

nicht vergessen !!!  
**Hobbytronik Stuttgart**  
**Sonntagstreff**  
**27.11.1994**

9.00Uhr oder 12.00Uhr  
nach der Kartenkontrolle  
beim unteren Eingang.

**CLUBINFO**  
45. Ausgabe

Kontaktadresse:  
**Club 80**  
**Hartmut Obermann**  
Wilhelm-Baur-Straße 8  
Postfach \_\_\_\_\_  
76135 Karlsruhe  
Tel.: 0721/85 40 6  
BTX: 0721/\_\_\_\_\_  
Anrufbeantworter  
FAX: 0721/85 40

# Inhaltsverzeichnis

## Hardware

Centronics-Schnittstellen autom. ...	1 - 2
Druckeranschluß für Z80-Systeme	3 - 4
Serielle Schnittstelle für 8085-System	5 - 6

## Hardware-Oldies

... aus INFO's 1-10

Der TRS-80 und die Außenwelt	7 - 8
Interface mit 8251 und 8255 für Z80-Rechner	9 - 12
Synchronisierte Taktumschaltung für VideoGenie	13 - 17
Billiger NF-Verstärker für den TRS-80	18
Umbau eines TRS-80 Modell1 16 auf 48k (64)	19 - 21
Ein sicheres Plätzchen	23 - 28
Ein sicheres Plätzchen (für TRS80)!	36
Supertape für TRS-80	29 - 36
BASICODE-2	37 - 39
BASICODE-Interface	40 - 41
Allerlei Nützliches für den TRS-80	42 - 43
Billiges CP/M von Bill Brewer	44 - 53

## Die letzten Seiten

Impressum	55
Schluß	56
Mitgliederadressenliste	Redaktion am INFO-Ende

## Autor & Seite

### Termine:

Multimedia World '94	
IVD Kongress '94	
Frankfurter Buchmesse	
Document Imaging '94	
NETWORLD+ INTEROP '94	
5. Elmshorner Computertage	
Computer '94	
LERNTEC	
TELMA	
SIMO TCI	
Networks Expo '94	
ComCoBit	
Windows Solutions	
5th CD-I World Conference	
Frankfurt	29.09.- 01.10.1994
Kassel	01.10.- 03.10.1994
Frankfurt	05.10.- 10.10.1994
Berlin	10.10.- 13.10.1994
Paris	24.10.- 28.10.1994
Elmshorn	29.10.- 30.10.1994
Köln	04.11.- 06.11.1994
Karlsruhe	08.11.- 10.11.1994
Magdeburg	10.11.- 13.11.1994
Madrid	08.11.- 13.11.1994
Frankfurt	22.11.- 24.11.1994
Fulda	26.11.- 27.11.1994
Frankfurt	29.11.- 02.12.1994
Düsseldorf	04.12.- 07.12.1994

### Bitte nicht vergessen:

Der obligatorische Hobbytronk-Sonntags-Treff am 27.11.1994 in  
Stuttgart um 9.00Uhr und/oder 12.00Uhr am unteren Eingang  
(bisher war dort immer der Conradstand gegenüber)

Redaktionsschluß für das nächste INFO ist der:  
01. Dezember 1994

# Centronics-Schnittstellen automatisch umgeschaltet

Oft ergeben sich praktische Probleme, wenn zwei Mikroprozessorsysteme einen gemeinsamen Drucker nutzen sollen. Im folgenden Beitrag wird eine Schaltung beschrieben, die das automatische und vollelektronische Umschalten zwischen zwei Centronics-Schnittstellen ermöglicht. Die Centronics-Schnittstelle ist durch acht parallele Datensignale sowie einige Steuer- bzw. Handshake-Signale definiert. Die wichtigsten davon sind STROBE, BUSY, ACKNOWLEDGE und INPUT PRIME. Diese Steuerleitungen werden in der nachfolgenden Schaltung berücksichtigt.

## Übertragungsprotokoll

Das Protokoll der Schnittstellen läßt sich anhand des Signal-Zeit-Diagramms in Bild 1 erklären.

Die Bereitschaft, Daten aufzunehmen, zeigt der Drucker durch LOW-Signal auf der Steuerleitung BUSY an. Durch das kurze LOW-Setzen des STROBE-Eingangs des Druckers werden die anstehenden Daten übernommen. Der Drucker quittiert die Übernahme über den ACKNOWLEDGE-Ausgang. Ein LOW-Impuls am Eingang INPUT PRIME setzt den Drucker zurück.

## Schaltung

Derjenige Rechner, welcher zuerst den Drucker belegen möchte, soll automatisch den Drucker zugeteilt bekommen. Jeder Rechner gibt seine Druckeranforderung durch ein logisches LOW auf der STROBE-Leitung bekannt.

Hat der akzeptierte Rechner seine Übertragung beendet, was durch das Fehlen von Datenzeichen während einer bestimmten Beobachtungszeit erkannt wird, so steht der Drucker wieder zur freien Verfügung. LEDs zeigen den jeweiligen Betriebszustand an.

Die Schaltung (Bild 2) zeigt den Fall, daß zwei Rechnersysteme mit einem gemeinsamen Drucker arbeiten. Den Kern des Umschalters bilden die drei Multiplexer IC1 bis IC3; diese schalten jeweils die acht Datenleitungen Data1 bis Data8 sowie die drei Steuerleitungen STROBE, INPUT PRIME und ACKNOWLEDGE. Die

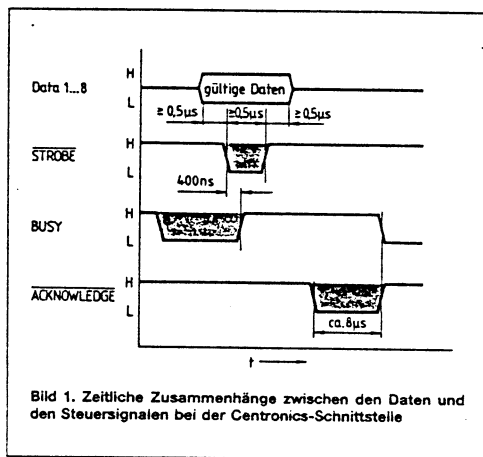
Auswahl der durchzuschaltenden Leitungen übernimmt das Flipflop IC4, das von den STROBE-Signalen der beiden Rechner gesteuert wird.

