
Debian Sarge für CAPI einrichten

Dipl. Volkswirt. Lukas Mensinck, Mensinck Consulting

<mensinck@mensinck.de>

2004-11-16

Version v.0.1	Versionsgeschichte 2004-07-19	LM
Version v.0.2	Work in Progress 2004-08-18	LM
Version v.0.3	Work in Progress 2004-11-16	LM
	Korrektur kleiner aber feiner Tippfehler Dank Hinweis	

Debian ist nach der Installation nicht standartmäßig für die Benutzung von CAPI eingerichtet. Die Lizenzen der Treiber z.B. von AVM verhindern dieses.

Wenn man sich mit dem Einsatz von *binary only* Treibern anfreunden kann, ist es ohne Probleme möglich, auch Debian für den Einsatz der CAPI-Treiber einzurichten.

Mit diesen kann dann ein Dialin-, Fax- oder auch GSM-Server betrieben werden.

Inhaltsverzeichnis

1. Voraussetzungen	1
2. Benötigte Pakete	2
3. Das Ganze zusammenbauen	2
3.1. Kernel bauen	2
3.2. CAPI Treiber für die AVM Fritzcard	5
3.3. Weite CAPI Pakete installieren	7
4. Konfiguration der CAPI Schnittstellen	7
4.1. Die ISDN Karte einrichten	7
4.2. Modul Konfiguration anpassen	8
4.3. Einrichtung der GSM/v110 Einwahl	8
5. Clients einrichten	10
5.1. PDA einrichten	11
5.2. Hany einrichten	12
5.3. Weitere Software	12

1. Voraussetzungen

Zu allererst muss eine funktionierende Debian Installation vorhanden sein sowie die üblichen Entwicklerpakete sollten auf dem System installiert worden sein.

Außerdem werden die Kernel Quellen benötigt, da der Kernel neu übersetzt werden muß. Folgende Pakete werden also mindestens benötigt.

Entwicklerpakete usw.



Wichtig

gcc, make, automake, autoconf usw.

Und natürlich auch:

- libstdc++-devel

2. Benötigte Pakete

Zunächst sind natürlich die Kernel Quellen für den entsprechenden Kernel zu besorgen. Für das hier verwendete System sind es die Quellen für den 2.4.26er Kernel.

```
[user@host:~/dir]$ apt-get install kernel-source-2.4.26
... Einige Hinweise, z.B. auf zusätzlich benötigte Pakete
```

3. Das Ganze zusammenbauen

3.1. Kernel bauen

Diese Anleitung geht nicht tiefer auf die Kernel-Konfiguration sowie die Erstellung eines neuen Kernels unter Linux bzw. Debian/GNU-Linux ein. Im Internet sind ausreichend Anleitungen verfügbar, in denen man die Details und Problemlösungen finden kann. Das hier beschriebene Vorgehen soll lediglich einen Einstieg ermöglichen, den Kernel für CAPI anzupassen und die entsprechenden Module zu erzeugen. Für weitere Fragen wird ausdrücklich auf weiterführende Literatur verwiesen.

- Kernel auspacken

```
[user@host:~/usr/src]$ tar -xvzf kernel-source-2.4.26.tar.bz2
... Jede Menge Text
```

