SERIELL mit PIP

Hallo Clubmitglieder !

Vor einiger Zeit traf sich der Club auf der Jahreshauptversammlung , zumindest war eine anberaumt worden.
Viele von Euch kennen den Ablauf eines Clubtreffens und auch diesmal verlief das Treffen wie gewohnt. Andreas hatte seinen Genie 3s inclusive Lötkolben und Oskar dabei , diverse REH-280 Rechner im Siemens-Gehäuse waren gekommen , Hartmuds M4P und Christopher mit seinem TANDY Modell II inclusive drei 8"-Drives waren dabei und viel Zeit zum fachsimpeln.
Meine Probleme fingen mit dem TANDY Modell II an, denn Christopher wollte ein Terminalprogramm auf dem TANDY haben um Dateien vom Modell 4P auf das Modell II zu übertragen.
Nun , der Genie 3s von Andreas wollte partut nicht mit den 8"-Drives umgehen, und ansonsten war auf dem TANDY Modell II nichts DFUe-mäßiges vorhanden.

Außer ... PIP.

Tja , mit PIP, da ging doch auch was mit der seriellen Verstöpselung. Wenn man genau weiß wies geht ist es einfach , aber ich möchte hier auch erzählen wie man es nicht machen sollte.
Man nehme nicht auf der einen Seite einen KAYPRO II mit TERM und auf der anderen Seite einen TANDY MII und PIP. Beide verstehen sich (gleiche Baudrate vorausgesetz) prächtig, aber wenn der TANDY speichert sendet das Terminalprogramm auf dem Kaypro fleißig weiter, wodurch natürlich einige BITs fehlen und das übertragene Programm kaum laufen dürfte.
Mir ging es so und eigendlich hätte ich dies wissen müßen, aber da ich den KAYPRO auf dem Clubtreffen erst bekam war ich natürlich ganz neugierig auf diesen Rechner und beachtete so eine wichtige Voraussetzung für die serielle Übertragung binärer Daten nicht, welche im normalen Terminalmodus selten gebraucht wird.

Es muß ein Handshake her. Als ich mir dies alles wieder klar gemacht hatte war schon einiges an kohlensäurehaltigem Gebräu die Kehle hinunter, alkohohlfrei versteht sich und ein Vormittag hiermit verbracht.
PIP gehört zur Standardsoftware eines CP/M - Rechners und kann auch die seriellen Schnittstellen bedienen, vorausgesetzt PIP (für CP/M2.2) ist hierfür gepatcht. PIP für CP/M 3.x macht dies ohne Änderungen, da die Devicetreiber AUXIN und AUXOUT die serielle Schnittstelle bedienen und mit dem Programm DEVICE die Parameter der Schnittstellen eingestellt werden können.
Leider ist bei der seriellen Datenübertragung das Controll-Z die Dateiendmarke und in einer binären Datei ist oft ein Controll-Z als Bildschirmsteuerzeichen oder sonstiges mitten im Programmcode enthalten.
Also geht es nicht so einfach mit der Übertragung einer binären Datei, welche schon vorzeitig durch Controll-Z beendet wird, aber es gibt da Abhilfe.

Die binäre Datei wird in eine HEX-Datei umgewandelt.

Mit dem Programm UNLOAD wird aus einer binären Datei eine Filename.HEX Datei erstellt.

Die HEX-Datei besteht nur aus hexadezimalen Zahlen und enthält so keine bei der serielle Übertragung störenden Zeichen.
Nach der korrekten Übertragung auf den Zielrechner wird die HEX-Datei mit LOAD, HEXCOM, DDT(Z) oder dergleichen in eine COM-Datei umgewandelt. LOAD und DDT wird dem CP/M2.x mitgeliefert und HEXCOM ist dem CP/M3.x beigefügt.

Die Syntax dieser Programme ist einfach:

Befehl	Eingabe	Ergebniss
UNLOAD	Filename.COM ----->	Filename.HEX
LOAD	Filename.HEX ----->	Filename.COM
HEXCOM	Filename.HEX ----->	Filename.COM

DDT(Z) und SID(Z) laden auch HEX-Dateien und speichern diese Dateien dann als COM-Datei ab. Bei DDT muß mit SAVE die Datei abgespeichert werden, während SID den Befehl W(rite) mitbringt.
Ihr seht es gibt einige Möglichkeiten eine HEX/COM - Datei zu bearbeiten.
Diese HEX-Datei ist also Voraussetzung für unsere problemlose serielle Datenübertragung. Nachdem dies nun bekannt ist und auf beiden Rechnern PIP zur Verfügung steht sucht man sich ein NULL-MODEM Kabel und meistens klappts. Falls beide Rechner die gleiche Belegung der RS232 Schnittstelle haben, sollte eine Probeübertragung klappen, ansonsten muß man an einem der Stecker wahrscheinlich PIN 2(TXD) und PIN 3(RXD) tauschen.
Hier noch mal zur Erinnnerung - zwei Rechner sind vorhanden, gleiche Baudzahl eingestellt, Kabel ist richtig konfiguriert und nun sollte ein ASCII-Text zum Schnittstellentest übertragen werden. Glücklicherweise war das PIP des TANDY MII schon entsprechend an die serielle Schnittstelle angepaßt und mein Rechner ist ein CP/M3 Gerät, und dessen PIP kann die serielle Schnittstelle von Haus aus bedienen.
Die Änderung an PIP unter CP/M 2.x beschreibe ich am Ende es Artikels.

