

ProNET V3.0

Benutzerdokumentation
23. August 1996

Michael Krause

Dieses Dokument wurde in Texinfo geschrieben und mittels PasT_EX und einem modifizertem `texinfo.tex` (für schönere Fonts) gesetzt.

1 Einleitung

1.1 Was ist *ProNET*?

ProNET ist ein einfaches Netzwerksystem. Die bedeutendste Anwendung ist die gemeinsame Nutzung von Geräten wie z.B. Festplatten und CD-ROMs in allen vernetzten Rechnern.

ProNET ist ein "einfaches" Netzwerksystem, weil es keinen Standard wie z.B. TCP/IP darstellt und höchstwahrscheinlich auch nie einer werden wird – es soll auch gar keine Konkurrenz zu ihnen sein. Dieses Programmpaket ist rein hobbymäßig entstanden und ist deswegen auch kein absolut professionelles Produkt.

Die Software basiert auf zweiseitigen Verbindungen, d.h. man kann immer nur zwei Computer auf einer Schnittstellenebene verbinden – jede Ebene hat genau zwei Enden. Dies entspricht dem sogenannten Client-Server-Modell:

```

/-----\           /-----\
|         | ---- ANFRAGE --->> |         |
| Client |           | Server |
|         | <<-- ANTWORT ----- |         |
\-----/           \-----/

```

Da *ProNET* jedoch ein modulares Treiberkonzept benutzt und keine festen Schnittstellen vorgibt, kann man ohne weiteres mehr als zwei Amigas vernetzen, indem man einfach eine zweite Verbindung öffnet.

Dieses Programm ist übrigens nur auf Amiga-Computern lauffähig. Es wird auch nie Versionen für andere Rechnerplattformen geben, weil das Dateisystem wesentlich darauf basiert, wie der Amiga generell mit Dateien umgeht.

Betrachtet man diese Tatsachen genau, ist *ProNET* genau das richtige für Sie, wenn Sie

1. nicht viel Geld ausgeben wollen
2. eine perfekte Integration in Ihre Arbeitsumgebung wünschen
3. hauptsächlich daran interessiert sind, zwei Amigas zu verbinden.

All dies mag Sie ein wenig an *ParNet* erinnern, und tatsächlich ist *ProNET* als Nachfolger zu dieser alten und in seinen Fähigkeiten begrenzten Software zu sehen. Es folgen ein paar Beispiele dessen, was gegenüber Matt Dillon's Software verbessert wurde:

- Modulares Treiberkonzept: parallele, serielle und MultiFaceCard3-Module sind mitgeliefert.
- Neues Netz-Gerät für jedes Zielgerät (kein NET: mehr)
- Erkennt deswegen auch Diskwechsel
- Unterstützt die meisten der V40-DosPackets
- 'cd's in Netzwerkverzeichnisse funktionieren!
- vollständig in Assembler geschrieben, deswegen kompakter und schneller

Bitte beachten Sie noch, daß es für dieses Paket kein Installationsskript gibt – Wenn Sie es benutzen wollen, sollten Sie sich ein wenig mit der Shell und mit Texteditoren auskennen.

1.2 Systemanforderungen

ProNET benötigt grundsätzlich Version 2 des AmigaOS und läuft auf jedem beliebigen Prozessor. Einige wenige Ausnahmen gibt es jedoch: '*pronet-server*', '*pronet.device*', '*pronet-page*', '*pronet-run*', '*pronet-talk*' und sämtliche Treibermodule laufen mit älteren Kickstart-Versionen. Sie wurden jedoch nur mit 1.3 getestet.

2 Installation

Bevor wir mit dem Kopieren beginnen, möchte ich Ihnen noch einmal erklären, was ‘*Devices*’ oder zu deutsch ‘*Geräte*’ sind. Wenn Sie finden, daß Sie hierüber genug wissen, können Sie diese Sektion gerne überspringen und gleich mit dem Installieren beginnen. Wenn Sie den Installationsvorgang nicht verstehen, kann Ihnen dies hier vielleicht weiterhelfen!

Weil AmigaDOS mehr als nur ein Speichermedium verwalten kann, bekommt jedes seinen eigenen Namen. Die prominentesten Beispiele wären da ‘*DF0:*’ für das eingebaute Diskettenlaufwerk und ‘*HD0:*’ für die erste Festplattenpartition. Immer, wenn Sie auf eine Datei auf so einem Medium zugreifen, geben Sie diesen Namen an, es sei denn natürlich, dieses Medium ist das Wurzelverzeichnis des aktuellen Verzeichnisses.

All diese Namen beschreiben einen Teil von AmigaDOS, den man *AmigaDOS-Device* oder *Gerät* nennt. ‘*CD0:*’, ‘*HD7:*’, ‘*PRT:*’ und ‘*RAM:*’ beschreiben alle verschiedene AmigaDOS-Geräte. Wie Sie vielleicht am Beispiel ‘*RAM:*’ bemerkt haben, ist das AmigaDOS-Device nur ein *Symbol* für ein Speichermedium, denn die Ram Disk ist schließlich kein reeller Teil der Hardware.

Dieses Device ist nicht mit den *Exec-Devices* zu verwechseln, obwohl man beide normalerweise nur *Device* nennt. Das erstere ist ein Symbol, das letztere ein Programm, das auf ‘*.device*’ endet und sich im ROM oder im ‘*DEVS:*’-Verzeichnis befindet. Exec-Devices sind Treiber für Hardware wie z.B. Festplatten- oder Diskettenlaufwerke. In diesem Fall müssen Sie nicht unbedingt wissen, wozu sie gut sind oder wie sie funktionieren.

2.1 Installieren des ‘*pronet.device*’

Das *pronet.device* ist der Kern des gesamten *ProNET*-Pakets. Es enthält alle Programmenteile, die den Datentransfer zwischen mehreren Amigas verwalten. Es enthält *keine* eigentlichen Übertragungsmodule, wie es der Fall in Version 1 war, weil es sich als sehr schwierig herausstellte, neue Übertragungsmodule für andere Schnittstellen zu schreiben.

2.1.1 Kopieren der Dateien

Dieses Device muß ins ‘*DEVS:*’-Verzeichnis aller am Netz beteiligten Maschinen kopiert werden.