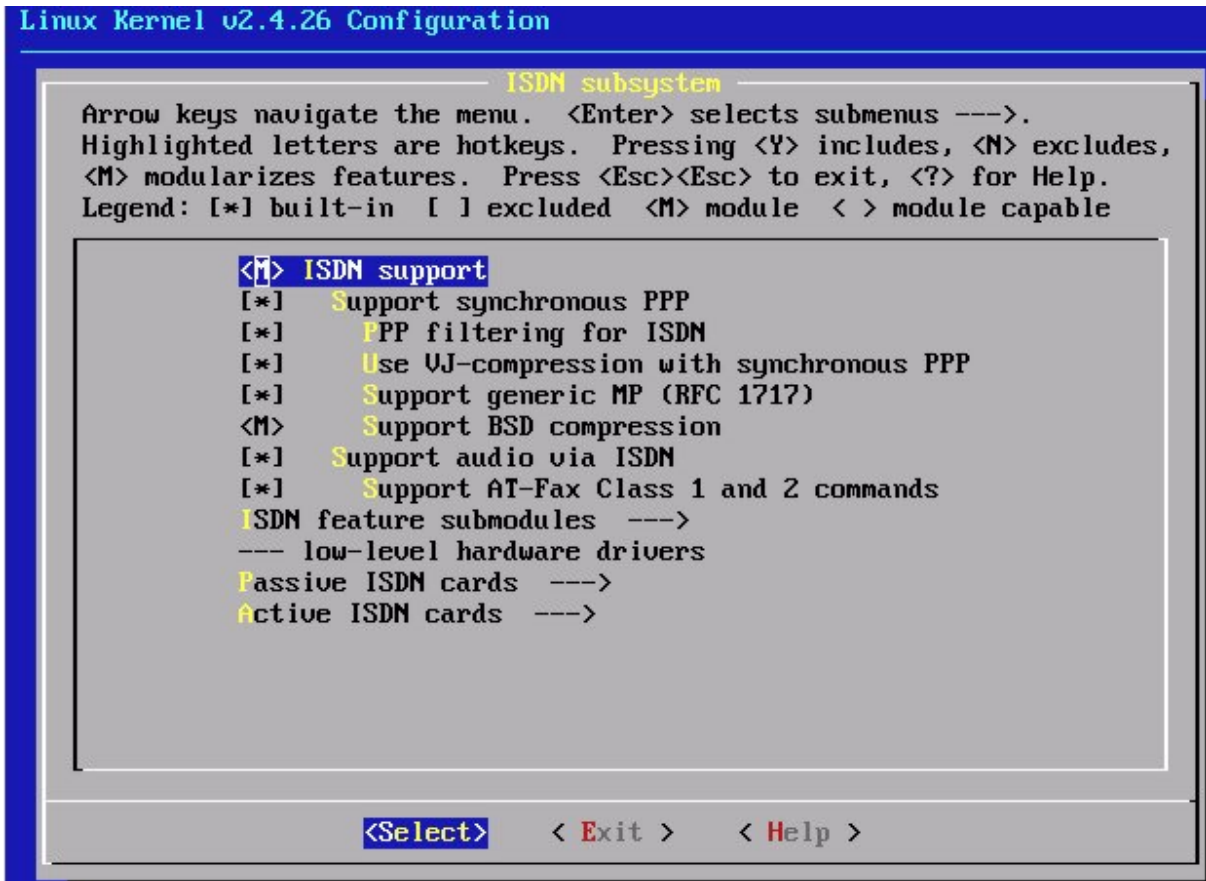
- Kernel link setzen

```
[user@host:~/usr/src]$ ln -s kernel-sources-2.4.26 linux
[user@host:~/usr/src]$ cd linux
```

- Kernel konfigurieren

```
[user@host:/usr/src/linux/]$ cp ./boot/yourconfig .config
[user@host:../linux/]$ make oldconfig
[user@host:../linux/]$ make menuconfig
```

Hier die Einstellungen für die ISDN Konfiguration:



Und hier nun die Konfiguration für die ISDN active Cards

```

Linux Kernel v2.4.26 Configuration

                Active ISDN cards

Arrow keys navigate the menu. <Enter> selects submenus --->.
Highlighted letters are hotkeys. Pressing <Y> includes, <N> excludes,
<M> modularizes features. Press <Esc><Esc> to exit, <?> for Help.
Legend: [*] built-in [ ] excluded <M> module < > module capable

< > Auvertech TurboPAM support
<M> CAPI2.0 support
[*]  Verbose reason code reporting (kernel size +=7K)
[*]  CAPI2.0 Middleware support (EXPERIMENTAL)
<M>  CAPI2.0 /dev/capi support
[*]  CAPI2.0 filesystem support
<M>  CAPI2.0 capidrv interface support
<M>  AVM B1 ISA support
<M>  AVM B1 PCI support
[*]  AVM B1 PCI V4 support
<M>  AVM T1/T1-B ISA support
<M>  AVM B1/M1/M2 PCMCIA support
<M>  AVM B1/M1/M2 PCMCIA cs module
<M>  AVM T1/T1-B PCI support
<M>  AVM C4/C2 support
< > Hypercope HYSDM cards (Champ, Ergo, Metro) support (module only

                <Select>    < Exit >    < Help >

```

Nachdem die Einstellungen entsprechend vorgenommen sind, kann die Kernel-Konfiguration gespeichert werden.