Für das Status-Signal BUSY ist die etwas aufwendigere Logik um IC5 zuständig. Wird der Drucker von keinem Rechner belegt, dann erhalten beide Rechner das Freigabesignal des Druckers (BUSY = LOW). Ist der Drucker bereit, dann wird das retriggerbare Monoflop IC7 freigegeben (Eingang B1 auf HIGH). Belegt nun einer der beiden Rechner den Drucker, indem dieser Daten und den STROBE-Impuls aussendet, geschieht folgendes:

Über das Flipflop IC4 wird der angeforderte Rechner zum Drucker durchgeschaltet und das Monoflop IC7 getriggert. Es bleibt bei der angegebenen Dimensionierung für rund 12 s aktiv.

Über die Logik IC5 erhält der jeweils andere Rechner ein aktives BUSY-Signal, was seine Datenübertragung suspendiert.

Solange der ausgewählte Rechner Daten aussendet, erhält IC7 ständig Triggerimpulse, und die Zuteilung bleibt erhalten. Erst wenn die Übertragung der Daten für



länger als etwa 12 s aussetzt, kann der andere Rechner die Zuteilung seinerseits anfordern.

Die LEDs 2 und 3 zeigen an, welcher Rechner gerade zugeteilt ist. Ist die Belegung offen, leuchten beide LEDs. Die LED1 leuchtet, wenn der Drucker nicht frei verfügbar, also BUSY ist.

Die Versorgungsspannung von 5 V für die Schaltung

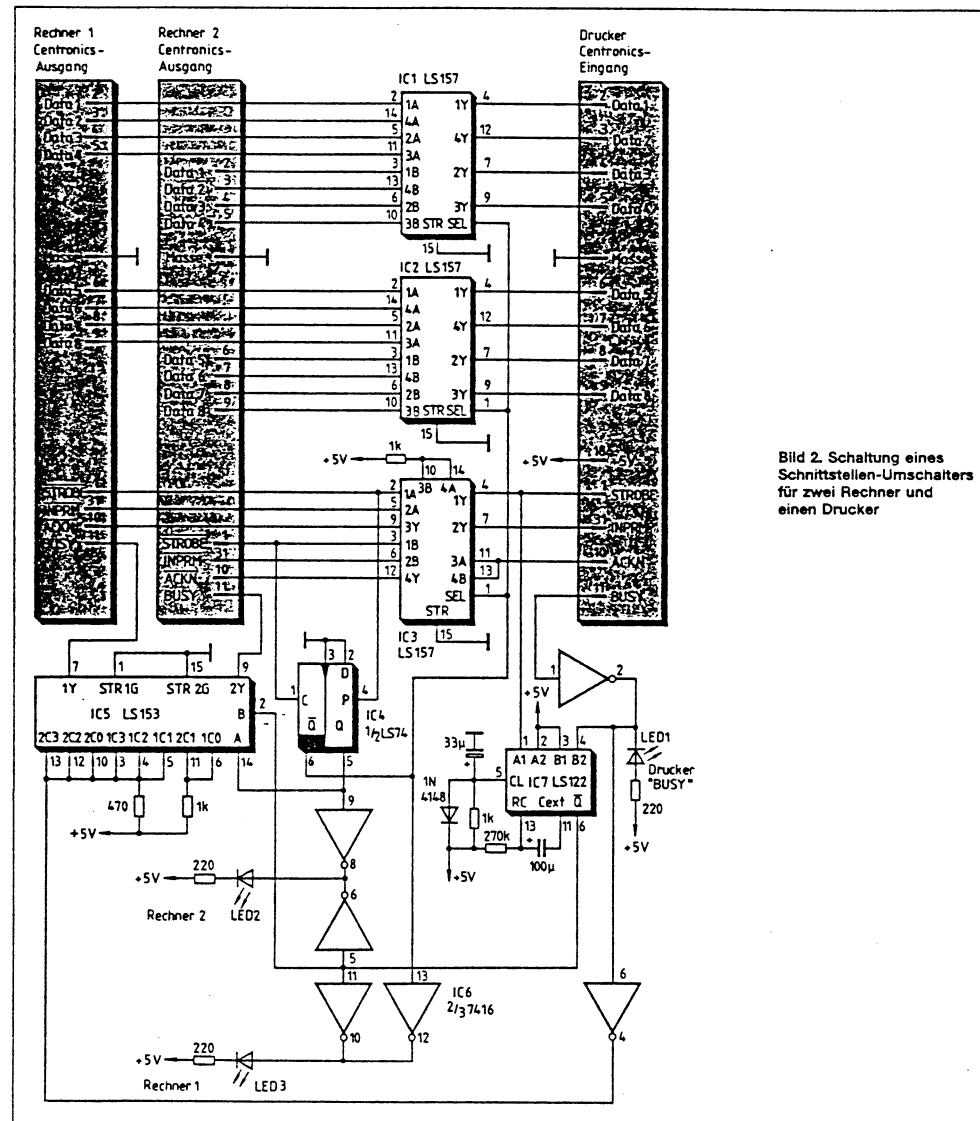
läßt sich dem Drucker über den Anschluß 18 der Centronics-Schnittstelle entnehmen.

Dipl.-Ing. K. Baderschneider

Cand. el. T. Lindner

## Literatur

- [1] Hartmann, I.: Die Centronics-Druckerschnittstelle. Micro Extra (1982), H. 2, S. 13 bis 16.
- [2] Siemens AG: Drucker PT 88/PT 89, Anwendungsbeschreibung. Firmenschrift (1983).



## Druckeranschluß für Z80-Systeme

Der Beitrag beschreibt die Realisierung einer Centronics-Schnittstelle mit dem PIO-Baustein der Z80-Familie. An Zusatzhardware sind nur noch wenige Bauteile erforderlich. Die Treiber Routinen sind ausgelegt für die Ausgabe einzelner ASCII-Zeichen bzw. von Zeichenketten, die mit ETX (03H) abgeschlossen sind.