Sobald Sie Ihr Keyfile empfangen, legen Sie es bitte als ‘*pronet.key*’ in das gleiche Verzeichnis.

Die *Übertragungsmodule* sind externe Programmmodule, die im 'DEVS:ProNET'-Verzeichnis gelagert werden. Bitte kopieren Sie alle mitgelieferten Module dorthin, natürlich können Sie später die nicht benutzen immer noch löschen. Wichtig: Nicht direkt nach 'DEVS:' kopieren, sondern in ein das eigene Verzeichnis!

Bis jetzt gibt es die Module 'internal-parallel' für den internen Parallelport (was übrigens die einzige Möglichkeit in *ProNET* Version 1 war), 'mfc-parallel' für den Parallelport einer möglicherweise vorhandenen MultiFaceCard3, und 'serial' für serielle Schnittstellen, die über *serial.device*-kompatible Treiber gesteuert werden. Das wären z.B. 'BaudBandit.device', 'duart.device' etc.

2.1.2 Installieren der Treiber

Bitte erstellen Sie nun die Datei 'DEVS:ProNET.config'. Diese wird Konfigurationsdaten für alle Schnittstellen beinhalten, die Sie für das Netzwerk benötigen.

Die Datei sollte wenigstens eine Zeile besitzen, die diesem Schema folgt:

```
pronet-device-[unitnummer]: [treibername] [treiberdaten]
```

Solch eine Zeile beschreibt eine sogenannte *Unit*, die einer durch einen *ProNET*-Treiber gesteuerten Schnittstelle entspricht. Je mehr Schnittstellen und Treiber Sie besitzen, desto mehr Verbindungen können Sie zu anderen Rechnern aufmachen. Sie werden die Möglichkeit haben, jede Unit zur Verbindung mit anderen Amigas zu benutzen. Wenn Sie z.B. eine Multiport-Karte mit 3 seriellen Ports haben, könnten Sie sich später mit 5 Amigas vernetzen, indem Sie die Multiport-Karte zusammen mit internem Parallel- und Seriellport benutzen.

Jede Unit wird durch eine beliebige positive Zahl gekennzeichnet (*unitnummer*). *treibername* ersetzen Sie bitte durch den Namen des externen Treibermoduls, *treiberdaten* sind Angaben, die der jeweilige Treiber zum korrekten Funktionieren benötigt. Diese können sich natürlich von Treiber zu Treiber unterscheiden.

2.1.2.1 'internal-parallel'

Als erstes werden wir diesen Treiber als Unit Nummer 0 einrichten. Natürlich können Sie auch eine beliebige andere Zahl benutzen, es ist Ihre Entscheidung! Sie müssen diesen Treiber nicht

unbedingt benutzen. Wie auch immer, ich gehe davon aus, daß sie es wollen, also erstellen wir folgende Zeile:

```
pronet-device-0: internal-parallel 0 5
```

Die Zahl 5 wird vom Treiber benötigt und gibt die Priorität des Transferprozesses an. Wenn Sie nicht wissen, wovon ich rede, lassen Sie diese Zahl einfach stehen! Kleinere Zahlen, wie zum Beispiel 0, bewirken, daß der Treiber nicht einfach die gesamte Rechenzeit für sich beansprucht, wenn Daten übertragen werden.

Die Zahl 0 ist die sogenannte *Rechnernummer* und ist von großer Bedeutung für das korrekte Funktionieren des Treibers. Zwei Rechner, die über den internal- oder mfc-parallel-Treiber vernetzt werden, müssen zwei verschiedene Nummern hier stehen haben: Eine Seite '0', die andere '1'. Wenn Sie beiden Seiten die gleiche Nummer geben, wird *ProNET* nicht funktionieren.

Dieser Treiber benötigt ein ParNet-kompatibles Kabel am eingebauten Parallelport.

2.1.2.2 'mfc-parallel'

Dieser Treiber wird für Sie nützlich sein, wenn Sie die MultiFaceCard3 von BSC besitzen. Hiermit können Sie den entsprechenden Parallelport benutzen, um Verbindungen zu anderen Rechnern aufzubauen. Dies funktioniert sowohl zu einem normalen internen Parallelport als auch zu einer zweiten MultiFaceCard3.

Bitte beachten Sie, daß Sie nur MultiFaceCards benutzen können, die mit einer 6821 PIA arbeiten – der Treiber wurde nur mit der MultiFaceCard3 getestet.

Also machen wir weiter mit Unit Nummer zwei:

```
pronet-device-2: mfc-parallel 0 1 5
```

0 ist die Nummer des MFC-Parallelports; falls Sie mehrere Karten dieses Typs besitzen, können Sie auch andere Nummern einsetzen.

Die Zahl 5 wird vom Treiber benötigt und gibt die Priorität des Transferprozesses an. Wenn Sie nicht wissen, wovon ich rede, lassen Sie diese Zahl einfach stehen! Kleinere Zahlen, wie zum Beispiel 0, bewirken, daß der Treiber nicht einfach die gesamte Rechenzeit für sich beansprucht, wenn Daten übertragen werden.

Die Zahl 0 ist die sogenannte *Rechnernummer* und ist von großer Bedeutung für das korrekte Funktionieren des Treibers. Zwei Rechner, die über den internal- oder mfc-parallel-Treiber vernetzt werden, müssen zwei verschiedene Nummern hier stehen haben: Eine Seite '0', die andere '1'. Wenn Sie beiden Seiten die gleiche Nummer geben, wird *ProNET* nicht funktionieren.

Dieser Treiber benötigt ein ParNet-kompatibles Kabel am entsprechenden Parallelport.

2.1.2.3 'serial'

Hier benutzen wir die Unit-Nummer eins:

```
pronet-device-1: serial BaudBandit.device 0 19200
```

`BaudBandit.device` ist der Name des `serial.device`, das Sie benutzen wollen, 0 ist die entsprechende Unit, die nur bei Multiport-Karten von Bedeutung ist. Bitte beachten Sie, daß das `8n1.device` auf einem A500 nicht mit diesem Treiber zusammenarbeitet. Wenn Sie ganz sichergehen wollen, benutzen Sie am besten das original `serial.device`

19200 ist die Baudrate, die auf beiden Seiten übereinstimmen muß. Probieren Sie, wie hoch Sie gehen können. 19200 ist eigentlich ein sehr sicherer Wert, der auch auf unbeschleunigten 500ern keine Fehler liefert. Typische höhere Werte wären z.B. 38400, 57600, 64000, 76800 oder 115200. Wenn Sie doch einen zu hohen Wert benutzen, kann es sein, daß das Netzwerk einfach aufhört zu arbeiten, besonders wenn der Prozessor sehr beansprucht wird. Es kann aber auch genauso gut sein, daß alles eine halbe Stunde in Ordnung ist und der Treiber dann "zusammenbricht".