- Kernel Abhängigkeiten bauen

```
[user@host:~/linux]$ make dep
```

- kernel mit kernelpackage bauen

```
[user@host:~/linux]$ su
[host:~/linux]$ make-kpkg --revision=custom_01 --initrd binary
```



Warnung

Wenn der Kernel mit inird Support gebaut werden soll, muss -- *initrd* beim Aufruf mit angegeben werden, sonst kann der neue Kernel später nicht gebootet werden.

Wenn der Kernel fertig gebaut ist, existieren im Verzeichnis `/usr/src/` die zum Installieren fertigen Kernel-Pakete.

```
[host:/usr/src/linux/]$ls -l ../kernel*
kernel-image-2.4.26_custom_01_i386.deb
kernel-headers-2.4.26_custom_01_i386.deb
```

- Den neuen Kernel installieren

Auf dem verwendeten System wurde der Bootmanager **grub** eingesetzt. Daher kann das neu erstellte Kernel-Paket den Bootmanager selbstständig anpassen. Für die Installation des neuen Kernels reicht daher nun ein:

```
[host:/usr/src]$ dpkg -i kernel-image-2.4.26_custom01_i386.deb
```

um den neuen Kernel zu installieren.

Nachdem **dpkg** seine Arbeit getan hat, kann der Rechner neu gestartet werden und der neue Kernel hierbei ausgewählt werden.

Sofern der Rechner mit dem neuen Kernel brav bootet - schön, die schweigste Arbeit ist schon getan.

3.2. CAPI Treiber für die AVM Fritzcard



Anmerkung

An dieser Stelle werden nun die Treiber für die ISDN Karte benötigt. Da die AVM Treiber teilweise nur als *binary only* Treiber verfügbar sind, wird hier die Debian Policy gebrochen. Man sollte sich darüber im klaren sein und sich mit dem Einsatz von solchen Treiber abfinden können.

Wenn dieses nicht der Fall ist, kann man die HISAX Treiber verwenden. Mit diesen kann aber dann nicht gefaxt werden. Alle übrigen ISDN Features sind aber vorhanden.

AVM vertreibt die Treiber für seine Karten auf seinen Webseiten. Vorgefertigte Binaries sind allerdings nur für SuSE™ Distributionen vorhanden. Man muss sich die Treiber also selber bauen. Hierzu sind zunächst die entsprechenden Sourcen zu besorgen.

```
[user@host:/usr/src]$ wget ftp://ftp.avm.de/cardware/fritzcrd/linux \
/suse.82/fcclassic-suse-8.2-03.11.02.tar.gz
```

Nun sind die Sourcen auszupacken, anzupassen und das entsprechende Kernel-Modul zu bauen.



Wichtig

Wenn der 2.6er Kernel benutzt werden soll, sind die Quellen für SuSE 9 zu wählen. Allerdings kann ich keine Hinweise für die Installation für den 2.6er Kernel geben.

- Auspacken

```
[user@host:/usr/src]$ tar -xvzf
fcclassic-suse-8.2-03.11.02.tar.gz
[user@host:/usr/src/]$ cd fritz
```

- Anpassen der Sourcen

Die Original AVM Sourcen lassen sich unter Debian nicht direkt kompilieren. Daher ist das Makefile anzupassen. Es sind folgende Änderungen nötig:1

1Dank an <drbob@voip-info.org>. Eine spezielle Anleitung hierzu ist auf der Webseite www.voip-info.org (<http://www.voip-info.org>) zu finden