Die Centronics-Schnittstelle ist eine parallele Schnittstelle zum Ansteuern von Druckern. Die Daten werden mit 8 Bit parallel als ASCII-Zeichen übertragen. Die Schnittstelle arbeitet mit TTL-Pegeln. Bild 1 zeigt, wie die Übertragung eines Zeichens mit Quittungssignalen gültig, d. h. eingeschungen sind, wird durch ein DATA STROBE = Low dem Drucker angezeigt, daß ein Zeichen zur Übertragung bereitsteht. Das BUSY-Signal geht auf High und zeigt dem Rechner damit, daß das Zeichen gelesen wird. Ist das Zeichen übernommen, geht das BUSY-Signal wieder auf Low, und gleichzeitig wird durch ein ACKNOWLEDGE = Low dem Rechner angezeigt, daß das Zeichen übernommen wurde. Jetzt kann das nächste Zeichen übertragen werden.

Von den vier Betriebsarten, die der Z80-PIO-Baustein bietet, ist die Betriebsart 0, Byteausgabe mit Quittungsinterrupt, für die Centronics-Schnittstelle am besten geeignet, da in diesem Fall der zweite PIO-Kanal anderweitig verwendet werden kann. Das Einstellen der Betriebsart 0 zeigt Bild 2. Die zwei Quittungsleitungen des einen Kanals reichen aus, um mit den acht Datenleitungen zusammen die Schnittstelle zu realisieren. Das Zeitdiagramm für die Betriebsart 0 zeigt Bild 3.

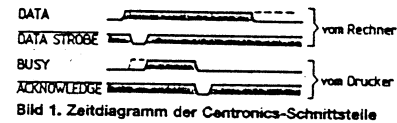
Vergleicht man die Bilder 3 und 1, so sieht man, daß aus dem READY-Signal des PIO-Bausteins das DATA-STROBE-Signal der Centronics-Schnittstelle werden muß. Das DATA-STROBE-Signal muß allerdings zurückgehalten werden, solange das BUSY-Signal des Druckers auf High liegt. Das STROBE-Signal für den PIO-Baustein stellt die Quittierung des Zeichens durch den Drucker dar und ist identisch mit dem ACKNOWLEDGE der Centronics-Schnittstelle. Bleibt nur noch eine Pufferung der Datenleitungen mit einem Treiberbaustein. Zur Erzeugung des DATA-STROBE-Signals kann man die Schaltung nach Bild 4 verwenden.

Der Baustein 74LS123 enthält zwei nachtriggerbare monostabile Kippstufen; davon wird in der Interface-

schaltung aber nur eine gebraucht, um einen DATA-STROBE-Impuls von bestimmter Dauer zu erzeugen. Da für die Schaltung aber noch ein D-Flipflop erforderlich ist, wird die zweite monostabile Kippstufe als D-Flipflop eingesetzt. Dies ist möglich, wenn man eine Zeitkonstante  $T \rightarrow \infty$  realisiert. Dann läßt sich das umfunktionierte Flipflop mit einer positiven Flanke am Triggereingang setzen (Q geht von 0 auf 1). Das Flipflop bleibt gesetzt, bis es über seinen R-Eingang ( $\bar{R}$  von 1 auf 0) wieder zurückgesetzt wird.

Die Zeitkonstante  $T \rightarrow \infty$  läßt sich sehr einfach realisieren, indem man den Widerstand  $R_{ext}$  und den Kondensator  $C_{ext}$  wegläßt. Wird statt des 74LS123 der 74123 verwendet, ist ein Widerstand von etwa 10 k $\Omega$  zwischen den Anschlüssen J (7) und Q (12) erforderlich.

Damit das auf den R-Eingang rückgeführte DATA-STROBE-Signal keine Nachtriggerung über die interne



PRGPI0 LD A, 0FH ; Steuervort für Betriebsart 0  
OUT (PIOCHA+2), A  
LD A, INTVEK ; Interruptvektor  
OUT (PIOCHA+2), A  
LD A, 87H ; Interrupt freigeben  
OUT (PIOCHA+2), A

Bild 2. So wird die Betriebsart 0 des PIO-Bausteins eingestellt. PIOCHA ist die Adresse des Kanals A

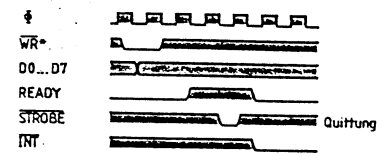
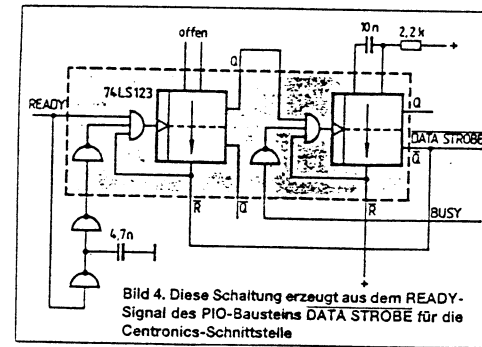


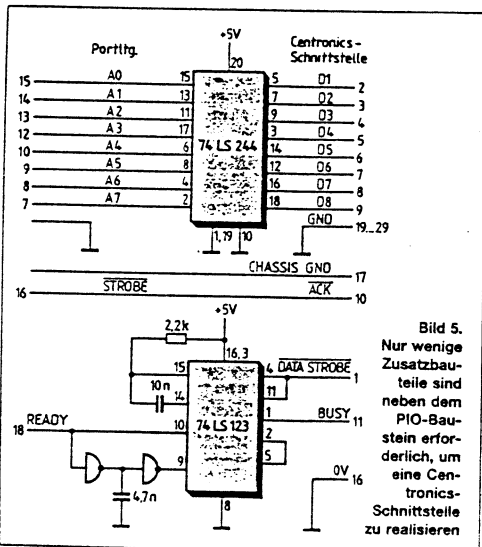
Bild 3. Dieser Zeitablauf ergibt sich in der PIO-Betriebsart 0



Verbindung von  $\bar{R}$  zu dem UND-Glied bewirken kann, wird das READY-Signal über eine Differenzierschaltung, auf den Eingang des Flipflops geführt. Dadurch ist der Triggereingang bereits gesperrt, wenn etwas später das DATA-STROBE-Signal mit der positiven Flanke ansteht. Bild 5 zeigt die Gesamtschaltung der Zusatzhardware.

