Ihrem System ist. Sollte es bei Ihnen notwendig sein einen neuen Kernel zu kompilieren, finden Sie ein HOWTO unter:**

**<http://www.linuxdocs.org/HOWTOs/Kernel-HOWTO.html>**

### 3.5.1. mISDN manuell installieren

Da es sich bei mISDN um Kernel-Module handelt, muss nun Ihr Kernel entsprechend „gepatched“ werden. In Zukunft wird das mISDN standardmäßig Bestandteil in den offiziellen Kernel-Quellen sein. Bis es soweit ist, muss der Kernel „gepatched“ werden. Dazu benötigen Sie die Kernel-Quellen Ihres aktuell laufenden Kernels und natürlich die Kernelkonfiguration. Empfehlenswert wäre eine Kernelversion ab 2.6.8 . Sollten Sie Ihre Kernel-Quellen nicht auf Ihrem System haben, installieren Sie sie nach. Bei einem Debian basierten System funktioniert das wie folgt:

**apt-get install kernel-source-<ihre Kernelversion> <ENTER>**

Nach dem Download befindet sich im Verzeichnis „/usr/src/“ eine Datei mit folgendem Namen:

**kernel-source-<ihre Kenelversion>.tar.bz**

Gehen Sie in das Verzeichnis, und packen Sie die Datei wie folgt aus:

**tar -vxjz kernel-source-<ihre Kenelversion>.tar.bz <ENTER>**

Danach befinden sich die Kernel-Quellen im Verzeichnis:  
„/usr/src/kernel-source-<ihre Kernelversion>/“

Für SuSE Systeme brauchen Sie nur das Paket „kernel-source-<Kernelversion>.i586.rpm“ von CD oder DVD zu installieren.

Nachdem nun die Kernel-Quellen auf Ihrem System installiert sind, müssen Sie noch die Kernel-Konfigurationsdatei in das Kernel-Quellenverzeichnis kopieren. Ihre Konfigurationsdatei finden Sie meist im Verzeichnis „/boot/“ unter dem Namen „config-<Kernelversion>“:

**cp /boot/config-<Kernelversion> /usr/src/kernel-source-<Kernelversion>/.config <ENTER>**

Nun führen Sie folgendes aus:

**make menuconfig <ENTER>**

Seite / Page Seitennummer von Statistik

Es öffnet sich das Kernel-Konfigurationsfenster.

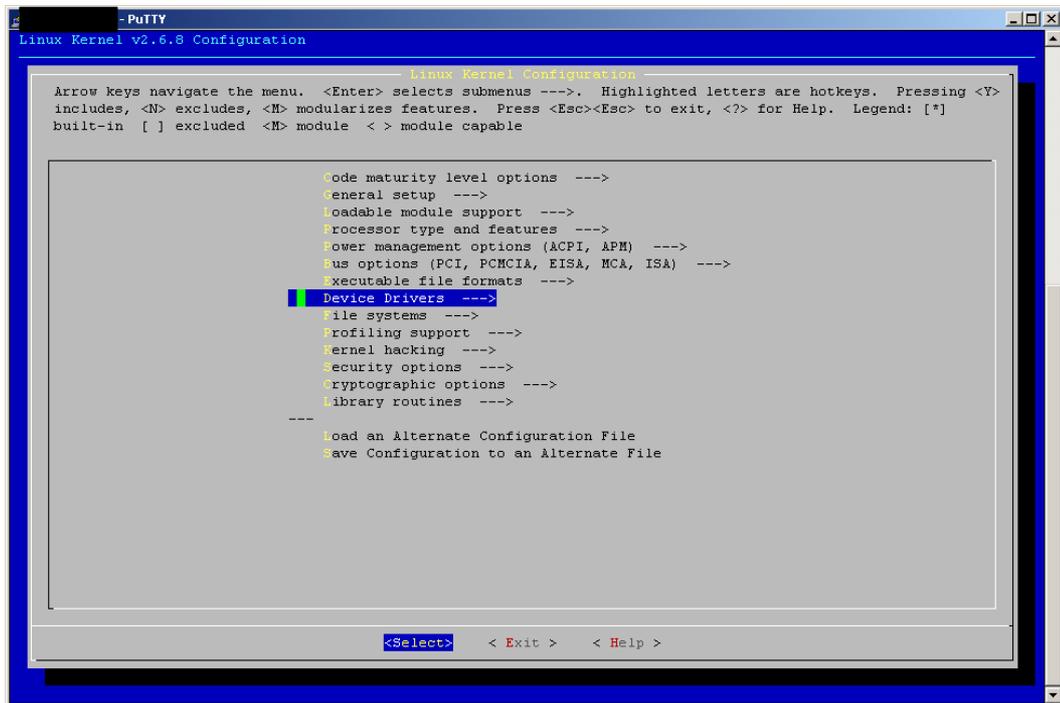


Abb. 3.5.1

Beenden Sie hier mit „Exit“. Sie werden gefragt, ob Sie die Konfigurationsdatei speichern wollen, antworten Sie mit „Ja“.