- Änderungen in `./src.drv/makefile`. Hier ist zunächst der Pfad zu den installierten Kernel-Headern/Includes zu prüfen und die Zeile mit `KRNLINCL` korrekt zu setzen. Meist ist folgende Zeile zielführend.

```
KRNLINCL=/lib/modules/`uname -r`/build/include
```

Nun muss folgendes ersetzt werden.

```
DEFINES = -DMODULE -D__KERNEL__ -DNDEBUG \  
-D__$(CARD)__ -DTARGET=\"$(CARD)\"
```

durch:

```
DEFINES = -DMODULE -DMODVERSIONS -D__KERNEL__ -DNDEBUG \  
-D__$CARD__ -DTARGET=\"$(CARD)\"
```

Außerdem ist noch folgende Zeile anzupassen:

```
CCFLAGS = -c $(DEFINES) -O2 -Wall -I $(KRNLINCL)
```

mit folgenden Erweiterungen:

```
CCFLAGS = -c $(DEFINES) -march YourArch (z.B. i586) -O2 \  
-Wall -I $(KRNLINCL) -include $(KRNLINCL)/linux/modversions.h
```

Hierbei ist darauf zu achten, daß die korrekte Architektur für das Zielsystem angegeben wird. Z.B `i586` für einen Pentium Rechner.

- Anpassungen in der Datei `src.drc/defs.h`

Folgende Zeile ist zu suchen und wie hier angegeben anzupassen.

```
#if LINUX_VERSION_CODE < KERNEL_VERSION(2,5,0)
```

ist zu ersetzen durch:

```
#if LINUX_VERSION_CODE < KERNEL_VERSION(2,4,26)
```

- Modul kompilieren

So, das waren die nötigen Anpassungen. Nun kann das Modul gebaut werden. Los gehts:

```
[user@host:/usr/src/fritz]$ make
```

Einige Warnungen, die erscheinen, können beruhigt ignoriert werden.

- Modul installieren

```
[user@host:/usr/src/fritz]$su
[host:/usr/src/fritz]$ mkdir /lib/modules/`uname -r`/misc
[host:/usr/src/fritz]$ cp src.drv/fcclassic.o /lib/modules/`uname -r`/misc/
[host:/usr/src/fritz]$ depmod -a
```

Nun sollte das Modul einsatzbereit sein.

- Modul testen

Nun ist es so weit, das Modul zu testen. Hierzu soll das capi-Modul sowie das Treibermodul geladen werden.

```
[host:/usr/src/fritz]$ modprobe capi
[host:/usr/src/fritz]$ modprobe fcclassic
```

Wobei das Laden des `fcclassic` Moduls einige Meldungen über die Verletzung der GLP ausgeben wird.

3.3. Weite CAPI Pakete installieren

Wie üblich bei Debian, werden auch die weiteren CAPI-Pakete mit apt (oder einer der vielen Alternativen) installiert

pppd	Na klar, um überhaupt eine ppp-Verbindung herstellen zu können
capi4hylafax	Wenn Hylafax eingesetzt werden soll, und nur dann macht der Einsatz von CAPI eigentlich Sinn, wird dieses Paket benötigt.
	test
pppdcapiplugin	Dieses Paket stellt die Möglichkeit zur Verfügung, mit dem pppd auch über die CAPI Schnittstelle Verbindungen aufzubauen.
isdnutils	Viele nette Programme für ISDN-Leitungen. So zb. eine Überwachung der ISDN-Leitung mit Anruferlisten usw. mit dem Programm isdnlog.
isdnactivecards	Tools für Aktive ISDN-Karten und zu der haben wir unsere schöne alte Fritzcard ja gerade <i>virtuell</i> gemacht.