Ohne weitere Argumente benötigt dieser Treiber ein Null-Modem-Kabel mit 7 Drähten, um korrekt zu arbeiten. Falls Sie in die Verlegenheit kommen sollten, nur ein 3-adriges zu haben, z.B. wenn Sie sich mit einem CD32 vernetzen, sollten Sie das Argument `3WIRE` ans Ende der Zeile setzen, natürlich auf beiden Seiten der Verbindung. 3-Wire-Verbindungen sind allerdings generell instabiler als 7-adrige, besonders wenn Ihre CPU stark ausgelastet ist. Hier hilft wieder nur ein Herumspielen mit der Baudrate.

2.2 Herstellen der Kabel

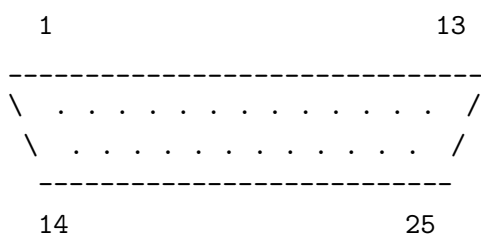
Nur damit eines klar ist: Was auch immer Sie sich hier zusammenlöten, ich übernehme keine Verantwortung für irgendwelche Schäden.

2.2.1 Das ParNet-Kabel

Sie löten jetzt ein Kabel, das die beiden Parallelports zweier Amigas verbindet. Verbinden Sie jeweils D7-D0, SEL, POUT und BUSY und schließen Sie ACK und SEL auf beiden Seiten kurz:

(2-9)	D7-D0	-----	D7-D0	
(12)	POUT	-----	POUT	
(11)	BUSY	-----	BUSY	PARALLEL PORT
(13)	SEL	--+-----+--	SEL	
(10)	ACK	-/ \-	ACK	
(18-22)	GND	-----	GND (18-22)	

Dies ist ein DB-25-Steckverbinder, wie man von außen draufschaut:



Achtung A1000-Benutzer: Der Parallelport ist ein Stecker, also brauchen Sie eine Buchse als Verbinder. Alle anderen benutzen einen Stecker. Schauen Sie sich den Port einfach mal an.

Das einfachste wäre es, ein fertiges 25er-Kabel zu kaufen, die ungültigen Verbindungen zu kappen und SEL und ACK auf beiden Seiten zu verbinden. **Verbinden Sie die Rechner niemals mit einem unmodifizierten Kabel! Prüfen Sie das Kabel lieber dreimal vorher, als sich hinterher über eine kaputte Kiste zu ärgern!**

2.2.1.1 Wechselwirkungen mit dem seriellen Port

Achtung: Die RI-(Ring Indicator-)Leitung am eingebauten *seriellen* Anschluß Ihres Amigas benutzt die SEL-Leitung, um einen Transistor zu speisen. Daraus ergeben sich aber leider einige Probleme, da diese Leitung am Netzwerk beteiligt ist.

Stellen Sie also sicher, daß entweder:

- kein seriellles Kabel angeschlossen ist, oder
- daß es die RI-Leitung nicht benutzt, oder
- Ihr Modem die RI-Leitung nicht benutzt, oder
- daß Sie nicht angerufen werden ;-)

Befolgen Sie diesen Ratschlag nicht, stürzt der ‘`internal-parallel`’-Treiber einfach ab, sobald ein Anruf reinkommt.

2.2.2 Das Null-Modem-Kabel

2.2.2.1 7-Draht-Schema

Zwei Schnittstellen müssen auf folgende Weise verbunden werden, damit Sie ein 7-adriges Null-Modem-Kabel erhalten:

```
Shield Earth -> Shield Earth
System Earth -> System Earth
TXD          ->      RXD
RXD          ->      TXD
RTS          ->      CTS
CTS          ->      RTS
DSR          ->      DTR
DTR          ->      DSR
```

Niemals Shield earth und System earth verbinden!

2.2.2.2 3-Draht-Schema

Um ein 3-Wire-Kabel zu erhalten, lassen Sie die DSR/DTR- und RTS/CTS-Verbindungen einfach weg, so daß sich folgender Aufbau ergibt:

```
Shield Earth -> Shield Earth
System Earth -> System Earth
TXD          ->      RXD
RXD          ->      TXD
```

Hier ebenfalls **niemals** Shield earth und System earth verbinden!

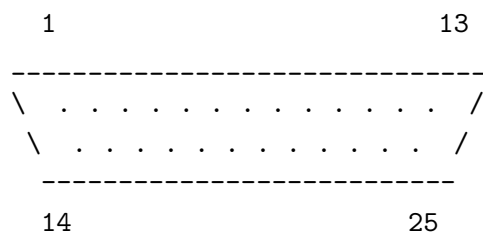
Wenn Sie Standard-Ports besitzen, sollten Sie lieber 7-Draht-Kabel verwenden, da sie einfach sicherer sind.

2.2.2.3 25-polige Steckverbinder

...sind die normalen seriellen Amiga-Ports.