Testen der seriellen Verbindung:

```
SENDER:      PIP PUN:=b:Filename[EB]
EMPFÄNGER:   PIP b:Filename:=RDR:[EB]
```

Bedeutung der Optionen E und B:
PIP schickt mit der OPTION "E" die Datei auch auf das angeschlossene Display, was aber nur bei Textdateien einen Sinn ergibt und hier nur zum Testen der ASCII-Übertragung gebraucht wird.
Mit der Option "B" arbeitet PIP im BLOCK-Modus und die übertragene Datei wird in Blöcken zu je 256 Bytes gesendet. Die Daten werden blockweise in den Puffer gelesen und zum Zielgerät übertragen, bevor ein neuer Block gelesen werden kann. Das Blockende-Zeichen ist der ASCII-Code 19, d.h. CTRL-S oder auch X-OFF.

Der Block-Modus ist also ein X-ON/X-OFF Protokoll und verhindert, daß beim Speichern der Daten welche verloren gehen, bzw. nicht empfangen werden.
Also, bei Datenübertragung mit PIP über die RS232 Schnittstelle nur im BLOCK-Modus arbeiten.
Falls der oben gezeigt "Test" der Verbindung fehlerfrei funktioniert hat, kann jetzt die eigendliche Datei im HEX-Format übertragen , und die HEX-Datei anschließend auf dem Zielrechner mit HEXCOM oder LOAD in eine COM-Datei umgewandelt werden.
Ein kleines Problem haben wir aber nur kurz angesprochen und wollen das hier nachholen.
PIP, unter CP/M3, ist für AUXIN/AUXOUT-Device generiert worden. Jeder CP/M 3.x (CP/M+) Rechner kann mit PIP für CP/M 3.0 die serielle Schnittstelle bedienen.
Unter CP/M 2.x haben wir leider nicht diese Standardisierung und PIP muß von der Lage der seriellen Schnittstelle informiert werden.
Im allgemeinen wird der serielle Baustein über IN/OUT-PORTS angesprochen und diese sind hoffentlich in den Handbüchern zum Rechner beschrieben.
PIP muß also wissen an welcher Portadresse der serielle Baustein angesprochen wird und wann der Baustein dazu in der Lage ist. Hierfür ist in PIP im Bereich von 10AH bis 1FFH Platz zur Verfügung gestellt worden.
Es wurden zwei Labels vorgesehen, und zwar INP:(INPUT) an der Stelle 103H. PIP springt, um Daten vom zu übernehmen, die Adresse 103H per CALL-Befehl an. Dann wird in 110H eine kleine Routine durchgeführt, welche das eingetroffene Zeichen an der Stelle 109H ablegt und mit RET nach PIP zurückspringt. Das Zerro-Bit (BIT 8) muß auf Null gesetzt sein.
Ähnliches geschieht auch beim Senden der Daten. Hier wird das Label OUT:(OUTPUT) an der Adresse 106H angesprungen , eine kleine Routine auf 120H angesprungen , welche das zu sendende Zeichen aus Register C liest und an die serielle Schnittstelle weitergibt.
Natürlich müssen die Baudraten der gekoppelten Rechner stimmen, aber da es die verschiedensten seriellen Bausteine gibt kann hier nur empfohlen werden, über ein SETUP Programm oder dergleichen die Baudrate vorher einzustellen. Meist sind Programme vorhanden, mit denen der Druckeranschluß, die Laufwerke und die serielle Schnittstelle konfiguriert werden können. Falls nichts dergleichen machbar ist muß man einen Fachmann fragen und ist in diesem Artikel nicht weiter ausgeführt.

Hier noch eine kleine Routine mit welcher PIP gepatcht werden kann. Damit es für alle nützlich ist auch in 8080 Code:
Die hier genutzten Portadressen sind für den TATUNG TPC2000 gültig.

Prozessor 8080:

```
IN:
    103H      JMP 110H  ; Sprung zur Eingabeunterroutine
    106H      JMP 120H  ; Sprung zur Ausgabeunterroutine
    110H      IN  1DH   ; Port 1DH liefert das Statuswort
    112H      ANI 2     ; Testen ob Zeichen eingetroffen
    114H      JZ  110H  ; Schleife bis Zeichen da ist
    117H      IN  1CH   ; Port 1CH liefert das Zeichen von
                        ; der seriellen Schnittstelle
```

```
     119H       ANI 7FH    ; höchstes Bit ausblenden
     11BH       STA 109H   ; Empfangenes Zeichen in 109H ablegen
     11EH       RET        ; und nach PIP zurück

OUT:
     120H       IN  1DH    ; liefert Statuswort
     122H       ANI 1      ; testet ob serieller Baustein bereit
                           ; ist zum senden
     124H       JZ  120H   ; Schleife bis senden möglich
     127H       MOV A,C    ; Auszugebendes Zeichen aus C in A
     128H       OUT 1DH    ; Zeichen über Port  1DH ausgeben
     12AH       RET        ; Sprung aus Sendeunterroutine  zu PIP
```