Also nun zur Installation des Ganzen:

```
[host:/usr/src]$ apt-get install isdnactivecards \
    pppd \
    pppdcapiplugin \
    capi4hylafax \
    isdnutils
Jede Menge Meldungen. Eventuell auch über zusätzlich zu
installierende Pakete wie Hylafax...
```

4. Konfiguration der CAPI Schnittstellen

4.1. Die ISDN Karte einrichten

Um die Hardware der CAPI Karte einzurichten ist unter Debian die Datei `/etc/isdn/capi.conf` den lokalen

Gegebenheiten entsprechend anzupassen. Die Datei wurde durch das Paket `isdnactivecards` installiert und in der default Version sind alle verfügbaren aktiven Karten Aktiviert. Diese sind zunächst zu aktivieren und ein Eintrag für die eigene "passive" Karte ist hinzuzufügen. Bei den ISA Karten muß dabei der IO-Port sowie der Interrupt angegeben werden, die je nach Karteneinstellungen variieren können. Hier ein Beispiel:

```
# card      file      proto    io      irg      mem      cardnr   options
# blisa     b1.t4    DSS1    OX150  7        -        -        p2p
# blpci     b1.t4    DSS1    -       -        -        -
...
# fcpci     -        -        -       -        -        -
fcclassic  -        -        0x200  5        -        -
```

Zeilen, die ein # Zeichen voran gestellt haben, sind auskommentiert. In der Originalfassung der Datei sind die aktiven Karten aktiviert, so daß das Startscript sie alle durchprobiert. Um den Startvorgang zu beschleunigen, sollten alle nicht verwendeten Karten auskommentiert sein. Da hier eine AVM Fritzcard Classic, also eine ISA-Karte, verwendet wird, ist die letzte Zeile einzufügen und die entsprechenden Parameter für IO-Port und IRQ einzustellen. Für eine PCI Karte würde der Eintrag in der vorletzten Zeile gelten.

4.2. Modul Konfiguration anpassen

Da wir statt der standardmäßig vorgesehenen HISAX Treiber hier jetzt die CAPI Treiber einsetzen, muss auch noch die Datei `/etc/modules.conf` angepaßt werden. Folgende Einträge sind nötig:



Warnung

Aber Achtung, wenn das Paket **modutils** eingesetzt wird, was auf den meisten Debian Systemen der Fall ist, wird die Datei `/etc/modules.conf` automatisch erzeugt und Änderungen an ihr daher überschrieben. Man prüfe also, ob `modutils` verwendet wird.

Wenn Das Paket verwendet wird, muß statt `/etc/modules.conf` die Datei `/etc/modutils/aliases` entsprechend bearbeitet werden.

Also, hier kommenen die wichtigen Einstellungen, damit auch die anderen ISDN Programme, wie z.B. `isdnlog` weiterhin verwendet werden können.

```
alias char-majro-43    capidrv
alias char-major-44    capidrv
alias char-major-45    capidrv
```

Nachdem diese Änderungen durchgeführt wurden und mit einem `depmod -a` dem System bekannt gemacht wurden, werden für ISDN-Verbindungen fortan die CAPI-Treiber verwendet.

4.3. Einrichtung der GSM/v110 Einwahl

4.3.1. Konfiguration des pppd

Um sich mit dem Handy auf dem Server einwählen zu können, muß eine V110-Einwahl eingerichtet werden. Hierzu wird der `pppd` Server mit dem **pppdcapiplugin** verwendet werden.

Am einfachsten gestaltet sich die Installation, wenn ein `call` script für den **pppd** erstellt wird. Dieses ist in das Verzeichnis `/etc/ppp/peers` zu speichern und dem `pppd` beim Start zu übergeben.

Für die Einwahl wird folgende Konfigurationsdatei verwendet

```
logfile /var/log/pppd_gsm.log
# debug
name isdnin
auth
login
require-pap
plugin capiplugin.so
inmsn yourMSN
protocol v110async
localIP:remoteIP
ms-dns DSN-Server-IP
nobsdcomp
noccp
nodefaultroute
nodeflate
noipx
novj
```

1. **name**

steht für den Namen des Systems und dient der korrekten Identifikation in der Datei `pap-secrets`. Außerdem wird der Name an den Partner (hier client) gesendet, wenn der Client eine Authentifizierung verlangen sollte.