Shield Earth (GND)	- Pin 1
System Earth (GND)	- Pin 7
TXD	- Pin 2
RXD	- Pin 3
RTS	- Pin 4
CTS	- Pin 5
DSR	- Pin 6
DTR	- Pin 20

Dies ist ein DB-25-Steckverbinder, wie man ihn von außen sieht:



2.2.2.4 9-polige Steckverbinder

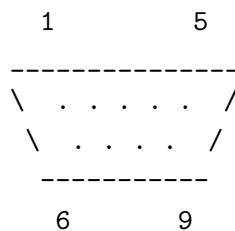
...werden häufig bei den restlichen Computersystemen benutzt und finden sich auch manchmal auf I/O-Extendern wie der *MultiFaceCard3* wieder.

```

Shield Earth (GND) - (mit dem Rahmen verbinden)
System Earth (GND) - Pin 5
    TXD             - Pin 3
    RXD             - Pin 2
    RTS             - Pin 7
    CTS             - Pin 8
    DSR             - Pin 6
    DTR             - Pin 4

```

Dies ist ein DB-9-Steckverbinder, wie man ihn von außen sieht:



2.3 Installieren des Dateisystems

Ganz einfach: Nehmen wir mal an, Ihr Arbeitsplatz ist ein 4000er. Sie wollen das CD-ROM-Laufwerk eines CDTV importieren, das unter dem Tisch steht, weil Sie keine Lust haben, Geld für ein extra SCSI-CD-ROM-Laufwerk auszugeben. Bei dieser Konfiguration ist es offensichtlich, daß das CDTV der *Server* sein wird, während der 4000er als *Client* fungiert.

Kopieren Sie einfach die Datei `'pronet-server'` in das `'C:'`-Verzeichnis oder an eine andere Stelle des Suchpfades auf dem Server, das ist der Rechner, von dem Sie Geräte importieren wollen. Danach kopieren Sie den `'pronet-handler'` in das `'L:'`-Verzeichnis des Hauptrechners, dem Client.

Das war's!

2.4 Installieren der Hilfsprogramme

Kopieren Sie `'pronet-run'`, `'pronet-page'`, `'pronet-start'`, `'pronet-stop'` und `'pronet-talk'` in die `'C:'`-Verzeichnisse Ihrer Amigas.

3 Anwendung

3.1 Starten des Servers ('pronet-server')

Auf jedem Amiga, den Sie als Server benutzen wollen, müssen Sie zuerst **'pronet-server'** starten. Als Vorgabe wartet es auf Client-Anfragen an der *ProNET*-Unit 0, aber Sie können die Unit ändern, indem Sie sie beim Start als Argument übergeben. Daraus resultiert, daß Sie natürlich mehrere Verbindungen von außen zu einem zentralen Server-Rechner herstellen können – dann müssen Sie lediglich den **'pronet-server'** für jede Unit einzeln starten.

Wenn Sie vergessen, den Server zu starten, werden die Client-Programme auf den anderen Rechnern einfach in eine Endlosschleife gehen – wenn also später mal nichts funktioniert, prüfen Sie als erstes, ob der Server überhaupt gestartet ist.

Sie können das Programm jederzeit durch ein **'CTRL-C'**-Signal unterbrechen, das empfiehlt sich besonders, bevor Sie den Serverrechner neustarten, weil er dann den angeschlossenen Clients Bescheid sagt. Ich empfehle Ihnen, den Server mittels `run <>nil:` zu starten, so daß er Ihre Shell nicht blockiert. Wenn Sie ihn abbrechen wollen, müssen Sie zu den Shell-Kommandos **'status'** und **'break'** greifen.

3.2 Importieren von Geräten ('pronet-start')

Das Netzwerk zu starten ist keine große Angelegenheit. Dazu dient der **'pronet-start'**-Befehl, den man folgendermaßen anwendet:

```
pronet-start LOCALNAME/A,REMOTENAME/A,UNIT/N,FLAGS/N,UNIQUE/S
```

LOCALNAME ist der Name des Gerätes, das Sie erzeugen wollen, während **REMOTENAME** den Namen des entsprechenden Geräts auf dem Server darstellt. Es ist wichtig, daß Sie hier nicht den Namen des Datenträgers (wie z.B. 'Work') angeben, sondern den Gerätenamen (z.B. 'hd0'). Bitte beachten Sie auch, daß Sie den Doppelpunkt weglassen müssen!

Für das genannte Beispiel könnten Sie also folgendes eingeben:

```

pronet-start cd1 cd0
      ^^^
      So heißt das Gerät auf dem A4000.
      ^^^
      So heißt es auf dem CDTV.

```

Die restlichen Argumente sind nicht zwingend erforderlich:

UNIT ist die *ProNET*-Unit, die Sie benutzen wollen (Vorgabe ist 0). **FLAGS** hat gegenwärtig keine Auswirkungen (und ist ebenfalls mit 0 vorgegeben). Der **UNIQUE**-Schalter hilft Ihnen bei folgender Situation:

Wenn Sie zwei Amigas vernetzen, die beide eine Festplattenpartition besitzen, die 'Workbench' heißt, werden Sie nach dem Netzwerkstart zwei Icons mit dem gleichen Namen 'Workbench' entdecken. Das ist übel und sollte vermieden werden, um einem seltsamen Verhalten von AmigaDOS vorzubeugen. Die Lösung ist eben dieser **UNIQUE**-Schalter. Er fügt an jeden importierten Datenträgernamen eine Zahl an, die der jeweiligen *ProNET*-Unit entspricht.

'**pronet-start**' zeigt eine Fehlermeldung an, falls etwas schiefgeht, z.B. wenn es das angegebene Gerät auf dem Server nicht gibt oder wenn die angegebene Unit nicht definiert ist. Treten keine Fehler auf, tätigt das Programm keine Ausgaben.

So, und jetzt wiederholen Sie diesen Schritt für jedes andere Gerät, das Sie noch importieren wollen. Viel Spaß!

3.3 'pronet-stop'

Wenn Sie meinen, daß Sie ein bestimmtes Netzwerkgerät nicht mehr benötigen, können Sie es entfernen. Rufen Sie einfach '**pronet-server**' mit dem entsprechenden Gerätenamen auf, und schon wird die entsprechende Verbindung geschlossen.

Warum sollten Sie das tun? Nun, falls Sie den Server ausschalten oder ihn zurücksetzen, ohne den Client zu stoppen, können ganz seltsame Dinge auf dem Client passieren: Es könnte sein, daß er komplett hängt, oder daß Sie auf keine Geräte mehr zugreifen können – das hängt damit zusammen, daß der Client denkt, daß der Server immer noch läuft und deswegen fleißig Anfragen an ihn schickt.

Klar: Wenn Sie sowieso beide Rechner abschalten wollen, können Sie sich diese Prozedur sparen.