Prozessor Z80:

```
IN:
     103H       JR  110H       ; Sprung zur Eingabeunterroutine
     106H       JR  120H       ; Sprung zur Ausgabeunterroutine
     110H       IN  A,(1DH)    ; Port 1DH liefert das Statuswort
     112H       AND 2          ; Testen ob Zeichen eingetroffen
     114H       JR  Z,110H     ; Schleife bis Zeichen da ist
     117H       IN  A,(1CH)    ; Port 1CH  liefert das Zeichen von
                               ; der seriellen Schnittstelle
     119H       AND 7FH        ; höchstes Bit ausblenden
     11BH       LD  (109H),A   ; Empfangenes Zeichen in 109H ablegen
     11EH       RET            ; und nach PIP zurück

OUT:
     120H       IN  A,(1DH)    ; liefert Statuswort
     122H       AND 1          ; testet ob serieller Baustein bereit
                               ; ist zum senden
     124H       JR  Z,120H     ; Schleife bis senden möglich
     127H       LD  A,C        ; Auszugebendes Zeichen aus C in A
     128H       OUT (1DH),A    ; Zeichen über Port 1DH ausgeben
     12AH       RET            ; Sprung aus Sendeunterroutine wieder
                               ; zu PIP
```

Mit dieser kleinen Anleitung müßte jeder sein PIP für CP/M2.2 patchen können um die seriellen Schnittstellen anzusprechen. Etwas Geduld braucht man hierzu, vor allem da die Konfiguration der seriellen Schnittstelle, bzw. der Stecker doch unterschiedlich sein können.
Unterschieden wird hier zwischen DTE (Data Terminal Equipment) oder deutsch DEE (Daten-Endeinrichtung/en) also Rechner; und DCE (Data Communication Equipment) beziehungsweise DÜE (Daten-Übertragungseinrichtung/en) also Modem.

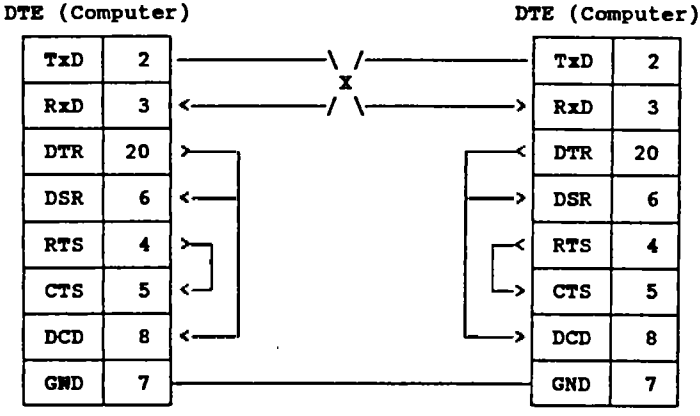
Modem und Computer passen direkt zusammen, da hier Sende- auf Empfangsleitung geschaltet ist. Manchmal ist aber auch der Computer (am Tatung kann dies gejumpert werden) als DCE konfiguriert.
Bei zwei Rechnern, welche als DTE beschaltet sind sollte das nachfolgende Nullmodem für eine funktionierende Verbindung der beiden Rechner sorgen.

In der Regel sind als RS232 Stecker 25-polige D-SUB Typen eingebaut, oder oft bei IBM und kompatiblen platzsparende 9polige D-SUB .
Da ein XON/XOFF - Protokoll genutzt wird reicht ein NULL-Modemkabel aus und braucht nur aus drei Adern zu bestehen, eventuell als vierte Ader eine Abschirmung. Die PINS 4,5,6,8,20 werden am Stecker miteinander verbunden.
Falls jemand Probleme mit der seriellen Übertragung bei seinem Rechner hat stehe ich gerne mit RAT & TAT beiseite. Für folgende Rechner sind bei mir funktionfähige Terminalprogramme zu erhalten: TATUNG TPC2000 - EPSON PX-8 - GENIE 3s - Morrow MD3 - KAYPRO II und Generic CP/M+ Versionen der gängigen Terminalprogramme.

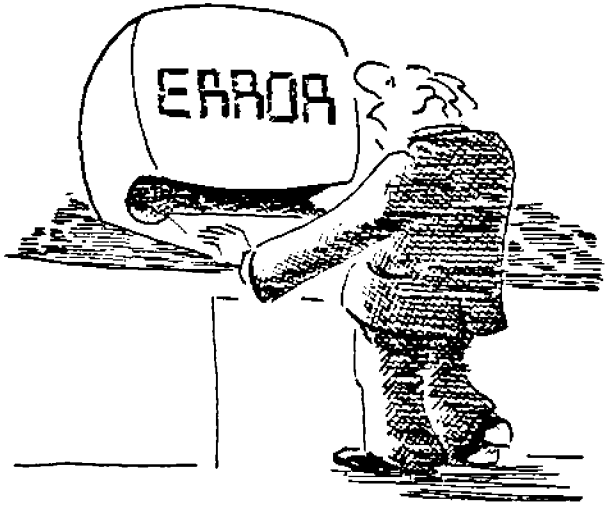
/ Meine E-Mail Adresse: fritz chwolka@solaris.hamm.sub.org /