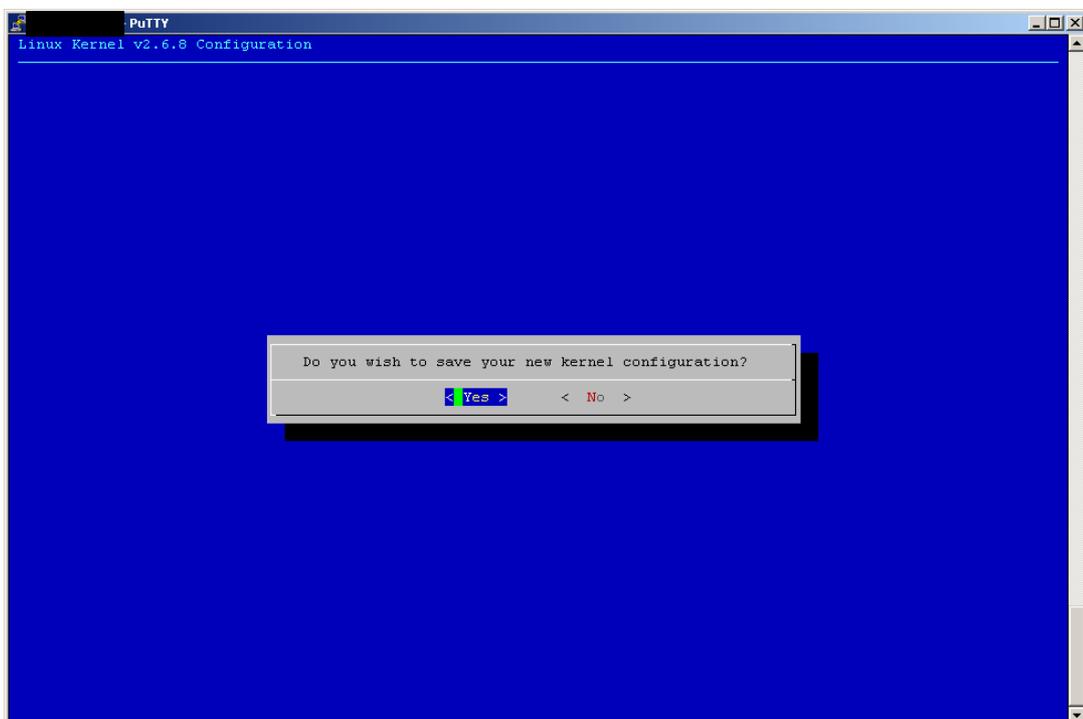


Abb. 3.5.2

Seite / Page Seitennummer von Statistik

Danach müssen Sie noch, falls die Kernelversionen sich unterscheiden, den Verzeichnislink „/usr/src/linux“ und „/usr/src/linux-2.6“ auf das richtige Kernelquellenverzeichnis verweisen lassen.

Nun muss das mISDN installiert werden. Dafür werden zwei Pakete benötigt. Das Paket „mISDN“ und „mISDNUser“. Diese Pakete können Sie entweder unter <ftp://ftp.isdn4linux.de/pub/isdn4linux/CVS-Snapshots/> herunterladen (bitte immer die neuesten Versionen nehmen) oder Sie laden die Pakete direkt aus dem CVS von isdn4linux.org, was wir auch für empfehlenswert halten:

Gehen Sie in das Verzeichnis „/usr/src/“, und geben Sie folgendes ein:

```
export CVSROOT=:pserver:anonymous@cvs.isdn4linux.de:/i4ldev <ENTER>  
cvs login <ENTER>
```

Geben Sie das CVS-Passwort ein: „readonly“  
Danach:

```
cvs co mISDN mISDNUser <ENTER>
```

Nachdem der Download beendet ist, befinden sich zwei neue Verzeichnisse in „/usr/src“:

- ./mISDN
- ./mISDNUser

Gehen Sie in das Verzeichnis „/usr/src/mISDN“, und führen sie folgendes Skript aus:

```
./std2kern <ENTER>
```

Nun sollten Ihre Kernelquellen „gepatched“ sein.  
Jetzt muss der Kernel nochmal konfiguriert werden. Gehen Sie dazu in das Verzeichnis „/usr/src/linux“ und geben Sie nochmal folgendes ein:

```
make menuconfig <ENTER>
```

Sie sehen wieder das Kernelkonfigurationsmenu wie in Abb. 3.3.1.

Navigieren Sie sich und aktivieren Sie die Module folgendermaßen:

#### Device Drivers

- ISDN Subsystem**
  - <M> ISDN Support**
  - <M> CAPI**
  - 
  - 
  - 
  - Modular ISDN Driver**

Nachdem Sie im Punkt "Modular ISDN Driver" angekommen sind, sehen Sie folgendes Bild:

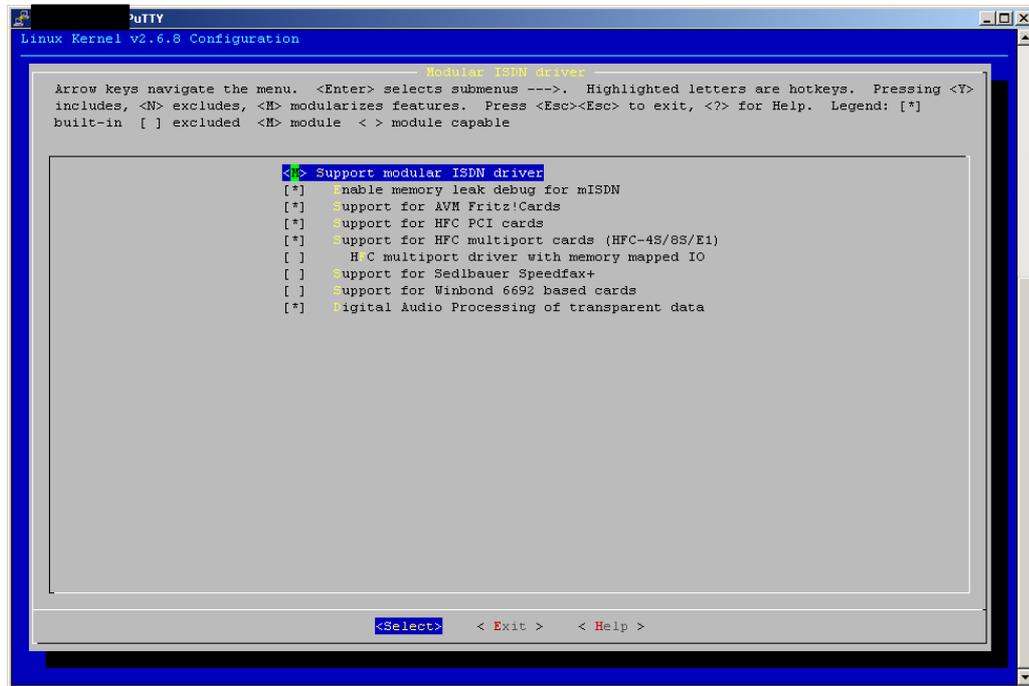


Abb. 3.5.3

Aktivieren Sie die Module entsprechend Abb. 3.5.1 .

Beenden Sie mit „Exit“, und speichern Sie die Konfiguration entsprechend Abb. 3.5.5 .  
Jetzt gehen Sie in das Verzeichnis „/usr/src/linux“ und geben Sie folgendes ein:

**make modules modules\_install <ENTER>**

Das könnte je nach Ausstattung der Maschine etwas dauern. Wenn das „make“ durchgelaufen ist, sollten die mISDN Kernelmodule auf Ihrem System installiert sein. Begeben Sie sich nun in das Verzeichnis „/usr/src/mISDNuser“, und geben Sie hier folgendes ein:

**make <ENTER>**

Beim Kompilieren von mISDNuser kann folgende Fehlermeldung auftreten:

```
....  
gsm.h: No such file or directory
```

....  
Das ist für die weiterführende Installation nicht relevant. Sie können diese Fehlermeldung getrost misachten.

Wenn alles fehlerfrei durchgelaufen ist, dann ist die Installation ersteinmal abgeschlossen.

### 3.5.2 Holen und kompilieren von chan\_mISDN

Nachdem die Pakete mISDN und mISDNuser nun installiert sind, das „hfcmulti“-Kernelmodul geladen ist, kann der „chan\_misdn“ Channel-Treiber installiert werden.

Es wird hier nochmals darauf hingewiesen, daß eine installierte Asterisk vorausgesetzt wird.