3.4 ‘pronet-flush’

Sie können dieses Programm (ohne weitere Argumente) benutzen, um das ‘`pronet.device`’ aus dem Speicher zu entfernen. Dadurch werden alle Ressourcen freigegeben, die von den *ProNET*-Treibern belegt wurden. ‘`pronet-server`’ und ‘`pronet-stop`’ haben diese Funktion bereits eingebaut. Somit werden Sie kaum eine Gelegenheit finden, bei der Sie dieses Programm nutzbringend einsetzen können, aber ich habe es der Vollständigkeit halber trotzdem geschrieben.

3.5 ‘pronet-page’

Mit Hilfe dieses Programms können Sie Nachrichten an die Benutzer anderer Netzwerkrechner schicken. Rufen Sie es mit der *ProNET*-Unit und der entsprechenden Nachricht auf, z.B.:

```
pronet-page 0 Guten Morgen!
```

Die Nachricht wird in einer Warnmeldung am oberen Bildschirmrand angezeigt.

3.6 ‘pronet-run’

Mit diesem Programm können Sie Programme auf anderen Netzwerkrechnern starten, ohne deren Maus oder Tastatur zu benutzen. Rufen Sie es mit der *ProNET*-Unit und dem auszuführenden Kommando auf. Der Befehl *muß* dabei in Anführungszeichen stehen, z.B.:

```
pronet-run 0 "dir hd0: all"
```

3.7 ‘pronet-talk’

Mit diesem Programm können sich die Benutzer zweier Netzwerkrechner miteinander unterhalten. Rufen Sie das Programm auf *beiden* Rechnern mit der richtigen *ProNET*-Unit auf. Als Vorgabe benutzt es die Unit 0.

4 Weitere Anwendungen

In diesem Kapitel möchte ich Ihnen ein paar Tips geben, wie Sie das letzte aus ProNET herausholen. Es gibt da ein paar Dinge, die nicht ganz offensichtlich sind, aber trotzdem nützlich sein könnten.

4.1 Wiederverbindung

Schauen wir uns noch einmal das Beispiel von vorhin an. Manchmal werden Sie in die Verlegenheit kommen, den A4000 neuzustarten, weil irgendein Programm mal wieder abgestürzt ist (...nein, nicht wirklich, weil.. wir benutzen ja einen Amiga ;-)). Wenn Sie das Netzwerk nun wieder hochfahren wollen, müssen Sie das CDTV nicht ebenfalls neu starten – Es reicht, wenn Sie die Geräte auf dem A4000 einfach wieder neu starten, mit Hilfe von `'pronet-start'`, das ist alles.

4.2 Verbindungen ueber Kreuz

Die meisten Konfigurationen werden so ähnlich wie meine Beispielkonfiguration aussehen, d.h. ein Hauptrechner (der Arbeitsplatz) und ein Server, an dem nicht gearbeitet wird, der aber seine Geräte zur Verfügung stellt. Wie auch immer, wenn Sie sich mit einem Freund vernetzen, und beide arbeiten an den Rechnern, können Sie das Netzwerk auch über Kreuz starten, d.h. beide Rechner sind Client und Server zugleich. Sie können beide die Geräte des jeweils anderen Systems ansprechen.

4.3 Netzwerk ueber die Telefonleitung

Stellen Sie sich vor, Sie wollen einem Freund, der weit weg wohnt, die Installation einer bestimmten Software erklären. Sie würden ihn anrufen und ihm genau sagen, was er zu tun hat – aber das wird schiefgehen, denn er wird Sie nicht verstehen (Diese Annahme ist von Murphys Gesetz abgeleitet und hat sich in vielen Situationen als wahr erwiesen ;-)).

Sie könnten auch eine Modemverbindung herstellen, ein Netzwerk zwischen Ihren Amigas einrichten, für ihn ein paar Shellscrips einrichten etc. Klingt das interessant? Dann lesen Sie weiter!

4.3.1 Vorbereitung

Beide Seiten müssen eine *ProNET*-Unit definieren, die den ‘**serial**’-Treiber verwendet. Die entsprechende Schnittstelle wird jedoch kein Nullmodem-Kabel verwenden, sondern ein richtiges Modem! Wie gewohnt, setzen Sie die Geschwindigkeit des seriellen Ports viermal so hoch wie die Leitungsgeschwindigkeit des Modems. Diese Geschwindigkeit muß nicht unbedingt auf beiden Seiten übereinstimmen.

Beide Modems müssen so konfiguriert sein, daß sie das DTR-Signal ignorieren, was meistens durch den Befehl **AT&D0** geschieht.

4.3.2 Verbindung

Laden Sie jetzt Ihr Terminalprogramm. Entscheiden Sie sich, welcher von Ihnen der Anrufer ist, dieser ruft jetzt den anderen mittels **ATD** an. Sobald der Angerufene mit **ATA** geantwortet hat und Sie beide die **CONNECT**-Nachricht erhalten haben, verlassen Sie das Terminalprogramm.

Jetzt fahren Sie einfach so fort, als wären Sie durch ein Nullmodem-Kabel verbunden: Starten Sie den Server und richten Sie die Geräte ein, natürlich alles mit der vorhin definierten Unit, die das Modem anspricht. Das ist alles – nicht gerade sehr schnell, aber es funktioniert.

5 Interna

5.1 Erstellen eigener *ProNET*-Treiber

Ich habe diesen Abschnitt nicht übersetzt, weil ich zu faul bin. Außerdem sollte man als Programmierer keine Probleme mit dem Englischen haben ;-)

A *ProNET* driver is an ordinary executable residing in the 'DEVS:ProNET/' directory. This executable will be loaded for every *ProNET* Unit requiring it; it doesn't have to be re-entrant. The code is entered at the very first position, which must contain an **initialization routine**:

d0 contains the bytes "RST!" for identification, so that this driver can not be executed by accident.

d1.1 is a Flags register. It is **not** mapped to the Flags argument of the OpenDevice() call!! For Version 3, this contains a 1; if it contains another value, your driver should return an error.

a1 contains a pointer to the configuration string following the driver ID. Example:

```
pronet-device-7: yourdriver bla this is data for your driver
```

In this case, **a1** would contain a pointer to the null-terminated string 'bla this is data for your driver'.

a0 finally points to an array which must be filled out by your routine:

```
UBYTE ReadSignalBit
UBYTE pad0
APTR ReadQuery
APTR Read
APTR Write
APTR      Exit
```

ReadSignalBit must contain a valid signal bit for the 'pronet.device' task. The task will include this bit into its Wait() loop and call the function pointed to by **ReadQuery**, if this bit becomes set.