Bevor ein ASCII-Zeichen an den Drucker ausgegeben wird, muß sichergestellt sein, daß der Drucker das vorhergehende ASCII-Zeichen angenommen, d. h. quittiert hat. Zu diesem Zweck wird ein Flag definiert, das sogenannte Druckerquittungsbit DQB. Das DQB ist einfach ein beliebiges Bit einer RAM-Adresse des zur Verfügung stehenden Speichers.

Ist das DQB nicht zurückgesetzt, wird es in einer Warteschleife so lange getestet, bis es durch einen Druck-



kerinterrupt – der als Quittierung des empfangenen ASCII-Zeichens dient – zurückgesetzt wird. Beim Einschalten des Systems muß das DQB während der Initialisierung mit 0 vorbesetzt werden, damit das erste ASCII-Zeichen überhaupt ausgegeben werden kann.

Ist das DQB durch einen Druckerinterrupt bzw. durch die Initialisierung zurückgesetzt, wird es wieder gesetzt, und das Zeitglied zur Überwachung des Druckers wird angestoßen. Dann wird das ASCII-Zeichen an den Drucker ausgegeben.

Übernimmt der Drucker die zu druckenden ASCII-Zeichen in einen Puffer, so erfolgt der Druckvorgang erst mit Ausgabe des Zeichens CR (0DH) an den Drucker.

Die Routine zur Quittungsinterruptbehandlung wird aufgerufen, wenn der Drucker das empfangene ASCII-Zeichen quittiert. Hierbei werden das Zeitglied zur Druckerüberwachung und das DQB zurückgesetzt, danach wird der Interrupt wieder freigegeben.

Läuft das bei der Ausgabe eines ASCII-Zeichens an den Drucker angestoßene Zeitglied ab, so erzeugt es einen Interrupt, der zu einer Fehlermeldung und zu einem Ausstieg aus dem Druckerprogramm genutzt werden kann. Als Zeitglied kann z. B. ein Z80-CTC-Baustein (zwei Kanäle kaskadiert, der erste Kanal als Zeitgeber, der zweite Kanal als Zähler) benutzt werden. Die Laufzeit des Zeitglieds muß auf die Druckgeschwindigkeit des Druckers abgestimmt werden, d. h. sie sollte etwas größer sein, als für den Ausdruck einer Zeile benötigt wird.

Steht in dem vorhandenen Z80-System kein CTC-Baustein zur Verfügung, so kann für die Druckerüberwachung auch ein Schleifenzähler (Software) in die Warteschleife des DQB-Tests eingefügt werden. Wird ein bestimmter Zählerstand erreicht, verzweigt das Programm zu einer Fehlermeldung und steigt aus dem Druckerprogramm aus. Der Schleifenzähler muß vor jedem auszugebenden ASCII-Zeichen auf seinen Startwert gesetzt werden.

Die auszugebenden ASCII-Zeichen stehen in einem Speicherbereich (Puffer), der mit ETX (03H) abgeschlossen ist. Der Zeiger HL zeigt auf das erste zu druckende Zeichen im Puffer.

Die Pufferausgabe läuft nach folgendem Schema ab: Der Akku wird aus der Speicherstelle geladen, auf die HL zeigt. Ist das Zeichen ein ETX, so erfolgt ein Rücksprung ins Hauptprogramm, andernfalls wird die Routine zur Ausgabe eines ASCII-Zeichens an den Drucker aufgerufen und danach der Zeiger HL inkrementiert. Der Vorgang wiederholt sich so lange, bis ein ETX auftritt.

Manfred Scherer

### Literatur

- [1] Digitale Schaltungen MOS. Datenbuch der Siemens AG, Bereich Bauelemente, 1974/75.
- [2] Z80-PIO Technical Manual, Zilog Incorporated, 1977.
- [3] Z80-CTC Counter/Timer Circuit Technical Manual, Zilog Incorporated, 1977.
- [4] Technical Manual Z80 Microprocessor Family, SGS-Ates, 1983.
- [5] RX-80-Bedienungshandbuch, EPSON Deutschland GmbH, Düsseldorf.

# Serielle Schnittstelle für 8085-System

Mikrocomputer werden in vielen Fällen über Datensichtgeräte bedient. Dabei genügt als Verbindung eine einfache Schnittstelle nach der V.24-Norm. Der Beitrag zeigt, wie man ein 8085-System um ein solches Interface erweitert.

## 1 Hardware

Einfache Datensichtgeräte (DSG) sind meist mit einer V.24-Schnittstelle ausgerüstet, die im Start-Stop-Betrieb und mit acht Datenbits pro Zeichen betrieben wird. Die Darstellung der Zeichen erfolgt nach dem ISO-7-Bit-Code [4]. Das achte Datenbit dient dabei als Paritätsbit, wobei nach gerader und ungerader Parität unterschieden wird (gerade Parität bedeutet: Anzahl der 1-Bits in den acht Datenbits ist gerade, das Paritätsbit ist entsprechend zu setzen).

Sollen Zeichen zwischen Mikrocomputer und Sichtgerät übertragen werden, dann muß ein E/A-Baustein die parallel auf dem Datenbus anliegende Information in einen seriellen Bitstrom umsetzen und umgekehrt. Diese Aufgabe kann der Baustein 8251A übernehmen [5]. Bild 1 zeigt, wie der SIO-Chip (Serial Input/Output) an die CPU angeschlossen wird. Als Mikroprozessor dient der 8085. Für diesen Prozessor, wie auch für den 8080, umfaßt der E/A-Adreßraum die Adressen 0...255 (00H...FFH), die an den Adreßleitungen A0...A7 anliegen. Im folgenden wird angenommen, daß die E/A-Adressen 3EH und 3FH für die zu realisierende V.24-Schnittstelle verfügbar sind.

Die Bausteinauswahl (CS-Chip Select) erfolgt über die Adreßbits A1...A7 und das Steuersignal IO/M (IO/M = 1 bedeutet: E/A-Adresse auf Adreßbus) nach der Beziehung:

$$\overline{CS} = \overline{A_7} \overline{A_6} A_5 A_4 A_3 A_2 A_1 \overline{IO/M}$$

Diese Beziehung kann leicht durch ein NAND-Gatter mit acht Eingängen realisiert werden.