Laden Sie sich das neueste Release von chan\_misdn unter <http://www.beronet.com/download/> herunter, und entpacken Sie es ebenfalls im Verzeichnis „/usr/src“. Wenn Sie gegen die aktuelle Headrevision bzw. Versionen jünger als 1.0.2 der Asterisk kompilieren wollen, müssen Sie im chan\_misdn-Makefile folgende Option auskommentieren:

**#CFLAGS+=-DASTERISK\_STABLE**

Gehen Sie in das nun entstandene Verzeichnis „/usr/src/chan\_misdn/“ und geben Sie folgendes ein:

**make install <ENTER>**

Wenn alles fehlerfrei durchgelaufen ist, ist chan\_misdn nun auf Ihrem System installiert.

### 3.5.3. mISDN automatisch mit dem beroNet Makefile installieren

Wenn Sie sich gegen die manuelle Installationsvariante entschieden haben, haben Sie die Möglichkeit mit dem beroNet Makefile zu arbeiten. Laden Sie sich dazu folgendes Paket herunter: <http://www.beronet.com/download/install-misdn.tgz>

***Hierfür ist ebenfalls Voraussetzung, daß sich die Kernelquellen oder mindestens die Kernel-Header und Asterisk installiert auf Ihrem System befinden.***

Kopieren Sie das Paket nach „/usr/src“ und entpacken Sie es folgendermassen:

**tar -vxjz install-misdn.tgz <ENTER>**

Danach gehen Sie in das neu entstandene Verzeichnis „/usr/src/install-misdn/“, und geben Sie folgendes ein:

**make install <ENTER>**

Nun startet ein vollautomatisches Installationsskript welches alle nötigen Pakete selbständig herunterlädt und installiert. Danach haben Sie mISDN und chan\_misdn auf Ihrem System fertig installiert.

### 3.5.4. Laden des Treibermoduls hfcmulti

Damit chan\_misdn in der Asterisk verwendet werden kann, muss vorher das entsprechende misdn-Kernelmodul geladen werden. In unserem Fall ist das ein Modul namens "hfcmulti". Wenn Sie sich erinnern können, haben Sie sich in Punkt 3.1 oder 3.2 für eine Portkonfiguration entschieden, und dieser Konfiguration entsprechend haben Sie die Jumper gesetzt. Da es leider nicht möglich ist diese Jumperkonfiguration bei jeder Karte über den Treiber auf der Karte auszulesen, müssen Sie beim Laden des Treibers noch angeben, welcher Port ein TE und welcher ein NT Port ist. Zusätzlich müssen Sie auch noch bestimmen, welcher Port im "Punkt zu Punkt" Modus und welcher im "Punkt zu Mehrpunkt" Modus laufen soll. Dafür gibt es die beiden folgenden Argumentoptionen, die beim Laden des Treibers mit angegeben werden müssen:

1. "layermask"

Mögliche Werte:

"0x3" für NT

"0xf" für TE

2. "protocol"

Mögliche Werte :

"0x2" für Punkt zu Mehrpunkt (PmP) bei einem TE Port

"0x22" für Punkt zu Punkt (PP) bei einem TE Port

"0x12" für Punkt zu Mehrpunkt bei einem NT Portart

("0x32" für Punkt zu Punkt bei einem NT Portart. Diesen Wert gibt es z.Zt. Noch nicht. Howto, siehe Ende Punkt 3.5)

"0x8002" für TE/PmP und Port als Zeitgeberquelle

"0x8022" für TE/PP und Port als Zeitgeberquelle

3. "type"

Mögliche Werte:

"0x04" für BN4S0

"0x08" für BN8S0

"0x01" für BNE1

**Bitte achten Sie bevor Sie das „hfcmulti“ Modul laden darauf, daß Sie keine anderen Module die auf mISDN zugreifen, wie z.B. CAPI, oder denselben Chipsatz unterstützen, wie z.B. hisax geladen haben. !**

Wenn Sie andere Karten wie z.B. die FritzCard benutzen wollen, dann laden Sie bitte das entsprechende Treibermodul (FritzCard = avmfritz), und geben Sie nur die "layermask" und den "protocol"-Parameter an.

Seite / Page Seitennummer von Statistik

Das Laden des Treibers erfolgt dann folgendermaßen:

```
modprobe hfcmulti type=0x04 layermask= 0xf,0xf,0x3,0x3 protocol=0x2,0x2,0x12,0x12 <ENTER>
```

**Wichtig:** Um chan\_misdn benutzen zu können muss außerdem noch folgendes Modul geladen werden:

```
modprobe mISDN_dsp <ENTER>
```

Damit hätten Sie z.B. Ihre BN4S0 wie folgt konfiguriert:

Port 1	TE,PMP
Port 2	TE,PMP
Port 3	NT,PMP
Port 4	NT,PMP

Die Anzahl und Position der pro Option übergebenen Werte richtet sich nach der Anzahl der Ports und der pro Port gewünschten Konfiguration.

Wenn Sie eine **hfc-pci-Chip** besierte 1 Port-Karte, wie z.B. die **Billion ISDN PCI** haben, dann laden Sie den Treiber folgendermaßen:

```
modprobe hfcpci layermask=0xf protocol=0x2
```

### 3.6 Die Datei misdn.conf

Doch bevor Sie chan\_misdn benutzen können, müssen Sie noch einige Konfigurationen in der Datei "/etc/asterisk/misdn.conf" vornehmen. Die Datei ist in sogenannte Sektionen aufgeteilt. Sektionsnamen sind in eckige Klammern gesetzt:

**[sektionX]**

Hierbei ist zu beachten, daß es immer eine Sektion mit dem Namen „general“ geben sollte. In dieser Sektion können Standardwerte festgelegt werden. Alle folgenden Sektionen können als sogenannte Kanalgruppen betrachtet werden. Innerhalb dieser Sektionen müssen die Ports angegeben werden, die dort Mitglied sein sollen.

Folgende Optionen können gesetzt werden:

**context=demo**

Gibt den zuständigen Kontext in der „extensions.conf“ an.

**language=en**

Legt die Sprache fest.

**immediate=no**

Bestimmt, ob der Ruf sofort nach dem Abheben in den entspr. Kontext bzw. in den Dialplan geleitet wird, oder ob erst ein Freizeichen kommt, und während des Wählens auf ein Timeout gewartet wird.

**debug=0**

Legt das s.g. Debug-Level fest.

**msns=\***

Legt fest auf welche MSN's der oder die Ports reagieren sollen.

**ports=1,2,3,4**

Legt fest welche Ports zu der angegebenen Gruppe (Sektion) gehören sollen.

**hold\_allowed=yes**

Legt fest, ob das Halten eines Gespräches erlaubt ist oder nicht.