2. **auth und login**

Diese Parameter dienen dazu, dem pppd mitzuteilen, daß der Client sich authentifizieren soll und die übertragenen Passwörter in der `passwd` Datei nachgeschlagen werden. Mit einem entsprechendem Eintrag in der Datei `pap-secrets` kann das System so eingestellt werden, daß sich alle Benutzer an dem System mit ihrem gewohnten Passwort anmelden können.

3. **require-pap**

Für die Authentifizierung soll das PAP Protokoll verwendet werden, da die meisten PDA's nur dieses unterstützen.

4. **plugin ...**

Das ist das `capi-plugin`, welches für die Kommunikation des pppd über die CAPI-Schnittstelle eingesetzt wird.

5. **inmsn**

Über den Parameter `inmsn` wird die lokale MSN-Nummer eingestellt, auf die der pppd reagieren soll. Wenn auf mehr als einer MSN eingehende Anrufe möglich sein sollen, müssen die verschiedenen Nummern durch Kommata getrennt angegeben werden.

6. **protocoll**

Hier muss das `v110async` Protokoll angegeben werden, damit die Einwahl mittel GSM bzw. HSCD möglich ist.

7. **localIP:remoteIP**

Einstellungen zur Vergabe der IP-Adressen, als `localIP` sollte eine Adresse des lokalen Servers gewählt werden. Die remote Adresse kann auch über die `pap-secrets` Datei vergeben werden.

8. **sonstiges**

Nun folgen noch einige Einstellungen zur Vergabe der DNS-Server Adresse und zu Komprimierung. Hier wird jegliche Komprimierung abgeschaltet. Ein Versuch mit verschiedenen Telefonen und Providern bzw. auch PDA's die Komprimierung zu aktivieren, kann zur Verbesserung der Performance der Verbindung führen, denn schließlich bietet die Verbindung standardmäßig nur 9,600 bps.



Anmerkung

Der Parameter *cli* kann dazu verwendet werden, die Liste der möglichen Anrufer zu beschränken. Der *pppd* reagiert dann nur auf die in der Liste des Parameter angegebenen Nummern. Allerdings muß dann auch sicher gestellt sein, daß die Anrufer die *Rufnummernübermittlung* eingeschaltet haben und auch alle Provider der Verbindung die Rufnummer übermitteln. Stellenweise kann dieses zu Problemen bei Anrufen aus dem Ausland führen.

4.3.2. Authentifizierung einrichten

Wie bereits beschrieben, soll die Authentifizierung über *PAP* erfolgen. Damit sich die User am System anmelden könne, müssen entsprechende Einträge in der Datei */etc/ppp/pap-secrets* vorgenommen werden. Hier ein Beispiel:

```
# file: /etc/ppp/pap-secrets
# client      server      secret      IP
gsmuser      isdnin     pass        192.168.1.22
*            isdnin     " "         *
```

In der ersten Zeile wird ein einzelner User definiert. Der Server wird diese Zeile benutzen, das das *Server* Feld den Einstellungen in *inc_v110* entspricht. Dem User wird die IP-Adresse 192.168.1.22 zugewiesen, wenn das *remoteIP* Feld in *inc_v10* nicht belegt wurde. Wenn der Parameter *login* gesetzt wurde, muss ebenfalls ein User *gsmuser* im System vorhanden sein.

In der zweiten Zeile wird jedem User der Login erlaubt, der einen gültigen Eintrag in der *passwd* Datei des Systems hat. Diese Einstellung erleichtert die Einrichtung des externen Zugriffs, birgt aber auch die Gefahr, daß unerwünschte User Zugriff von Außen auf das System erhalten.

4.3.3. Starten der GSM-Einwahl

Nun sollten alle Bedingungen erfüllt sein, um den *Einwahlserver* erfolgreich starten zu können. Also los gehts!

Der Einwahlserver soll über die *inittab* des Systems, wie ein normales Terminal gestartet werden. Dazu ist die Datei entsprechend anzupassen und anschließend die Aktualisierung dem System bekannt zu machen. Folgende Einstellung ist nötig:

```
I0:23:respawn:/usr/sbin/pppd nodetach call inc_v110
```

Nun muß die Änderung dem System als root mitgeteilt werden:

```
[host:/root/]$ init q
```

5. Clients einrichten

Bei der Einrichtung des Clients muß neben dem PDA (hier ein Visor als Beispiel) auch das Handy entsprechend konfiguriert werden. Es sei denn, man ist in der glücklichen Situation, einen PDA zu besitzen, der telefonieren kann.