Der Baudratengenerator hat die Aufgabe, die Übertragungsgeschwindigkeit für den seriellen Bitstrom auf den Leitungen TxD und RxD festzulegen. Zur Schaltungsvereinfachung wird ihm der Systemtakt des Mikrocom-

puters zugeleitet und entsprechend heruntergeteilt. In diesem Beispiel beträgt der Systemtakt 3072 kHz, er wird auf die in Bild 2 angegebenen Frequenzen geteilt.

## 2 Software

Beim Baustein 8251A ist eine Reihe von Parametern programmierbar: z. B. die Betriebsart (asynchron oder synchron), die Anzahl der Bits pro Zeichen und die Paritätsprüfung (ja/nein oder gerade/ungerade).

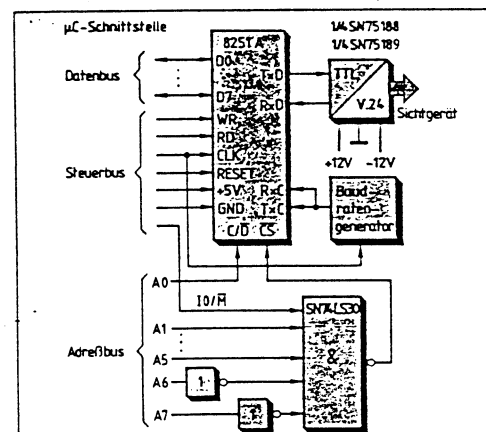


Bild 1. So wird der SIO-Baustein an den Mikrocomputer angeschlossen

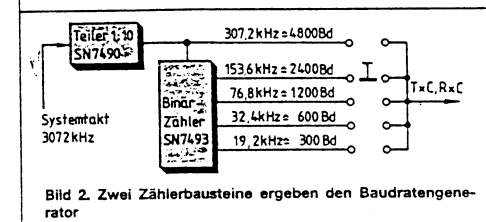


Bild 2. Zwei Zählerbausteine ergeben den Baudratengenerator

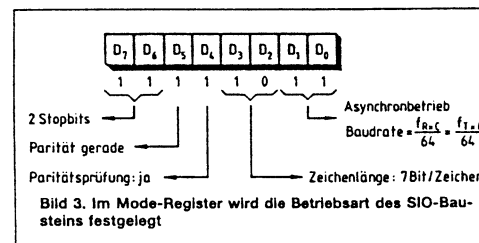


Bild 3. Im Mode-Register wird die Betriebsart des SIO-Bausteins festgelegt

Außerdem besteht jederzeit die Möglichkeit, seinen Zustand durch den Prozessor abzufragen, z. B. das Signal am Steueranschluß DSR (DSR = Data Set Ready, Betriebsbereitschaft).

Bevor man Zeichen empfangen oder senden kann, muß der 8251A zurückgesetzt und für ein bestimmtes Datenübertragungsverfahren programmiert werden. Das Rücksetzen übernimmt das Resetsignal des Mikroprozessors. Da das verwendete Sichtgerät nur die Betriebsart „asynchron“ zuläßt, sind nach [5] zwei Steuerwörter (Steuerbytes) vom Prozessor an den Baustein zu übergeben. Anschließend können Zeichen gesendet und empfangen werden. Dazu übergibt der Prozessor dem SIO Zeichen in einem Byte über den Datenbus als Parallelinformation und holt Zeichen in Parallelendarstellung wieder ab.

Die Übergabe der beiden Steuerbytes muß in einer festen Reihenfolge geschehen. Zuerst ist das sogenannte Mode-Wort zu übertragen. Die Bedeutung der einzelnen Bits für den Asynchronbetrieb ist in Bild 3 erläutert.

Das zweite Steuerwort erlaubt unter anderem die Freigabe und das Sperren von Sender und Empfänger und wählt verschiedene Hilfsfunktionen an. Dieses Steuerwort wird als Kommandowort bezeichnet (Bild 4).

Der SIO muß Steuerwörter von den „wirklichen“ Bildschirmaten unterscheiden. Diese Unterscheidung wird durch ein spezielles Eingangssignal (C/D) getroffen. Ist es log. 1, dann wird ein Steuerwort vom Datenbus an den SIO oder umgekehrt geschickt. C/D = 0 entspricht der Übergabe von Nutzdaten. Wie aus Bild 1 hervorgeht, ist dieser Eingang mit der Adreßleitung A0 verbunden, d. h. Daten und Steuerinformationen unterscheiden sich durch die Verwendung unterschiedlicher E/A-Adressen.

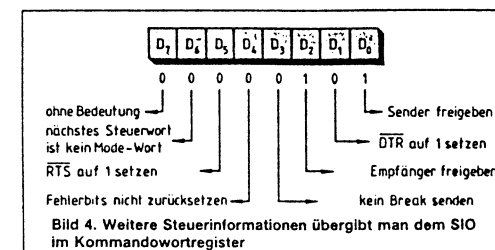


Bild 4. Weitere Steuerinformationen übergibt man dem SIO im Kommandowortregister

Aus der Beschaltung des  $\overline{CS}$ -Eingangs für die E/A-Adresse ergibt sich der Wert 3EH für Daten und der Wert 3FH für Steuerinformationen.

Damit wird der Baustein mit folgenden Befehlen auf seine Aufgabe vorbereitet:

```
MVI A,0FBH ;Mode-Wort in
OUT 3FH ;8251A einschreiben
MVI A,05H ;Kommandowort
OUT 3FH ;in 8251A einschreiben
```

Empfangen eines Zeichens:

```
CIN: IN 3FH ;Statuswort in Akku
ANI 02H ;D1 im Statusw. gesetzt?
JZ CIN ;nein, nochmal erfragen
IN 3EH ;Empf. Zeichen in Akku
```

Anschließend ist die eigentliche Datenübertragung möglich. Hierbei muß allerdings beachtet werden, daß der 8251A nur dann ein neues Zeichen aufnehmen kann, wenn das Signal T×RDY (Transmitter Ready, Sender bereit) log. 1 führt. Dieses Signal läßt sich über das sogenannte Statuswort abfragen. Das gleiche gilt auch für das Empfangen von Zeichen. Hierbei ist das Signal R×RDY (Receiver Ready, Empfänger bereit) abzufragen. Es informiert den Prozessor, ob ein empfangenes Zeichen abgeholt werden kann.

Die möglichen Statusabfragen sind in Bild 5 näher erläutert. Für einfache Anwendungen genügt die Auswertung von D1 und D0, das verdeutlichen folgende Programmzeilen.