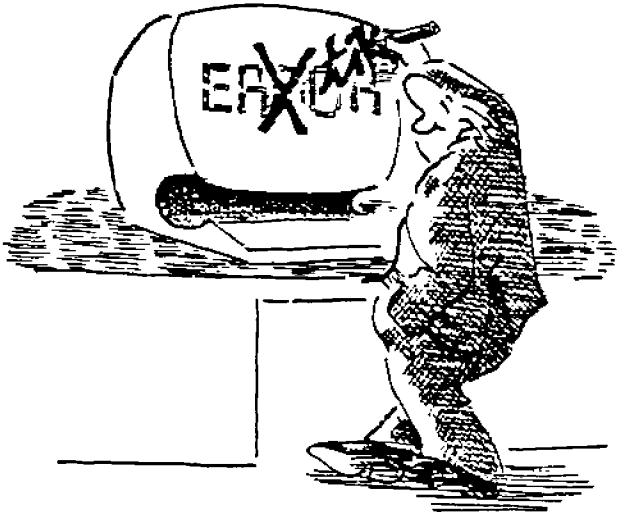
oder Postadresse laut INFO.

Belegung des Nullmodemkabels bei 25 D-Sub.



Anmerkung: Dieser Artikel wurde mit Wordstar 4 geschrieben und auf einem Panasonic KX-P1124 mit Einzelblatteinzug der Firma Orbitronic ausgedruckt.





© Dietmar Große

Noch serieller mit PIP

Im Info Nr. 35 hat "uns Fritz" geschrieben, wie man im Notfall auch ohne kompatible Laufwerke, bzw. wenn keine Informationen über das Diskettenformat vorliegen, Dateien mit Hilfe von PIP von einem Rechner zum anderen übertragen kann. Dazu noch zwei kleine Anmerkungen/Ergänzungen:

Wenn man PIP als Parameter für die Übertragung zusätzlich noch ein "H" (Hex) mitgibt, schaltet es in den Hex-Modus und überprüft, ob mit der übertragenen Datei (in der Z80-Welt üblicherweise im sog. Intel-Hex-Format) alles stimmt. Diesen Service hat man nur mit "B" (Block) alleine nicht und kann verfälschte Zeichen höchstens daran erkennen, daß der HEX-COM-Wandler (HEXCOM, LOAD o.ä.) hinterher meckert, oder das Programm nicht läuft. Im Hex-Modus stellt PIP anhand der Prüfsumme in jeder Zeile fest, ob möglicherweise ein Fehler aufgetreten ist und meldet dies dann sofort. Wenn man nur beim sendenden Rechner ein E (Echo) als Parameter angibt, wird dieser weiter gebremst, was dem empfangenden Rechner etwas mehr Zeit gibt, die ankommenden Zeichen zu verarbeiten. Normalerweise dürfte das nur bei einem relativ langsamen Rechner auf der einen und einem schnelleren Rechner auf der anderen Seite notwendig sein, aber aus Erfahrung würde ich sagen, daß man das immer so machen sollte. Auch ist es nicht unbedingt gesagt, daß es mit einer kleineren Baudrate immer besser geht, ich habe es schon erlebt, daß es mit 1200 Baud einwandfrei ging, mit 300 Baud aber nur Schrott angekommen ist. Warum ? Keine Ahnung, aber wenn alle Geheimnisse gelöst wären, wäre es ja langweilig. Außerdem sollte man beim sendenden Rechner die Schnittstelle tunlichst auf das XON/XOFF-Protokoll einstellen, sonst funktioniert der Blockmodus nicht. PIP sendet zwar XOFF- und XON-Zeichen, aber die Übertragung wirklich anhalten muß der Schnittstellentreiber bzw. das BIOS. Wenn sich der empfangende Rechner nach der Übertragung nicht mehr zurückmeldet, kann man ihm mit "PIP PUN:=EOF:" bzw. "PIP OUT:=EOF:" noch ein explizites Dateiende zukommen lassen.

Was in Fritz' Artikel vielleicht auch nicht ganz klar rausgekommen ist und viel Zeit und Nerven kosten kann: Der Test mit

PIP PUN:=filename[EB] und PIP filename=RDR:[EB]

(hinter dem Filenamen darf, entgegen der Angabe von Fritz, übrigens KEIN Doppelpunkt kommen, sonst gibt's eine Fehlermeldung !!)

funktioniert nur, wenn das CP/M das IO-Byte unterstützt ! Es gibt Implementationen, die das Byte einfach nicht benutzen und dann wundert man sich, daß nichts geht. Ausprobieren kann man das mit "PIP LST:=CON:", mit dem alle Eingaben von der Tastatur auf den Drucker umgelenkt werden sollten. Da LST, genauso wie EOF weiter oben, ein logisches Gerät ist, muß hier zur Unterscheidung von einem Filenamen nun tatsächlich ein Doppelpunkt dahinter stehen. Wenn man aber ein komfortabes BIOS hat, das u.a. bei einem leeren Empfangspuffer brav selber auf ein neues Zeichen wartet und der Testtext so fehlerfrei angekommen ist, ist man schon fertig und braucht die ganze Patcherei garnicht. Dann kann man mit PUN: bzw. RDR: arbeiten und im Prinzip sofort loslegen und beim CP/M Plus, wo man in beiden Fällen nur über AUX: geht, muß man sowieso nicht fummeln und kann auch gleich anfangen.