**stop\_tone\_after\_first\_digit=yes**

Legt fest, ob nach der ersten Ziffer kein Freizeichen mehr ertönen soll, oder ob das Freizeichen erst abbricht, wenn die Nummer gültig ist, d.h., die Nummer im Dialplan bekannt ist. Standardeinstellung ist „yes“

**append\_digits2exten=yes**

Legt fest, ob nachgewählte Ziffern an die Extension (\${EXTEN}) rangehängen werden.

**tracefile=/tmp/misdn.trace**

Legt fest, in welcher Datei die Debug-Ausgaben gespeichert werden sollen.

Seite / Page Seitennummer von Statistik

**nationalprfeix=0**

**internationalprefix=00**

Legt fest, welches Präfix für nationale und internationale Rufe angehängt wird.

**volume=0**

Legt die Lautstärke des ISDN-Kanals fest. Mögliche Werte sind Ganzzahlen zwischen -8 und +8.

**te\_choose\_channel=0**

Workaround für den Fall, daß die Vermittlungsstelle da Kanalanforderung „ANY\_CHANNEL“ nicht versteht, und eine konkrete Kanalwahl fordert (z.B. in den Niederlanden)

**dialplan=0**

Legt den ISDN Wählplan fest. Möglich Werte:

- 0 - unknown
- 1 - National
- 2 - International
- 4 - Subscriber

**dynamic\_crypt=no**

Zukünftige Option für die generelle Aktivierung oder Deaktivierung des dynamischen Verschlüsseln von ISDN-Gesprächen während des Gesprächs auf Anforderung (crypting on demand).

**crypt\_keys=key1,key2, .... ,keyn**

Array von möglichen Schlüsseln für die ISDN-Verschlüsselung.

**callgroup=1**

**pickupgroup=1**

Legt die Rufgruppe und die Ranholgruppe fest. Analog zu den Optionen in den versch. Anderen Konfigurationsdateien, da Asteriskweit verfügbar.

**Callerid=1234**

Legt die „Callerid“ für den ausgehenden Kanal fest.

Für unsere Beispielkonfiguration könnte die „misdn.conf“ folgendermaßen aussehen:

```
[general]  
context=demo  
language=en  
immediate=no  
debug=0  
msns=*  
  
[TEports]  
context=incoming  
ports=1,2  
msns=5555550,5555599,5555510  
  
[NTports]  
context=outgoing  
ports=3,4
```

In der Sektion "general" sind einige Standardwerte für weitere Sektionen vorgegeben. So ist z.B. Die Option "context" wie bei den anderen Channel-Treibern auf "demo" gestellt. Die Option "immediate" bestimmt, wie auch bei den ZapChannels, ob der Kanal sofort nach dem Abheben in den Kontext springt, ohne das Wählen einer Nummer abzuwarten. "msns" ist hier auf "\*" gestellt, was für weitere Sektionen bedeutet, daß bei allen MSN's reagiert werden soll, wenn nicht wie in Sektion [TEports] etwas anderes angegeben ist. Port 1 und 2 würden auf die dort angegebenen MSN's reagieren. Andersherum könnte ich auf der Gruppe "TEports" mit "Dial(mISDN/g:TEports/\${EXTEN})" herauswählen, und es würde der erste freie Kanal innerhalb dieser Gruppe verwendet werden. Innerhalb der beiden Sektionen [TEports] und [NTports] wird die in "general" vorgegebene Option "context" mit den jeweils dort angegebenen Werten überschrieben. Mit dem vorangegangenen Beispiel hätten wir eine BN4S0 mit zwei TE und zwei NT Ports in der Asterisk über chan\_misdn eingebunden. Wenn alles nach Wunsch konfiguriert ist, kann die Asterisk mit chan\_misdn gestartet werden.

**Für den Betrieb eines NT Ports als Punkt-zu-Punkt (PP) Anschluss muß an der Portnummer noch das Kürzel "ptp" angehängt werden:**

```
ports=3ptp,4
```

Seite / Page Seitennummer von Statistik

### 3.7 chan\_misdn CLI Kommandos

Zusammen mit dem geladenen chan\_misdn Modul, stehe Ihnen auch einige neue CLI Kommandos zur Verfügung:

**misdn show config**

Zeigt die aktuelle Konfiguration aller misdn Ports/Kanäle

**misdn reload**

Befehl zur Rekonfiguration des chan\_misdn Moduls. Wird ebenfalls bei einem globalen „reload“ der Asterisk ausgeführt.

**misdn show channels**

Gibt eine Liste aller aktiven mISDN Kanäle aus.

**misdn show fullstacks**

Zeigt den Zustand der Ports an, und zusätzlich die Kanalzustände.

**misdn show stacks**

Zeigt nur den Zustand der Ports an.

**misdn restart <port no.>**

Startet den angegebenen Port neu.

**misdn toggle echocancel <channel>**

Aktiviert oder Deaktiviert das Echocanceling auf dem angegebenen Kanal.

**misdn set debug <debuglevel>**

Bestimmt die Genauigkeit von Debugausgaben. Folgende Werte sind möglich:

**0 - Kein Debug**

**1 - mISDN Nachrichten und \* - Nachrichten, und \* - Statusänderungen**

**2 - mISDN Nachrichten mit spezifischen Informationen (z.B. bearer capability)**

**3 - sehr gesprächig, Stufe 2 + viele zusätzliche Treiberinformationen**

**4 - noch gesprächiger als Stufe 3**

**misdn send display <channel> <msg>**

Sendet eine s.g. Display-Nachricht zum angegebenen "channel". Wenn das dort angeschlossenen Endgerät dazu in der Lage ist, wird diese Nachricht auf der Anzeige des Endgerätes angezeigt.

### 3.8 mISDN Dial-Parameter

Der "Dialstring" für chan\_isdn in der "extensions.conf" sieht folgendermaßen aus:

**Dial(misdn/<portno>/<extension>[/<OPTIONSSTRING>])**

oder Gruppe:

**Dial(misdn/g:<groupname>/<extension>[/<OPTIONSSTRING>])**

**Beispiel:**

**Dial(misdn/1/123456789)**

**oder Gruppe:**

**Dial(misdn/g:TEPorts/123456789)**

Mit mISDN haben Sie die Möglichkeit beim Rufaufbau verschiedene Optionen zu übergeben.

Der "OPTIONSTRING" setzt sich wie folgt zusammen:

**<optchar1><OptParam1>:<optchar2><OptParam2>**

Beispiel:

**Dial(misdn/g:TEports/123456789/n:h)**

Baut einen digitalen Ruf ohne DTMF-Erkennung auf.