Ausgabe des Zeichens ‚A‘ mit vorheriger Statusabfrage:

```
COU: IN 3FH ;Statuswort in Akku
ANI 01H ;D0 im Statuswort gesetzt?
JZ COU ;nein, nochmal erfragen
MVI A,41H ;Zeichen ‚A‘ in Akku
OUT 3EH ;und zum 8251A übertragen
```

Ing. (grad.) Karl-Alfred Knapp

## Literatur

- [1] DIN 66020, Datenübertragung – Anforderungen an die Schnittstelle bei Übergabe bipolarer Datensignale.
- [2] CCITT-Empfehlungen der V-Serie und der X-Serie, Band 2, Datenübertragung im Fernsprechnetz. R. v. Döcker's Verlag, 4. Auflage, 1982.
- [3] ELEKTRONIK-Arbeitsblatt Nr. 124, Die V- und X-Empfehlungen des CCITT. ELEKTRONIK 1979, H. 21, S. 87...90.
- [4] DIN 66003, Informationsverarbeitung – 7-Bit-Code.
- [5] Mikrocomputer-Bausteine, Band 3: Peripherie. Fa. Siemens.
- [6] Experimentiercomputer ECB85, Bedienungsanleitung, Ausgabe 11.81. Fa. Siemens.

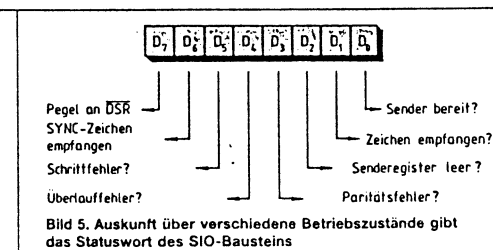


Bild 5. Auskunft über verschiedene Betriebszustände gibt das Statuswort des SIO-Bausteins



## INTERFACE MIT 8251 UND 8255 FÜR Z80-RECHNER

Das Interface wurde entwickelt, um beliebige Geräte mit dem Computer verbinden zu können.

Es gibt bereits folgende Anwendungen:

Eprom - Programmierer  
Joystick Anschluß  
Morsezeichen - Decoder  
Morsetrainer

Funkfern schreiben

Steuerung von el. Verbrauchern über Relais

Bei der Entwicklung wurde darauf Bedacht genommen, daß es nicht nur in Maschinensprache, sondern auch in Basic leicht programmiert werden kann.

Bei Verbindung der ser. Schnittstelle mit RS-232 od. V-24 ist zu beachten, daß der 8251 TTL - Signale verarbeitet. Sollte die Schnittstelle mit EIA-Pegeln ( $\pm 3 \dots 15$  Volt) arbeiten, sind noch entsprechende Line-Driver nachzuschalten. (z.B.: 75189)

Literatur zur Programmierung dieser Bausteine gibt es bei Siemens: Die Beschreibungen der SAB 8080-Serie

## SCHALTUNGSBESCHREIBUNG

Der Adreß - und Datenbus wird aus dem Computer über ein Kabel entnommen. Die Steckerbezeichnungen gelten für den KOMTEK. Bei anderen Geräten bitte im User-Manual nachsehen. Direkt am Eingang ins Interface sind alle Bus-Leitungen mit je 3,9 k nach Masse terminiert. Diese Widerstände habe ich im Schaltbild nicht eingezeichnet.

Alle Signale sind mit 74245 voll gepuffert, sodaß Rückwirkungen auf den Computer ausgeschlossen sind.

Das Signal I/O request ist mit einem 74121 um ca. 0.4 usec verzögert. Das ist notwendig bei Floppy-Betrieb, um die Timer-Interrupts vom Interface fernzuhalten.

Das Signal 'OSZ 2 MHz' an Pin 45 ist in der Frequenz völlig unkritisch. Notfalls lässt sich auch die Leitung 'MI' verwenden. Diese Clock ist nur zur internen Steuerung des 8251 notwendig.

Die Adreß-Decodierung besorgt ein 74138. Die Ausgänge liegen an Lötstiften: sodaß eine leichte Änderung möglich ist. Originalbelegung:

PIO 8255: 7C = Port A  
7D = Port B  
7E = Port C  
7F = Control-Register (für Steuerung der Ports)  
SIO 8251: 64 = Datenregister  
66 = Steuer- und Statusregister  
Timer : 60 = Steuerwort für Baudrate (1,2,4,8)

## AUFBAU UND ABGLEICH

Die Verdrahtung habe ich auf einer Veroboard-Platte in der Größe einer Europa-Karte (160 X 100 mm) vorgenommen.

Es wurde die sogenannte Fädeltechnik angewandt, die für ein Einzelstück die schnellste Aufbauart darstellt.

Bei einer größeren Stückzahl würde sich der Entwurf einer Platine durchaus lohnen. Es wurde allerdings noch kein Layout dafür erstellt.

Einstellung:

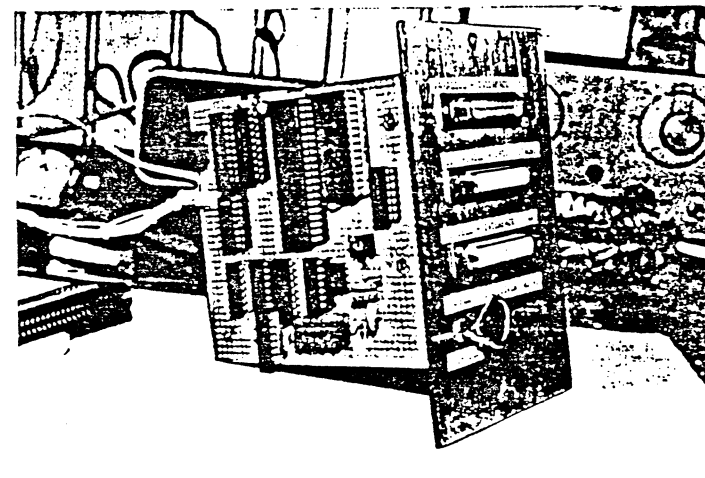
Der Timer kann mit einem OUT-Befehl auf 4 verschiedene Takt-Raten eingestellt werden. Diese ist am Punkt MP mit einem Frequenzzähler meßbar. Im Schaltbild sind die Werte für 45,110,300.1200 Baud angegeben. Durch Wahl der Potis und ggf der Vorwiderstände läßt sich jede beliebige Frequenz einstellen. Formel: Frequenz=Baudrate X 16 oder X64 oder X1 je nach Steuerwort im 8251.