Anmerkung

Die Einstellungen für die Einwahl sind je nach Handy unterschiedlich. Ich kann hier nur für die Mobiles Anleitungen geben, auf die ich Zugriff habe. Vor allem die Konfiguration der Kommunikationsparameter z.B. zum Einschalten des Übertragungsprotokolls variieren stark. Selbst bei den Geräten 7110 und 6310i von Nokia sind schon unterschiedliche Einstellungen nötig.

5.1. PDA einrichten

Die älteren Handspring Geräte verfügen standardmäßig noch nicht über die Fähigkeit, die IR-Schnittstelle für die Datenübertragung zum Mobile zu nutzen. Daher ist es nötig, die entsprechende Software zu installieren.

Für den Palm existiert ein **IR-enhancement Pack**, daß für den Visor verwendet werden kann.

5.1.1. Installation den IR-Enhancement Pack

Die benötigten Dateien für das Paket sind unter der Adresse <http://www.palmone.com/us/support/downloads/irenhance.html> [<http://www.palmone.com/us/support/downloads/irenhance.html>] bei Palmone zu finden.

Für die Installation kann den auf der Seite angegebenen Hinweisen gefolgt werden.



Warnung

Aber Achtung: Für den Visor dürfen nicht alle Dateien des Packs installiert werden. Die Dateien `AMX.prc` und `IrLib.rpc` dürfen auf keinen Fall installiert werden, sonst hilft nur noch ein komplettes Reset des Visors.

5.1.2. Modem einrichten

Wenn die Infrarot-Schnittstelle so weit vorbereitet ist, kann das Modem des Visors eingerichtet werden. Dazu ist zunächst unter **Einstellen-Serial/IR** die Datenverbindung auf **Infrared** eingestellt. Danach können die Einstellungen für das Modem vorgenommen werden. Diese sind abhängig von dem verwendeten Telefon und unten beschrieben.

Für alle Mobiles gilt aber:

```
Modem: Definiert
Geschwindigkeit: 14.400 bps
Lautstärke: aus
Flusskontrolle: aus
Land: andere
Folge: vom Handy abhängig
```

5.1.3. Netzwerkinstallation

Unter **Einstellungen-Netzwerk** werden die Zugangsdaten für die Netzwerkverbindung eingestellt. Dort kann mit dem Befehl **Dienst - neu** eine neue Verbindung erstellt werden. Als Verbindungstyp wird **ppp** gewählt sowie die Kontrollkästchen für die automatische Zuweisung von DNS und IP-Adresse aktiviert. Nach der Eingabe der Telefonnummer sowie des Passwortes kann die Verbindung dann genutzt werden. Ein Anwahlsript braucht nicht

definiert werden.

5.2. Hany einrichten

Hier nun die Einstellungen für den Anwahlsting, die vom verwendeten Mobile abhängig sind.

5.2.1. Nokia 6110i

Bei diesem Handy konnte eine Verbindung erfolgreich mit folgendem Anwahlsting hergestellt werden:

AT+cbst=75,0,1;+chsn=2,1,0,8

Beim Handy ist lediglich der Infrarot-Empfang einzuschalten und eine Sichtverbindung zwischen Handy und PDA sicherzustellen.

5.2.2. Nokia 7110

Beim Nokia 7110 funktioniert die Einwahl mit folgendem Init-String: **ATS35=0;+cbst=71,6,1**

5.3. Weitere Software

Für den Abruf von emails oder die Nutzung weiterer Programme, sind entsprechende Clients auf dem PDA zu installieren. So reicht der standardmäßig installierte email-Client für den Abruf von POP-Konten nicht aus. Man kann aber über die Verbindung auch den PDA mit dem Heimrechner synchronisieren, wenn das Konto für die anfallenden Gesprächsgebühren entsprechend gefüllt ist.