Wenn es aber doch sein muß, sei es weil das mit dem IO-Byte nicht funktioniert, oder dauernd Zeichen ankommen, ohne daß der andere Rechner

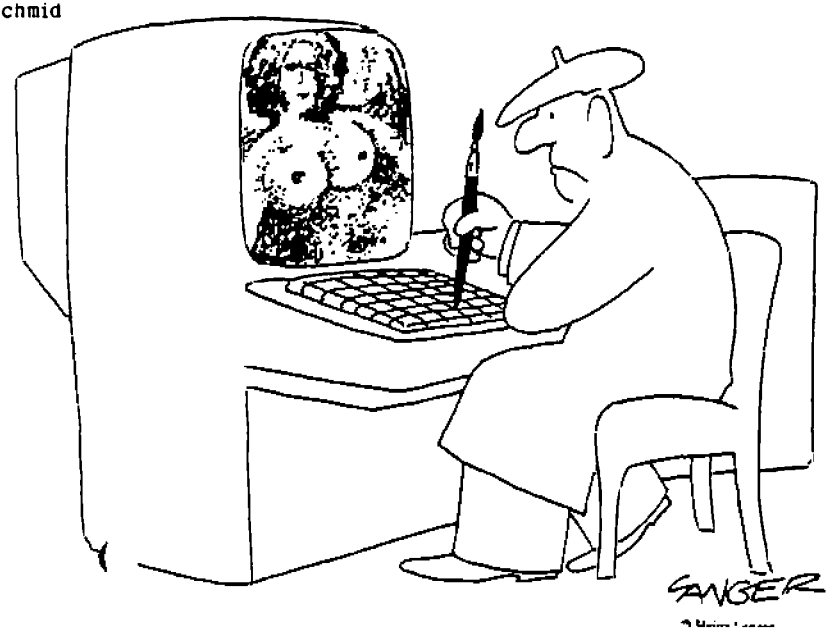
sendet, und man PIP schließlich so gepatcht hat, daß es die Ports direkt anspricht, muß die Ein- und Ausgabe folgendermaßen aussehen: (PUN: und RDR: gehen nach wie vor über das Betriebssystem, da helfen alle Patches nichts !)

PIP OUT:=filename[EBH] und PIP filename=(NP:[BH]

Wenn diese Hürde dann schließlich genommen ist und die Übertragung klappt, sollte man so schnell wie möglich LYNC rüberziehen. Das Programm besteht aus dem eigentlichen COM-File (LYNC.COM), einem Installer (LYNC-INST.COM) und einer Treiber-Bibliothek (IO.DRV). Am besten konfiguriert man das Programm schon vorher, dann braucht man nur ein File zu übertragen. Das File ist auch als Hex-File so klein, daß es zusammen mit PIP in den Arbeitsspeicher paßt, sodaß im Block-Modus erst nach Ende der Übertragung auf die (langsame) Floppy zugegriffen werden muß und es so keine Probleme mit dem Protokoll geben kann.

Warum aber nun ausgerechnet LYNC ? Ganz einfach, es ist verhältnismäßig klein und so ziemlich das einfachste und sicherste Programm für diesen Job, bei dem man eigentlich nichts falsch machen kann. Der Installer arbeitet menügeführt und man muß nur die Ports der seriellen Schnittstelle, die man ja eventuell schon von den Patches in PIP kennt, per Hand eingeben; für viele Rechner (u.a. TRS-80 Model I und IV) gibt es einen fertigen Menüpunkt, den man nur anzuwählen braucht. Danach kann man von einem der beiden Rechnern beliebig Files senden und empfangen, ohne daß am anderen Rechner jemand sitzen muß. Man kann sich z.B. sogar das Directory des anderen Rechners ferngesteuert ansehen und dann mit SEND und FETCH loslegen. Das ist viel bequemer als mit Kermit oder sonstigen Terminalprogrammen, wo man immer noch einen Helfer braucht. Die Übertragung wird durch ein LYNC-eigenes Protokoll gesichert und nichtmal das mutwillige Unterbrechen der Leitung bringt es aus dem Tritt. Nachdem die Verbindung wieder steht, geht es sofort weiter. Außerdem überprüft sich LYNC beim Starten selber, sodaß man dadurch nochmal eine Kontrolle hat, ob die Übertragung des Hex-Files fehlerfrei war.

Alexander Schmid



.Hardware

Das Club 80 Netzwerk

oder: Poor Man's LAN

von: Rainer Schmitz, Göppingen
Stand: 11. August 1991

Warum?