Mögliche Optionen sind:

- d** – Sendet eine s.g "Displaymessage" auf einen ISDN Kanal, wobei der übergebene Parameter den darzustellenden Text angibt
- n** – deaktivierung DTMF-Tonerkennung auf ausgehendem Kanal
- h** – Baut einen digitalen Kanal auf (DfÜ)
- c** – initiiert ein verschlüsseltes, ausgehendes Gespräch, wobei der übergebene Parameter den zu verwendenden Schlüsselindex aus dem Schlüsselarray in der misdn.conf (crypt\_keys= ....[Index=1],.....[Index=2]) angibt
- e** – aktiviert oder deaktiviert die Echocanceling auf dem ausgehenden Kanal

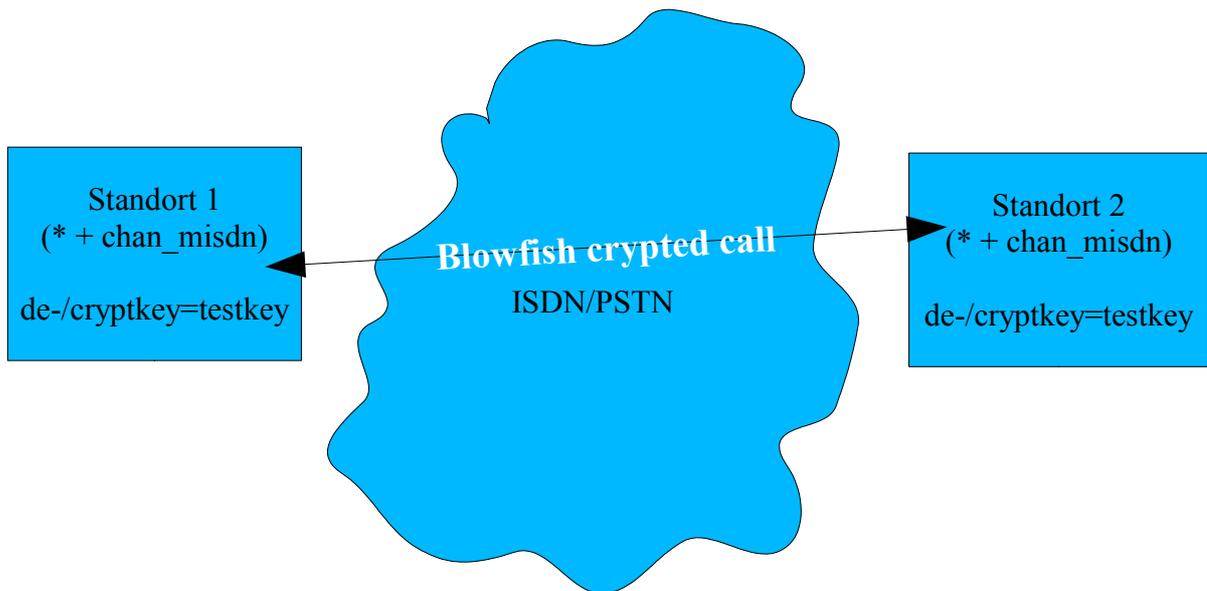
Wenn diese Optionen mit der "Dial"-Applikation übergeben werden, beziehen sie sich immer auf den ausgehenden Kanal. Wenn Sie die Optionen auf den ankommenden Kanal setzen möchten, benutzen Sie bitte die Applikation "misdn\_set\_opt" innerhalb des Wählplans:

Beispiel:

**exten => 1234,1,misdn\_set\_opt(<opt><optarg>:<opt><optarg>..)**

### 3.9 Verschlüsselung einer ISDN-Verbindung

Nehmen wir an, Sie haben zwei Standorte, die jeweils über ISDN-Anschlüsse erreichbar sind. Die Telefongespräche zwischen diesen Standorten handeln häufig von brisanten Themen oder Firmengeheimnissen. Wie jeder weiß, sind bestimmte Organisationen dazu in der Lage diese Gespräche abzuhören, aufzuzeichnen und die darüber gelaufenen Informationen entsprechend zu verwerten. Dagegen können Sie sich mit einer Verschlüsselung Ihrer Gespräche schützen, indem Sie an beiden Standorten eine Asterisk PBX mit chan\_misdn installieren. Das Szenario sähe dann folgendermaßen aus:



Für das Verschlüsseln eines erfordert einen Schlüssel, der auf beiden Seiten bekannt ist. Diese Schlüssel können in der misdn.conf abgelegt werden:

**crypt\_keys=key1,key2,.....,keyn**

Um ein verschlüsseltes Gespräch abzusetzen, wählen mit „Dial“ folgendermaßen:

**[outgoing-context-location1]**

**exten => \_X.,1,Dial(misdn/g:TEports/\${EXTEN}/c1)**

Es wird ein verschlüsseltes Gespräch abgesetzt welches als Schlüssel den Ersten Schlüssel aus der „crypt\_keys“-Liste benutzen soll.

Auf der Gegenseite muss das Gespräch dann folgendermassen angenommen werden:

**[incoming-context-location2]**

**exten => 12345,1,misdn\_set\_opt(c2)**

**exten => 12345,2,Dial(SIP/12345)**

Das von Standort1 eingehende Gespräch soll hier Decryptiert werden, und zwar mit den an zweiter Stellen stehenden Schlüssel aus der „crypt\_keys“-Liste welcher sich mit dem Schlüssel in Standort1 decken muss.

Sind alle Gespräche zwischen diesen beiden Standorten verschlüsselt.

### **3.10 Einige Worte zum Betrieb von BNxS0 Karten mit dem „bristuff“-Treiber von Junghanns.net**

Es kommt immerwieder die Frage auf, ob die beroNet Karten kompatibel mit den Junghanns-Karten sind. Die Antwort auf diese Frage ist "jein". Die Karten sind zwar völlig baugleich, jedoch gibt es einen kleinen Unterschied, und das ist die s.g. Vendor ID die auf der Karte gespeichert ist. In der Implementation des "bri-stuff" Treibers, macht Herr Junghanns unterschiede zwischen seinen Karten und anderen. Deshalb kann es sein, dass sich der "bri-stuff" Treiber mit anderen Karten als denen von Junghanns.net nicht laden lässt. Da der "bri-stuff" Treiber jedoch, wie fast alle anderen Treiber, der GPL unterliegt, steht es hier jedem frei entsprechende Änderung in den "bri-stuff" Quellen vorzunehmen. Bei unserem Treiber gibt es eine solche Unterscheidung nicht, deshalb sollten die Karten von Junghanns.net auch mit chan\_misdn zusammenarbeiten. Bitte haben Sie Verständnis dafür, das wir für den Betrieb unserer Karten zusammen mit dem "bri-stuff" Treiber keinen 100%-igen Support bieten können.

#### 4. Installieren von Asterisk

Wie schon mehrmals erwähnt setzen die vorangegangenen Punkte alle voraus, daß die Asterisk kompiliert und installiert auf Ihrem System ist. Falls Sie nicht wissen wie das geht, wird es in diesem Punkt noch einmal kurz erklärt.