Your **ReadQuery** routine must analyze the incoming packet and return the following values:

```
d0.w length of the packet's main data
d1.w ProNET destination port
d2.w ProNET source port
```

If your routine comes to the conclusion that it was called by mistake and no incoming packet is pending, it must return NULL in d0! If this routine returns something useful, it's guaranteed that the routine pointed to by **Read** will be called after that.

The **Read** routine will be called with a pointer to the place to put the packet data into in a0. It must copy the incoming packet there and return. You don't have to return an error code, because this routine must **always** work!

Let's come to the **Write** routine now. It is called when the '**pronet.device**' wants your driver to transmit data to the other machine. The routine will be called with following arguments:

```
a0 *data1
d0.l length of data1 (guaranteed to be even)
a1 *data2
d1.l length of data2 (guaranteed to be even)
d2 destination port
d3 source port
```

Both chunks of data must be appended so that the other side thinks you've transmitted only one big chunk of data! The expression d0+d1 is always guaranteed to be lower than or equal to \$4000. If d1 contains zero, only one chunk of data is transmitted. If your routine succeeds in transmitting the data, you must return NULL in d0; if timeout occurred or the line was busy or anything else which prevented you from transmitting, you must return a different value.

The **Exit** routine is new for ProNET V3, since this version offers network shutdown. Do here what is necessary, in your opinion ;-)

The initialization routine must return an error code in d0. If NULL is returned, the initialization succeeded, other values indicate an error.

All routines must preserve d2-d7/a2-a6.

5.2 Unterstützte DosPacket-Typen

ProNET unterstützt die meisten neuen DosPackets, die in V36 und V39 eingeführt wurden: Vielleicht wurden Sie schon von *ParNet* durch so einen '2.0 Pkt ACT_PARENT_FH'-Requester genervt – Sie werden ihn nie wiedersehen! Hier ist die Liste der von *ProNET* unterstützten Dos-Packets:

ACTION_FINDINPUT	
ACTION_FINDUPDATE	V33
ACTION_FINDOUTPUT	
ACTION_END	
ACTION_READ	
ACTION_WRITE	
ACTION_SEEK	
ACTION_CURRENT_VOLUME	
ACTION_SET_FILE_SIZE	V36
ACTION_LOCK_RECORD	V36
ACTION_FREE_RECORD	V36
ACTION_LOCATE_OBJECT	
ACTION_FREE_LOCK	
ACTION_COPY_DIR	
ACTION_PARENT	
ACTION_SAME_LOCK	V36
ACTION_CREATE_DIR	
ACTION_CHANGE_MODE	V36
ACTION_FH_FROM_LOCK	V36
ACTION_COPY_DIR_FH	V36
ACTION_PARENT_FH	V36
ACTION_EXAMINE_OBJECT	
ACTION_EXAMINE_NEXT	
ACTION_EXAMINE_FH	V36
ACTION_DELETE_OBJECT	
ACTION_RENAME_OBJECT	
ACTION_MAKE_LINK	V36
ACTION_READ_LINK	V36
ACTION_SET_COMMENT	
ACTION_SET_DATE	
ACTION_SET_PROTECT	
ACTION_INFO	
ACTION_RENAME_DISK	
ACTION_INHIBIT	
ACTION_FORMAT	V36
ACTION_SERIALIZE_DISK	V39
ACTION_MORE_CACHE	
ACTION_WRITE_PROTECT	
ACTION_IS_FILESYSTEM	V36

```
ACTION_NIL  
ACTION_FLUSH  
ACTION_DISK_INFO
```

Diese Typen werden noch nicht unterstützt und werden vielleicht später implementiert:

```
ACTION_EXAMINE_ALL      V36  
ACTION_EXAMINE_ALL_END  V39  
ACTION_SET_OWNER        V39  
ACTION_ADD_NOTIFY       V36  
ACTION_REMOVE_NOTIFY    V36
```

6 Verschiedenes

6.1 Registration

ProNET ist Shareware. Ohne ein Keyfile wird Ihnen aufgefallen sein, daß das Netzwerk nach 5 Minuten aufhört zu arbeiten. Um ein eigenes Keyfile zu erhalten, schicken Sie bitte 20DM an meine Adresse, die Sie im Support-Abschnitt finden. Benutzer, die sich bereits für Version 2 registriert haben, erhalten ihr Keyfile für 10DM.

Bitte schreiben Sie Ihre komplette Anschrift **lesbar** auf. Wenn Sie eine e-mail-Adresse besitzen, kann ich Ihnen das Keyfile auch uucodiert zuschicken. Bitte geben Sie auch deutlich an, welche Version von *ProNET* Sie verwenden.

Sobald ich eine Registration erhalte, werde ich sie normalerweise innerhalb von zwei oder drei Tagen beantworten – falls Sie drei Wochen warten müssen, können Sie sicher sein, daß ich im Urlaub war ;-) Zusammen mit dem Keyfile erhalten Sie auch die neueste Version von *ProNET*.

6.2 Support

Wegen der niedrigen Registrationsgebühr und den Kosten herkömmlicher Post, ist es mir nicht möglich, jede Frage auf diesem Wege zu beantworten. Besitzen Sie eine e-mail-Adresse, steigen die Chancen auf eine Antwort ins Unermessliche ;-) Bitte versuchen Sie auch zunächst, eine Antwort in dieser Anleitung zu finden, besonders im FAQ-Kapitel.

Neue Versionen von *ProNET* werden ins Aminet und auf verschiedene lokale BBS upgeloadet. Sie können auch einen Blick auf die WWW Support Site werfen:

`http://www.szczecin.pl/~rawstyle/pronet.html`

Wenn Sie einen Bug gefunden haben, erreichen Sie mich unter:

Michael Krause
Mannesallee 24
D-21107 Hamburg
Germany

rawstyle@blackbox.dame.de

Bug-Reports sollten so ausführlich wie möglich sein, besonders was die Konfiguration angeht. Mich interessiert es dabei übrigens nicht, ob Sie eine 2-Tasten- oder eine 3-Tasten-Maus benutzen ;-)

6.3 Credits & Danksagungen

Gesamte *ProNET*-Software von

Michael Krause

ParNet (Inspiration) und Entwurf des Parallelkabels von

Matt Dillon

Beta-Testing (Vielen Dank!)