Jeder kennt wohl die Situation, wenn man auf einen CP/M-Rechner trifft, dessen Diskettenformat man (beziehungsweise die eigene Kiste!) nicht verarbeiten kann und es steht ein Datentransfer an. Sofern beide Rechner über eine serielle Schnittstelle und passende Modemprogramme (LYNC, ZMP, QTERM, MEX, KERMIT oder ähnliche) verfügen, kann man den Datentransfer zwar langsam, aber sicher über ein Nullmodem-Kabel durchführen. Schwierig wird die Sache aber, wenn mehrere Rechner miteinander in Verbindung treten sollen, wie es bei umfangreichen "Gebietsrechenzentren" à la Fritz Chwolka oder Helmut Jungkunz oder bei Clubtreffen (auch ausgefallenen...) vorkommen kann. Man steht dann vor der Alternative, entweder bei den Hobbykollegen mit dem Formatmanager (hallo Z280-Besitzer!) Schlange zu stehen, um die Datenpumpe anlaufen zu lassen oder mit einem möglichst langen Nullmodemkabel und einigen Adaptern und "Geschlechtswechslern" zu versuchen, die RS-232-"Norm" zu umgehen und die Kommunikation zu beginnen. Dabei stört mich, daß man den "Stolperdraht" regelmäßig umstecken muß, um mit einem anderen Comp zu kommunizieren. Kurz, beide Lösungen können nicht als optimal betrachtet werden.

Die Super-Luxus-Lösung des Problems wäre natürlich ein LAN (Local Area Network) in PC-Manier, dem steht allerdings entgegen, daß kaum jemand über ein paar überzählige Ethernet-Karten und ein passendes CP/M-NetBIOS verfügt, außerdem ist die zum Betrieb nötige Kabelage nebst Steckverbindern nicht gerade billig. Zusätzlich stellt sich die Frage, ob ein LAN nicht unnötiger "Overkill" für die oben angesprochene Aufgabenstellung ist.
Der Wunschzettel, den ich daraufhin im Geiste aufstellte, sah folgendermaßen aus:

- kein Umbau am Rechner selbst erforderlich
- keine besondere Hard- und Software nötig, speziell kein "Rumfummeln" am Betriebssystem, möglichst Verwendung vorhandener Modemprogramme
- niedriger Materialaufwand, damit niedriger Geldaufwand
- einfach nachzubauen, auch für Nichtelektroniker
- störungssicher und robust

Wie?

Einen brauchbaren Lösungsansatz fand ich im Leitartikel von Herwig Feichtinger in der mc Heft 7/1986 : dort beschrieb er ein einfaches Netzwerk, das in der mc-Redaktion eingesetzt wurde, um Texte zwischen den Redakteuren auszutauschen, da diese es leid waren, dauernd Disketten hin und her zu tragen. Es basierte darauf, daß die seriellen Schnittstellen in einer Art "verdrahtetem Oder" miteinander verbunden waren. Der Materialaufwand war extrem gering (drei Widerstände, eine Diode), als Verbindungskabel war Koaxkabel vorgeschlagen. Ich habe diese Schaltung aufgebaut und ausprobiert und dabei festgestellt, daß die "handelsüblichen" Modemprogramme damit bei Datentransfer nicht liefen: die Ursache war schnell gefunden. Wenn man sich die Schaltung (Bild 1) anschaut, wird man feststellen, daß der Sender sein eigenes Echo mithört! Das Modemprogramm erwartet aber bei einem Datentransfer nicht irgendetwas, sondern ein entsprechendes Quittungszeichen, mit dem das empfangende Programm den erfolgreichen Empfang eines Datenblocks kundtut (oder auch nicht). Beim Terminalbetrieb stört das Echo nicht weiter, da man dann gleich am Schirm sieht, was man eingetippt hat, sofern man nicht im Terminalmodus "Local Echo" eingestellt hat, denn dann hat man den "Jägermeister-Effekt": man sieht alles doppelt!

Die Schaltung funktioniert so: der Widerstand R3 in der Masseleitung soll ganz einfach Ausgleichsströme zwischen den Rechnern (unterschiedliche Massepotentiale) auf ein ungefährliches Maß begrenzen. Durch den Widerstand R2 in der RXD-Leitung wird der Empfängereingang hochohmiger gemacht, so daß mehr Eingänge parallelgeschaltet werden können. Der Senderausgang TXD zieht bei Ruhepegel den Anschluß Netzwerk+ über den Widerstand R1 unter die Empfängerschaltschwelle, bei aktivem Pegel wird R1 durch D1 überbrückt. Die Abhilfe des Problems beim Datentransfer erforderte kurzes Nachdenken und noch ein paar Bauteile der billigen Natur. In Bild 2 ist zu erkennen, wie's gemacht wird: der Transistor T1 schließt ganz einfach den Empfängereingang kurz, wenn der zugehörige Senderausgang aktiv wird. Angesteuert wird er mit R4 von TXD aus. D2 verhindert, daß der Transistor durch die negative Ausgangsspannung von TXD bei Ruhepegel zerstört wird.