Auch hier werden wir die Asterisk Quellen aus dem CVS-Server von Digium holen, und selbst kompilieren. Alternativ könnte man in einem Debian basierten System auch folgendes eingeben:

**apt-get install asterisk <ENTER>**

Das entbindet Sie jedoch nicht davon je nach Arte Ihrer erworbenen Hardware die entsprechenden Kernelmodule zu installieren, da diese Module immer für den laufenden Kernel kompiliert sein müssen.

Wenn Sie in Punkt 2.1. schon die Quellen geholt haben, überspringen Sie die nächsten drei Eingaben.

Um die Quellen aus dem CVS zu holen, gehen Sie in das Verzeichnis „/usr/src/“, und geben Sie folgendes ein:

**export CVSROOT:pserver:cvcs.digium.com:/usr/cvsroot <ENTER>**  
**cvs login <ENTER>**

Geben Sie das Passwort ein: „anoncvs“

**cvs co asterisk**

Der Download wird gestartet. Wenn er beendet ist, gehen Sie in das Verzeichnis „/usr/src/asterisk/“, und geben Sie folgendes ein:

**make install <ENTER>**

Wenn alles fehlerfrei verlaufen ist, ist Asterisk auf Ihrem System installiert. Um die Beispielkonfigurationen zu installieren geben Sie folgendes ein:

**make samples <ENTER>**

Dabei werden eventuell schon vorhandene Konfigurationsdateien nicht überschrieben. Wenn Sie Ihre Interfacekarten schon konfiguriert haben, sollte sich die Asterisk starten lassen:

**asterisk -c <ENTER>**

Anschliessend befinden Sie sich im Kommandozeileninterface der Asterisk.

## 5. Weitere Informationsquellen

[www.digium.com](http://www.digium.com)

Webseite von Digium Inc., Sponsor des Asterisk Projektes

[www.beronet.com](http://www.beronet.com) [Webseite beroNet GmbH](#)

Download chan\_misdn

[www.asterisk.org](http://www.asterisk.org)

Offizielle Projektseite von Asterisk

[www.voip-info.org](http://www.voip-info.org)

Sehr umfangreiches Infoportal zum Thema VoIP und speziell auch für Asterisk

[www.ip-phone-forum.de](http://www.ip-phone-forum.de)

Deutschsprachiges Forum zum Thema VoIP mit eigenem Asterisk-Bereich

[www.isdn4linux.de](http://www.isdn4linux.de)

Projektseite von mISDN

[www.jolly.de](http://www.jolly.de)

Webseite von Andreas Eversberg, Entwickler von PBX4Linux

[www.freshmeat.net](http://www.freshmeat.net)

Portal und Hosting für open source Projekte

## 6. Beispielkonfigurationen

Die Beispielkonfigurationen können Sie auf unserer Webseite unter [www.beronet.com](http://www.beronet.com) herunterladen. Es sind folgende Beispielkonfigurationen vorhanden:

- [TDM11B](#)  
[http://www.beronet.com/download/config\\_samples/conf\\_TDM11B.zip](http://www.beronet.com/download/config_samples/conf_TDM11B.zip)
- [TDM40B](#)  
[http://www.beronet.com/download/config\\_samples/conf\\_TDM40B.zip](http://www.beronet.com/download/config_samples/conf_TDM40B.zip)
- [TDM04B](#)  
[http://www.beronet.com/download/config\\_samples/conf\\_TDM04B.zip](http://www.beronet.com/download/config_samples/conf_TDM04B.zip)
- [TE110P](#)  
[http://www.beronet.com/download/config\\_samples/conf\\_TE110P.zip](http://www.beronet.com/download/config_samples/conf_TE110P.zip)
- [TE4xxP](#)  
[http://www.beronet.com/download/config\\_samples/conf\\_TE4xxP.zip](http://www.beronet.com/download/config_samples/conf_TE4xxP.zip)
- [BN4S0](#)  
[http://www.beronet.com/download/config\\_samples/conf\\_BN4S0.zip](http://www.beronet.com/download/config_samples/conf_BN4S0.zip)
- [BN8S0](#)  
[http://www.beronet.com/download/config\\_samples/conf\\_BN8S0.zip](http://www.beronet.com/download/config_samples/conf_BN8S0.zip)

## 7. E1 Kabelbelegungen für den Anschluss an PRI oder andere Telefonanlagen/Switches

Pinbelegung Portseitig (Kartenseitig):

Übliche Anschlussweise zwischen einer Digium PRI Karte und einer anderen TK-Anlage:

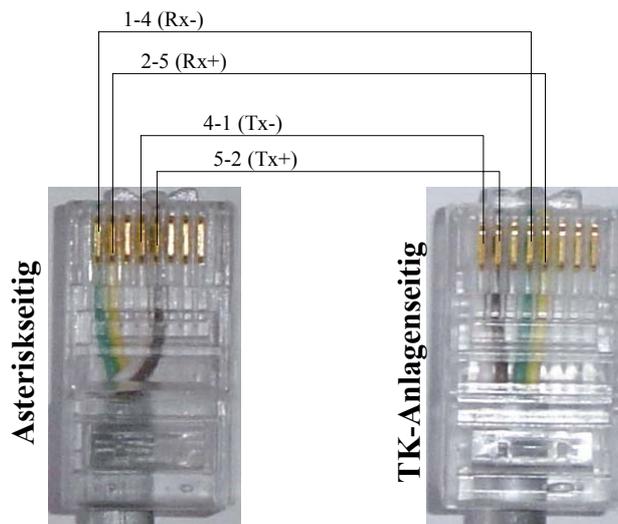


Abb 7.1.

Beim Anschluss eines NTPM kann es sein, dass die Adern schon innerhalb des NTPM gekreuzt sind, und deshalb können Sie dann ein einfaches Patch-Kabel (Adern 1 zu 1) nehmen.