Das Poti am NE555 bestimmt das Tastverhältnis. Es wird so eingestellt, daß bei der niedrigsten und auch bei der höchsten Taktrate ein sicheres Schalten erfolgt. (nicht sehr kritisch, aber etwas Einfluß auf die Frequenz)

Einstellung des Trimmers am 74121: Oszilloskop mit I/O request triggern. Signal am MP2 auf 0.4 usec Verzögerung mit dem Trimpoti einstellen. (Fallende Flanke)

Ohne Oszi: Trimmer auf Mittelstellung

Walter Zwickel  
Lengfelden 123  
A5101-Bergheim









## Synchronisierte Taktumschaltung für VideoGenie

Liebe Clubfreunde,

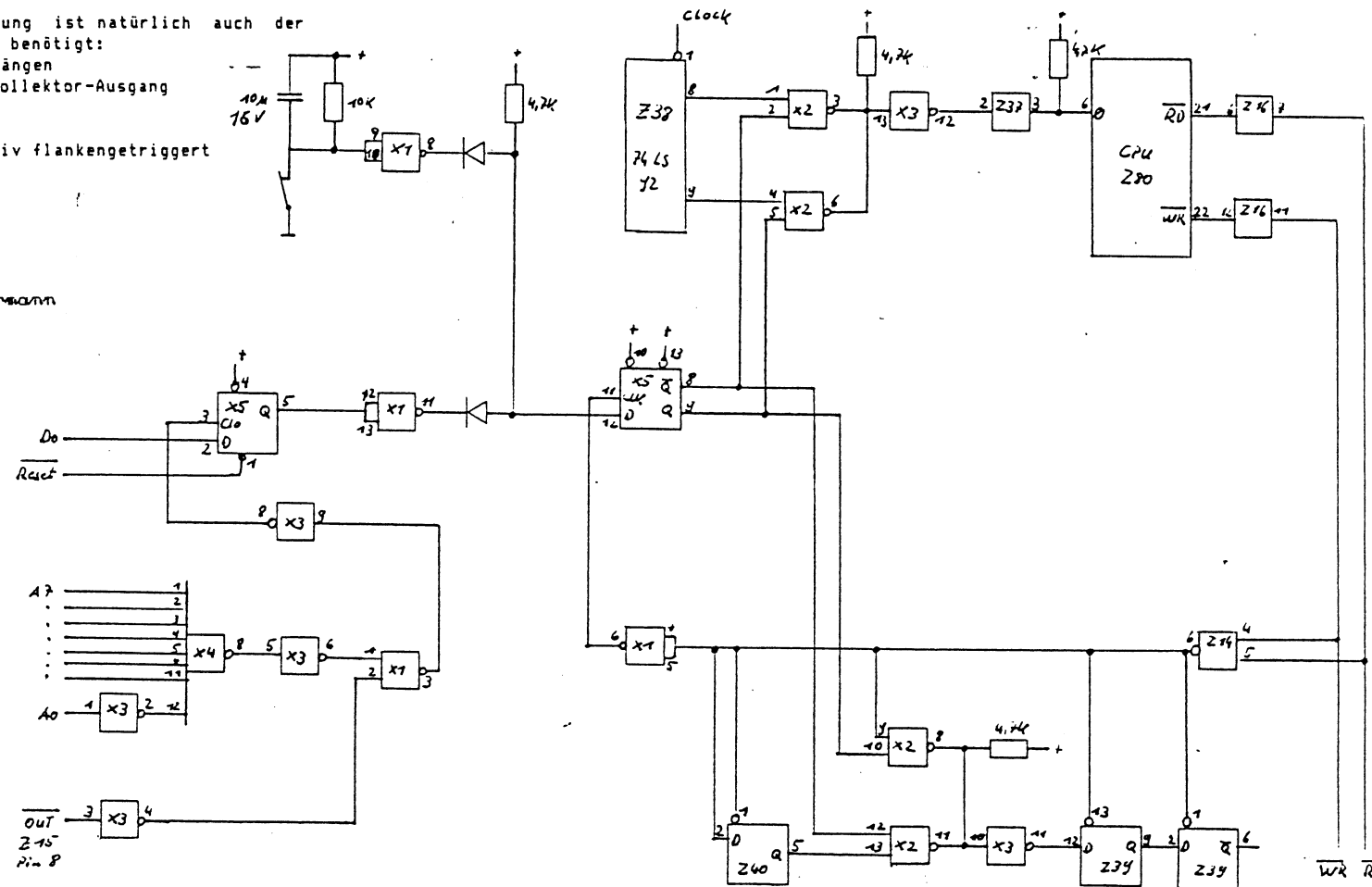
Im Clubinfo Nr. 5 hat Gerald Dreyer eine Schaltung veröffentlicht, mit der man die Taktfrequenz des VideoGenie verdoppeln kann. Nachteil dieser Schaltung ist die Tatsache, daß bei der Umschaltung zwischen einfacher und doppelter Taktfrequenz in den meisten Fällen der Rechner hängenbleibt. Außerdem kann man die Taktfrequenz nur mit einem Schalter, also hardwaremäßig, jedoch nicht softwaremäßig umschalten. Um diese Nachteile zu beseitigen, hat Manfred Held die unten abgebildete Schaltung entworfen.

Die eigentliche Umschaltung der Taktfrequenz bewerkstelligen zwei NAND-Gatter (x2, rechts neben Z38). Die Umschaltung kann durch einen Schalter (linke obere Ecke) oder über den Port FEh (254d) (Decodierlogik linke untere Ecke) erfolgen. Um einen Systemabsturz während der Umschaltung zu verhindern, wird durch Z14 sichergestellt, daß der Umschaltvorgang erst erfolgt, wenn die CPU weder Schreib- noch Leseoperationen durchführt.