Thomas Schwarz, Wolfgang Gutberlet, Anti-"Es-funktioniert-nicht"-brain

Mein besonderer Dank gilt Nik Soggia <nsoggia@falcon.telnetwork.it>, der eine Lösung für das ReqTools-Problem gefunden hat.

Schließlich möchte ich noch meinen Dank allen denen aussprechen, die mir Mails geschrieben haben, die mehr als nur 'ProNET funktioniert nicht' enthielten und mich so ermutigten, an diesem Paket weiterzuarbeiten. Diese Version wäre ohne Eure Hilfe niemals entstanden!

6.4 Geschichte

6.4.1 Version 3

Freigegeben am 23-Aug-96, mit `pronet.device` **36.6**, `pronet-handler` **36.4**, `pronet-server` **36.4**, `pronet-start` **36.3**, `pronet-stop` **36.1**, `pronet-flush` **36.0**, `pronet-talk` **36.0**, `pronet-run` **36.1**, `pronet-page` **36.0**.

- Kein Herumfummeln mehr mit der MountList und der ProNET.config, `'pronet-start'` erledigt alles!
- MultiFaceCard3-Treiber geschrieben.
- Serial-Treiber funktioniert jetzt richtig.
- Netzwerk kann wieder heruntergefahren werden, ohne den Rechner neuzustarten.
- Komplett neue Dokumentation.
- Serial-Treiber arbeitet auf Wunsch auch mit 3-Draht-Kabeln.
- Kommandozeilenprüfung in `'pronet-page'` and `'pronet-run'`.
- `'pronet-talk'` benutzt jetzt immer Port 1994.
- Neuer `'pronet.device'` Fehlercode `'PNDERR_UNIT_NOT_DEFINED'`, Fehler in der Device-Open-Routine behoben.

6.4.2 Version 2.1

(Bugfix) Freigegeben am 08-Jul-95, mit `pronet-talk` **34.2**, `pronet-server` **35.0**.

- `pronet-talk` UND `pronet-server` akzeptierten das UNIT-Argument nicht...

6.4.3 Version 2

Freigegeben am 24-Jun-95, mit `pronet.device` **35.4**, `pronet-handler` **34.2**, `pronet-server` **34.9**, `pronet-talk` **34.1**, `pronet-run` **34.1**, `pronet-page` **34.0**.

- Wiederverbindung.
- Device friert den Rechner nicht mehr ein.
- Priorität des Transferprozesses kann eingestellt werden.
- Diskwechsel-Erkennung verbessert.
- Komplett neue Dokumentation

- READ-Actions werden gesplittet übertragen und verbrauchen nicht mehr so viel Speicher.
- EXAMINE-Packets bewirkten Speicherverlust auf dem Server.
- Komplett neues pronet.device, mit modularem Treiberkonzept.
- "+"-Erweiterung >> Unitnumber
- C include file
- Alles, was ich sonst noch vergessen habe.
- Viele Bugfixes, ProNET läuft stabiler.

6.4.4 Version 1

Freigegeben am 01-Nov-94 (Erste Release), mit pronet.device **33.9**, pronet-handler **33.5**, pronet-server **33.5**, pronet-talk **33.0**, pronet-run **33.0**, pronet-page **33.0**.

6.4.5 Version 0

Mir kam die Idee für *ProNET* im Dezember 1993, nachdem ich von ParNet so enttäuscht war. Mit dem pronet.device fing ich im April 1994 an.

6.5 Zukunft

- Server rebootbar machen.
- MUI-GUI fertigstellen.
- Volumenamen-Erweiterung frei konfigurierbar machen.
- Lokales Caching von :.backdrop und :Disk.info
- Netz-Keyboard, Netz-Maus
- XPK crunched packet transfer (a la v42)
- Sicherheit: Geräte vorm Netz sperren.
- Notification DOS-Packets.
- DiskSalv-Problem fixen.
- Net-CON:
- Net-Serial-Device
- ProNET-Driver für SANA-II-Devices.

- ProNET-Driver für parnet.device
- ProNET-Driver für AmigaDOS-Devices (PAR: etc.)
- SANA-II-Device für pronet.device

7 FAQ - Frequently Asked Questions

- Ist es möglich, das Netzwerk zu unterbrechen, um kurz etwas zu drucken und dann über eine Switchbox wieder aufs Netz zurückzuschalten?

Nein, das ist nicht möglich, und das wird es auch vermutlich nie sein, weil der Parallel-Treiber davon ausgeht, daß die Hardware-Register für den Parallelport nicht verändert werden.

- Gibt es eine Möglichkeit, ein Programm auf einem Rechner zu starten und seine Ausgaben in einem CON:-Window auf einem anderen zu sehen?

Nein, aber das möchte ich in der nächsten Version einbauen.

- Ist es möglich, den seriellen Port eines anderen Netzrechners vom Hauptrechner aus mitzubenutzen, um z.B. ein zweites Modem oder ein MIDI-Interface anzuschließen?

Nein, auch das wird es erst in einer späteren Version geben. Danke an Giovanni Gigante für den Vorschlag!

- Wird der 'pronet-handler' jemals auf Kickstart 1.3 laufen?

Nein.

- Würde eine spezielle 020/030/040-Version des 'pronet.device' die Geschwindigkeit erhöhen?

Nein.

- Ich habe einen Bug entdeckt: Wenn ich den Hauptcomputer resette und alle Devices neu mounte, kann ich ohne Probleme weiterarbeiten. Wenn ich allerdings den Server resette und den 'pronet-server' dann neu starte, läuft nichts mehr.

Das ist kein Bug, weil der 'pronet-server' einige wichtige Datenstrukturen für den Hauptrechner erzeugt, und diese werden gelöscht, sobald Sie den Server resettet. Eine Folge davon ist auch, daß in 'Über-Kreuz- Netzwerken' kein Rechner einen Reset machen darf. Das ist ein essentieller Unterschied zwischen *ProNET* und professionellen Netzwerksystemen.