Bei geschicktem Aufbau ist es möglich, die gesamte Schaltung in eine Steckerhaube für einen Submin-D-Stecker (so die offizielle Bezeichnung) einzubauen. Damit der Kontakt zum restlichen Netzwerk hergestellt werden kann, sollte jeder Anschluß mit zwei parallelgeschalteten Cinch-Buchsen (auch als RCA-Buchsen bekannt) versehen werden. Zum Anschluß gehört natürlich auch das passende Verbindungs[illegible].l: ich empfehle aus eigener Erfahrung ganz normales NF-

Kabel, und zwar die einadrige abgeschirmte Sorte. Bei den niedrigen Kabelpreisen halte ich eine Lange von drei Metern pro Station für angemessen, das reicht in den meisten Fällen (sprich: Clubtreffen). Der Innenleiter (Seele) kommt an den Netzwerk + Anschluß, die Abschirmung an den Netzwerk - Anschluß, das entspricht jeweils Stift und Massehülse beim Cinch-Stecher. Den Stecker am freien Kabelende steckt man dann in die freie Netzwerk-Buchse des nächsten Rechners.

Versuche ergaben noch bei ca. 30 m Kabellänge einwandfreie Übertragungen bei 9600 Bd, was ja wohl ausreicht. Ich sehe aber keinen Grund, warum es bei den höheren Baudraten nicht auch gehen soll! Das Oszilloskop zeigte jedenfalls bei 19200 Bd und ca. 250 m Kabel (!) am freien Ende noch schöne saubere Flanken. Ach ja, noch was: wenn der Rechner auf Hardware-Handshake Wert legt, sollte man ihm den Gefallen tun und RTS mit CTS verbinden (Pins 4 und 5), dasselbe gilt für die Pins 6,8 und 20 (das Dreieck rund um Masse, Pin 7). Diese Pinbelegung gilt aber nur für 25-polige Verbinder nach Norm! Bei meiner Kiste (Joyce) kann ich aber wählen, ob ich Handshake haben will oder nicht (via SETSIO). Besitzer anderer Rechner sollten lieber ins Handbuch schauen (so vorhanden!). Gerade bei Normbelegungen sind die Hersteller ja besonders einfallsreich...

Eine weitere Frage wäre noch in einem Feldversuch zu klären: nämlich die, wieviele Stationen das Netz verträgt. Nach meinen Berechnungen sollten bis zu sechs Stationen möglich sein, das hängt aber von verschiedenen Randbedingungen ab. Der in den meisten Schnittstellenschaltungen verwendete 1488 (75488) kann durchschnittlich 10 mA Ausgangsstrom liefern. Da jeder Eingang eines Empfängers 1489 ca. 1.2 mA Eingangsstrom oberhalb der Schaltschwelle benötigt, kann man sich schnell ausrechnen, wann Schluß ist. In der Realität gibt es aber Streuungen der Bauteilewerte, die diese Grenze nach oben oder unten verschieben können, und das probiert man wohl am besten aus. Bei der Originalschaltung aus der mc war bei drei Stationen am Netz Schluß, wie man durch Berechnung der Ströme im Netzwerk leicht nachweisen kann! Daher habe ich auch einige Werte gegenüber der Ursprungsschaltung verändert, ohne daß bei den Versuchen Probleme auftraten.

Womit?

Wie weiter oben schon angedeutet wurde, war angestrebt, das Netzwerk mit Standardsoftware zu betreiben, um den bei Software-Neuentwicklungen üblichen Zyklus Editieren - Assemblieren - Systemabsturz zu umgehen (ich gebe zu, mir fehlte dazu die Motivation und Erfahrung). Bei den ersten Versuchen setzte ich (oh Schande!!) mangels eines weiteren CP/M-Rechners einen kompandebilen Computer (mit MeSsy-DOS) mit Procomm ein, dazu meinen "Taschenknecht"

Portfolio. Alle diese Rechner verstanden sich unter XMODEM prächtig! Auf CP/M-Seite setzte ich ZMP ein, das ich inzwischen auch auf meinen neuen Rechner, einen Wavemate Super Bullet, angepaßt habe (falls jemand den Quelltext für den Overlay braucht, bei mir melden!).