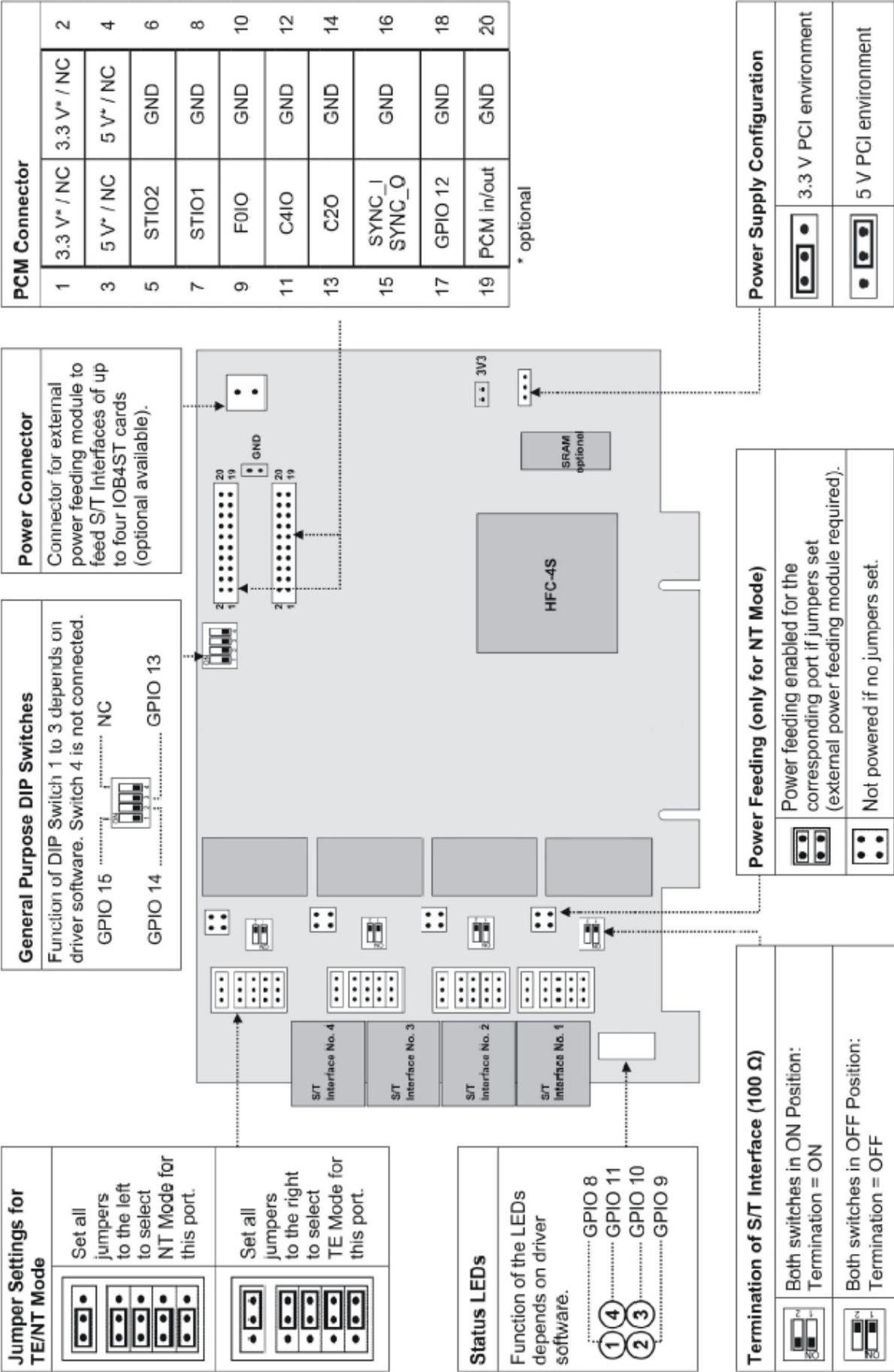
Da es bei der Belegung der Adernpaare eines PRI Anschlusses keinen Standard gibt, weichen einige Hersteller von dem oben gezeigten ab.

Beim Anschluss an LSA-Leisten Ihres Providers, ist die Belegung abhängig vom Installateur, d.h., Sie müssten die Belegung bei Ihrem Provider erfragen, oder durch Ausprobieren rauskriegen. Das gleiche gilt für manche TK-Anlagen.