- Kann man das pronet.device als Treiber für SANA-II Netzwerksoftware benutzen?

Nein, weil mein Device nicht nach den SANA-II-Regeln gebaut ist. Allerdings möchte ich eine Art Aufsatz für das pronet.device programmieren, so daß man alle ProNET-Treiber dann z.B. mit AmiTCP benutzen kann.

- Warum habe ich das Device nicht gleich SANA-II-kompatibel gemacht?

Weil ich noch nicht von diesem Standard gehört habe, als ich mit *ProNET* angefangen habe.

- Muß ich 'pronet-start' für jedes Gerät einzeln starten oder kann ich irgendwie die Aufrufe für mehrere Geräte zusammenfassen?

Nein. Erstellen Sie sich ein Shell-Script, wenn Sie zu faul sind ;-)

- Ich benutze ein CDTV bzw. A500+A570 als Server. Das CD-ROM-Laufwerk scheint Diskwechsel nicht zu erkennen. Ist das ein Bug in *ProNET*?

Nein, es handelt sich hierbei um einen Bug im CDTV-Dateisystem. Verwenden Sie einfach den Shell-Befehl `diskchange cd0:` auf dem Client oder auf dem Server.

- Ich benutze die Workbench, um große Files auf den Server zu kopieren. Manchmal beschwert sich mein Rechner über zu wenig Speicher.

Versuchen Sie, zum Kopieren die Shell zu benutzen oder ein DOPus-ähnliches Tool. Es handelt sich dabei um ein Problem, welches bei großen ACTION_WRITEs auftritt, weil ein entsprechender Speicherblock auf dem Server alloziert werden muß.

- Ich benutze DiskSalv, um eine Disk über das Netzwerk auf einen anderen Rechner zu retten. Das geschieht allerdings ungewohnt langsam.

Zu dem Problem habe ich bis jetzt noch keine Lösung gefunden, aber ich arbeite dran.

- Wenn ich eine neue Schublade auf einem Netzwerkrechner einrichte, geschieht das ebenfalls sehr langsam.

Hierbei handelt es sich anscheinend um das gleiche Problem wie das zuletzt beschriebene.

Hmm... bitte entschuldigen Sie, daß nicht allzuviele Fragen dabei waren, die ich mit einem einfachen 'Ja' beantworten konnte :) Obwohl, Moment.. hier ist doch noch eine:

- Ist *ProNET* besser als ParNet?

<Was wohl?> :-)

8 Rechtliches

ProNET Version 3 is Copyright © 1994-1996 by Michael Krause.

8.1 Disclaimer

"Ich werde das nicht schon wieder lesen..."

THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDER AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

8.2 License

- This license applies to the product called *ProNET*, a collection of programs for the Amiga computer, published by Michael Krause under the concepts of shareware, and the accompanying documentation. The terms "Program" and "Package" below, refer to this product. The licensee is addressed as "you".
- You may copy and distribute verbatim copies of the package as you receive it, in any medium, provided that you conspicuously and appropriately publish only the original, unmodified package, with all copyright notices and disclaimers of warranty intact and including all the accompanying documentation, example files and anything else that came with the original.

- You may not copy your personal key file to other persons nor make it publically available. You may not use other key files than your very own with your *ProNET* copy. You may only use this key file on your very own computers.
- Except when otherwise stated in this documentation, you may not copy and/or distribute this program without the accompanying documentation and other additional files that came with the original. You may not copy and/or distribute modified versions of this program.
- You may not copy, modify, sublicense, distribute or transfer the program except as expressly provided under this license. Any attempt otherwise to copy, modify, sublicense, distribute or transfer the program is void, and will automatically terminate your rights to use the program under this license. However, parties who have received copies, or rights to use copies, from you under this license will not have their licenses terminated so long as such parties remain in full compliance.
- By copying, distributing and/or using the program you indicate your acceptance of this license to do so, and all its terms and conditions.
- Each time you redistribute the program, the recipient automatically receives a license from the original licensor to copy, distribute and/or use the program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein.
- You agree to cease distributing the program and data involved if requested to do so by the author.

Table of Contents

1	Einleitung	1
1.1	Was ist <i>ProNET</i> ?	1
1.2	Systemanforderungen	2
2	Installation	3
2.1	Installieren des 'pronet.device'	3
2.1.1	Kopieren der Dateien	3
2.1.2	Installieren der Treiber	4
2.1.2.1	'internal-parallel'	4
2.1.2.2	'mfc-parallel'	5
2.1.2.3	'serial'	6
2.2	Herstellen der Kabel	6
2.2.1	Das <i>ParNet</i> -Kabel	7
2.2.1.1	Wechselwirkungen mit dem seriellen Port	7
2.2.2	Das Null-Modem-Kabel	8
2.2.2.1	7-Draht-Schema	8
2.2.2.2	3-Draht-Schema	8
2.2.2.3	25-polige Steckverbinder	9
2.2.2.4	9-polige Steckverbinder	9
2.3	Installieren des Dateisystems	10
2.4	Installieren der Hilfsprogramme	10
3	Anwendung	11
3.1	Starten des Servers ('pronet-server')	11
3.2	Importieren von Geraeten ('pronet-start')	11
3.3	'pronet-stop'	12
3.4	'pronet-flush'	13
3.5	'pronet-page'	13
3.6	'pronet-run'	13
3.7	'pronet-talk'	13
4	Weitere Anwendungen	15
4.1	Wiederverbindung	15
4.2	Verbindungen ueber Kreuz	15
4.3	Netzwerk ueber die Telefonleitung	15
4.3.1	Vorbereitung	16
4.3.2	Verbindung	16

5	Internas	17
5.1	Erstellen eigener <i>ProNET</i> -Treiber	17
5.2	Unterstuetzte DosPacket-Typen	19
6	Verschiedenes	21
6.1	Registration	21
6.2	Support	21
6.3	Credits & Danksagungen	22
6.4	Geschichte	22
6.4.1	Version 3	23
6.4.2	Version 2.1	23
6.4.3	Version 2	23
6.4.4	Version 1	24
6.4.5	Version 0	24
6.5	Zukunft	24
7	FAQ - Frequently Asked Questions	27
8	Rechtliches	29
8.1	Disclaimer	29
8.2	License	29