Einen ersten Härtetest bestand das Netzwerk bein Treffen des Club 80 am 9. bis 12. Mai in Leipheim (nicht Leibheim, Gerald!). Wie bereits im letzten Clubinfo vermerkt, funktionierte das Ganze mit ca. 20 m Kabel recht gut. Es wurde hauptsächlich ZMP und LYNC zur Kommunikation eingesetzt, wobei man mit LYNC sogar den anderen Rechner "fernsteuern" kann, also Laufwerke wechseln, Directories anzeigen und dergleichen mehr (nur Disketten wechseln muß man von Hand!). Auf diese Weise gelangten einige Programmpakete aus der PD von einem zum anderen Rechner, und das über nur zwei Drähte! Mit 9600 Bd geht der Filetransfer sogar recht flott, so daß eigentlich bei einem der nächsten Treffen ein Rechner mit einer mit dem Vereinsarchiv gefüllten Platte und LYNC in der Ecke stehen und als "Fileserver" dienen könnte. Denkbar wäre auch ein Mailboxprogramm, mit dem man sich unterhalten könnte. Dabei muß man sich aber bewußt sein, daß dann nur Kommunikation mit dem Fileserver oder der Mailbox möglich ist, nicht aber mit anderen Rechnern. Das Netzwerk funktioniert ähnlich wie CB-Funk, wenn einer spricht, ist halt der Kanal belegt und die anderen müssen warten. Wenn zwei gleichzeitig sprechen, kommt Müll raus, aber das ist der Preis der Einfachheit. Besonders lustige Effekte kann man beobachten, wenn man mehr als zwei Stationen unter LYNC betreibt: da das Programm versucht, bei jeder Aktion eine Verbindung herzustellen, und jedesmal zwei antworten, sind bald alle drei damit beschäftigt, sich gegenseitig "die Hand zu schütteln". Die Konsequenz der Eigenheiten dieses Netzwerks ist daher: Disziplin der Benutzer! Bevor man die Leitung belegt, sollte man im Terminal-Modus nachschauen, ob gerade Ruhe herrscht, und erst dann loslegen. Wenn dann doch zwei gleichzeitig senden, passiert außer einer versauten Übertragung nichts weiter, das heißt, ihr müßt in diesem Fall nicht mit kleinen Atompilzen, die eurem Rechner entsteigen, rechnen! (Das gilt aber nur, wenn ihr bei der Hardware keinen Mist gebaut habt, auch 220 V sind von der Netzwerk-Teilnahme ausgeschlossen!)

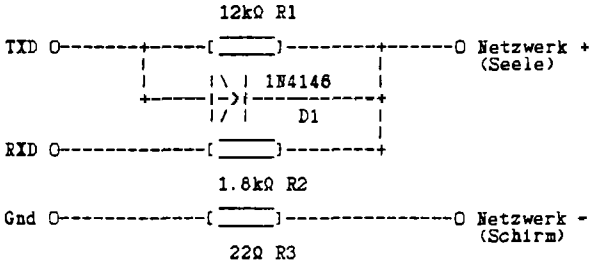
Zusammenfassung (für Diagonalleser)

In diesem Artikel habe ich ein Netzwerk für den Filetransfer "nach Holzfällerart" beschrieben, das für unsere Clubtreffen und den heimischen Gebrauch gedacht war. Seine Einfachheit wird durch ein paar Funktionseinschränkungen erkauft, diese stellten sich aber in der Praxis als tolerabel heraus, besonders bei Berücksichtigung des Aufwands. Hinzu kommt, daß das Netzwerk unabhängig vom Betriebssystem funktioniert, also auch für PC's, Atari's, Amiga's und andere geeignet ist, sofern diese über eine serielle Schnittstelle nach RS 232 (V-24) mit TXD und RXD verfügen. Ob es zu einer Kommunikation kommt, hängt dann nur noch von der (Modem-)Software ab.
Soweit mein Bauvorschlag, seht zu, was ihr draus macht, beim nächsten Clubtreffen will ich jede Menge CP/M-Rechner mit Netzwerkanschluß sehen!

Ausblick (auch das noch!)

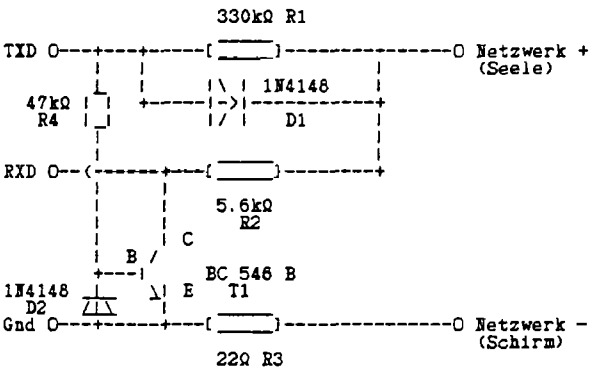
Seit ich mit diesem Artikel angefangen habe, erhielt ich einige faszinierende Informationen aus der Welt des Amateurfunks: dort ist im UKW-Band ein drahtloses Netzwerk namens "Packet Radio" entstanden. Es handelt sich dabei um ein modifiziertes X.25-Protokoll à la ISBN. Dadurch ist es möglich, auf einem Funkkanal mehrere Verbindungen gleichzeitig zwischen verschiedenen Stationen aufzubauen. Speziell die Hardware dürfte für uns interessant sein: der typische TNC (das ist das Modem zum Anschluß eines Computers an ein Funkgerät) ist mit einer Z80 CPU und SIO (wegen dem synchronen X.25-Protokoll) aufgebaut. Ein Bekannter besorgt mir Schaltpläne und ein Eprom mit der Betriebssoftware, so daß man sich die Sache mal anschauen kann. ZNP enthalt ja ein Packet Radio Overlay namens ZMYAPP, das erleichtert die Sache gewaltig. Soweit also die neuesten Infos zum Thema Ausbau des Netzwerks.

Bild 1



Netzwerkankopplung nach mc 7/1986

Bild 2



Erweiterte Netzwerkankopplung