## 8. Technische Daten

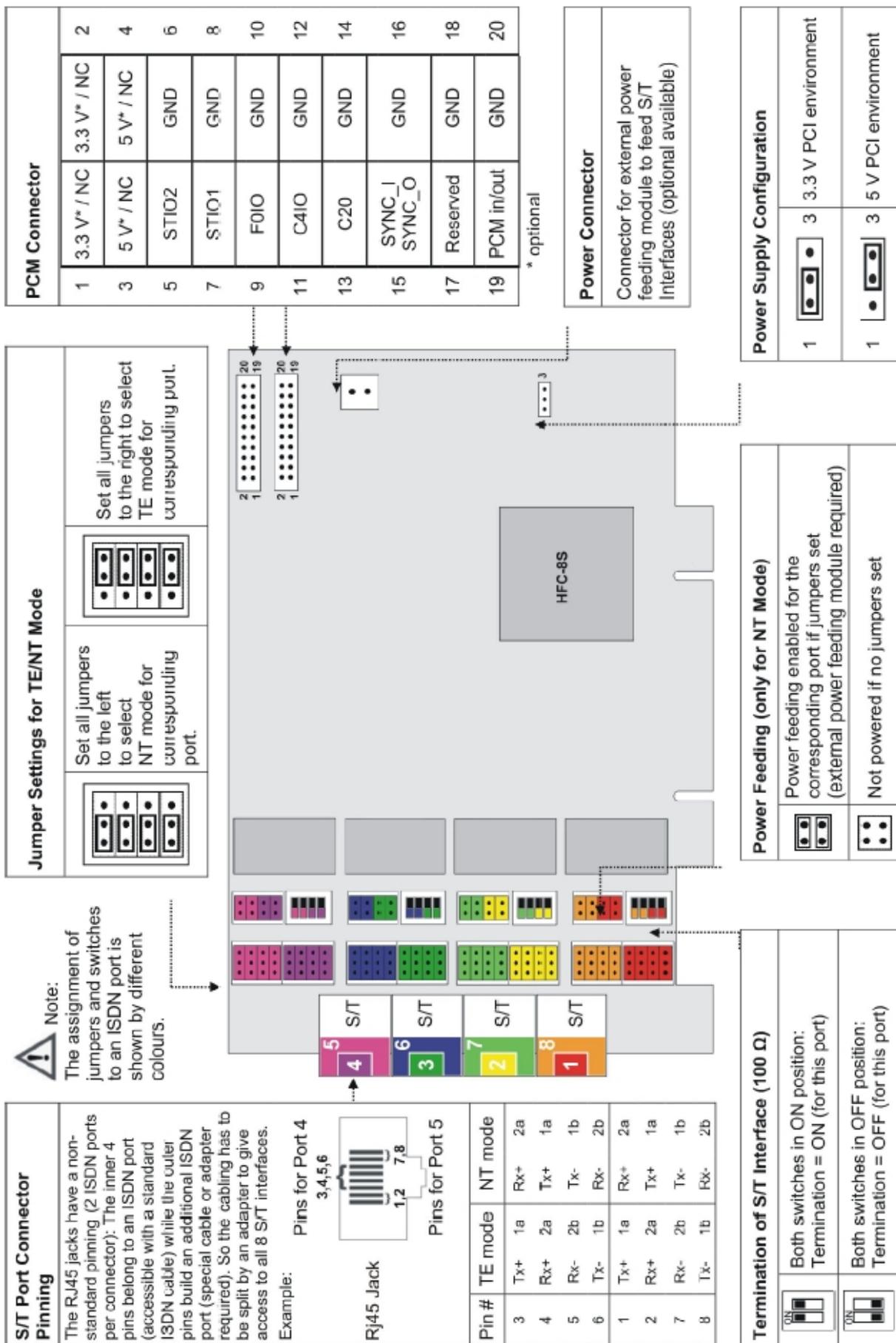
### 8.1 BN4S0 Board

- **S/T Interface**  
4 BRI ports (TE/NT mode; feeding by external power supply unit; line termination (100 Ohm) selectable)  
short circuit protection for the ISDN cabling by fuses
- **PCI Interface**  
PCI interface is suitable for 3.3V as well as 5V PCI 2.2 slots
- **PCM Bus**  
daisy chaining possible by two connectors  
2/4/8 Mbit/s data transfer rate
- **Chipset**  
Cologne Chip HFC-4S ISDN IC  
high precision 49.152 MHz quartz oscillator  
512 bit x 8 serial EEPROM (PCI configuration information)
- **General Purpose I/O**  
4 dual LEDs at the slot bracket (states *green, red, off*)  
3-DIP switch can be checked by software
- **ISDN Conformity**  
The ISDN OEM board family is compliant to the hardware specifications of the ISDN standards (I.430, CTR3).
- **Board Dimensions**  
15.5 x 10.5 x 1.3 (cm)



## 8.2 BN8S0 Board

- **S/T Interface**  
8 S/T interfaces: 4 RJ45 jacks with a non-standard pinning (2 ISDN ports per connector); TE/NT mode; feeding by external power supply unit; line termination (100 Ω) selectable)  
short circuit protection for the ISDN cabling by fuses
- **PCI Interface**  
PCI interface is suitable for 3.3V as well as 5V PCI 2.2 slots
- **PCM Bus**  
daisy chaining possible by two connectors  
2/4/8 Mbit/s data transfer rate
- **Chipset**  
Cologne Chip HFC-8S ISDN IC  
high precision 49.152 MHz quartz oscillator  
512 bit x 8 serial EEPROM (PCI configuration information)
- **ISDN Conformity**  
The BN4S0 board is compliant to the hardware specifications of the ISDN standards (I.430, CTR3)
- **Board Dimensions**  
16 x 10.5 x 2 cm (the standard PCI dimensions are exceeded)



### 8.3 TE410P / TE405P (associated with Asterisk)

- **Applications**

- Legacy PBX/IVR services
- Voice-over Internet Protocol (VoIP) services
- Complex IVR Trees
- "Meet-me-Bridge" conferencing
- VoIP Gateways (supports SIP, H.323 and IAX)
- Calling Card Platforms
- Voice/Data Router (replace expensive routers)

- **PRI Switch Compatibility**

- EuroISDN (PRI or PRA) — Q.931/Q.921
- AT&T 4ESS
- DMS 100
- Lucent 5E
- Network or CPE
- National ISDN 2

- **CAS Voice Modes**

- Feature Group D
- E&M Wink
- A-Law, Mu-Law, and Linear Modes Supported

- **Data Modes**

- SyncPPP (both Fixed and Dialup)
- Frame Relay
- Cisco HDLC
- Multi-link PPP

- **PCI**

- The TE410P is for use only with a 3.3 volt PCI slot.
- The TE405P is for use only with a „normal“ 5 volt PCI slot.

## 8.4 TDMxxP / TDM400P (associated with Asterisk)

- **Target Applications**
  - Small Office Home Office (SOHO) applications
  - Gateway Termination to Analog Telephones
  - Add Inexpensive Analog Phones to Existing PBXs
  - Wireless Point-to-Point Applications between Asterisk Servers
- **Services and Features**
  - Caller ID and Call Waiting Caller ID
  - ADSI Telephones
  - PCI Half-length Slot
  - RJ-11C Connector
- **Environment Conditions**
  - Operation Range: 0° to 50°C, 32° to 122° F
  - Storage Range: -20° to 65°C, 4° to 149° F
  - Humidity: 10-90% non-condensing
- **Standard Configurations**
  - TDM10B: 1-port FXS bundle
  - TDM40B: 4-port FXS bundle
  - TDM01B: 1-port FXO bundle
  - TDM04B: 4-port FXO bundle
  - TDM11B: 1-port FXS &1-port FXO bundle
  - TDM22B: 2-port FXS &2-port FXO bundle
  - TDM31B: 3-port FXS &1-port FXO bundle
  - \*Other configurations available on request
- **Hardware and Software Requirements**
  - 500-Mhz Pentium III or better with 64MB RAM
  - Available PCI Slot

## 8.5 TE110P (associated with Asterisk)

- **Target Applications**
  - Packet Voice Gateways and Switches
  - Calling Card Services
  - One Number Services
  - Message Services
  - Conferencing
  - Customized and Web Telephony
  - Voice/Data Integration
  - Futureproof PBX
  - ISDN Remote Access Servers
- **PRI Switch Capability**
  - AT&T 4ESS
  - DMS 100
  - Lucent 5E
  - National ISDN 2
  - Euro ISDN
  - Network or CPE
  - NFAS
- **RBS Voice Modes**
  - GR-303
  - A-Law,  $\mu$ -Law, and Linear
  - Modes Supported
  - E&M
  - E&M Wink
  - Feature Group D
  - Groundstart (FXO and FXS)
  - Loopstart (FXO and FXS) with
  - Optional Disconnect Supervision
- **Data Modes**
  - SyncPPP (both Fixed and Dialup)
  - Frame Relay
  - Cisco HDLC

## 8.6 S100I (IAXy) (associated with Asterisk)

- **Applications**

- Internet Telephony Service Provider
- Remote PBX Extensions
- Wireless Phone Service with External Bridge
- Features
- Auto Upgrade
- Caller ID
- Call Waiting
- Cancel Call Waiting (\*70)
- Caller ID on Call Waiting
- Caller ID Disable, Enable (\*67, \*82)
- Three-way Calling
- Call Transfer
- Blind Transfer
- Call Parking
- VMWI (Voice Mail Waiting Indicator)
- Mute Rx on-Hook
- Pulse Dial
- Call Hold

- **VoIP Codecs**

- ulaw (G.711)
- ADPCM

- **VoIP Protocol**

- Inter-Asterisk eXchange (IAX)

- **Telephone**

- Ringer Equivalence Number: 5

- **Power Requirements**

- 9V DC, 1500mA Regulated
- Tip Positive 5.0mm outer-diameter 2.5mm inner-diameter connector

- **Environment Conditions**

- Operation Range: 0° to 50° C, 32° to 122° F
- Storage Range: -20° to 65° C, 4° to 149° F
- Humidity: 10-90% non-condensing