Durch die erweiterten Möglichkeiten der Schaltung ist natürlich auch der Bauteileaufwand gestiegen. Es werden folgende Teile benötigt:

- IC x1 = 74LS00 = vierfach NAND-Gatter mit je 2 Eingängen
- IC x2 = 74LS03 = vierfach NAND-Gatter mit offenem Kollektor-Ausgang
- IC x3 = 74LS04 = sechsfach Inverter
- IC x4 = 74LS30 = NAND-Gatter mit 8 Eingängen
- IC x5 = 74LS74 = zweifach Speicher-Flip-Flop, positiv flankengetriggert
- Alle IC's je 1 mal. Weiterhin:
- 2\* 1N4148 = Siliziumdiode
- 3\* 4700  $\Omega$  Widerstand
- 1\* 10 k $\Omega$  Widerstand
- 1\* 10  $\mu$ F/16V Kondensator

Viel Spaß beim Basteln, wünscht euch Hartmut Obermann



Eine Operateuse aus Wien  
dacht' beim Arbeiten stets nur an „Ihn“:  
Ihr Computer, voll Frust,  
verlor deshalb die Lust –  
er war auch nur ein Mensch, wie es schien.

\*\*\*\*\*  
\* Ergaenzung und Aenderung \*  
\*\*\*\*\*

# Synchronisierte Taktumschaltung fuer Videogenie

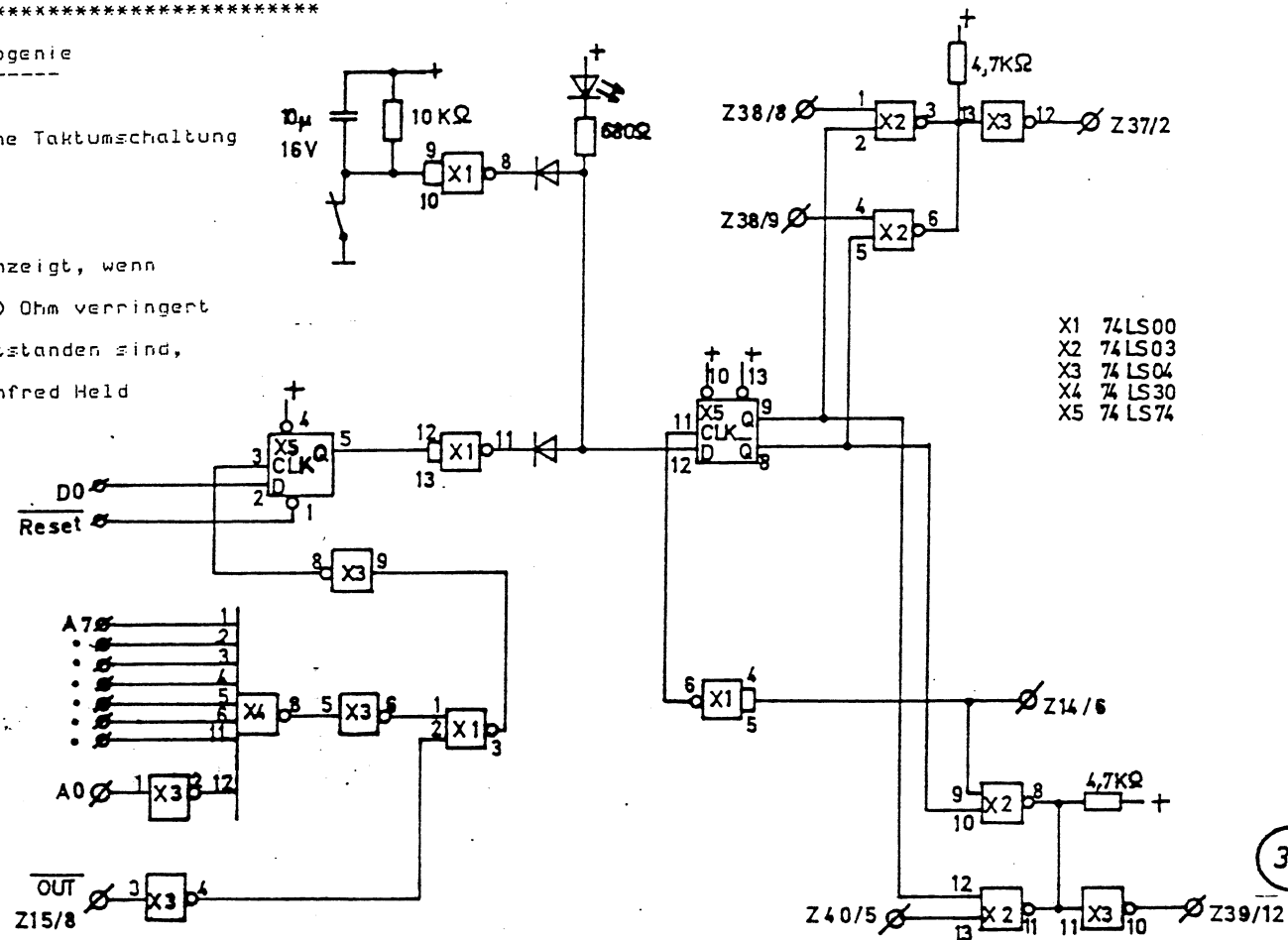
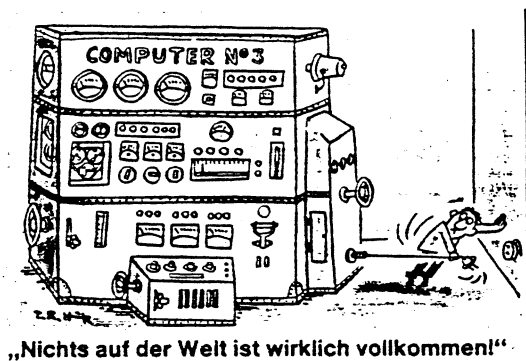
In der Clubinfo Nr. 8 ist der Schaltplan fuer eine Taktumschaltung fuer das Videogenie, der zwei Fehler enthaelt.

- Am Ic X3 ist Pin 10 und Pin 11 vertauscht.
- Am Ic X5 ist Pin 8 und Pin 9 vertauscht.

Als ergaenzung ist eine LED hinzugekommen, die anzeigt, wenn die CPU mit 3.55 MHz arbeitet.  
Bedingt durch die LED muss der Widerstand auf 680 Ohm verringert werden.

Die Fehler, die beim Zeichnen des Schaltplans entstanden sind, bitte ich zu entschuldigen.

Manfred Held



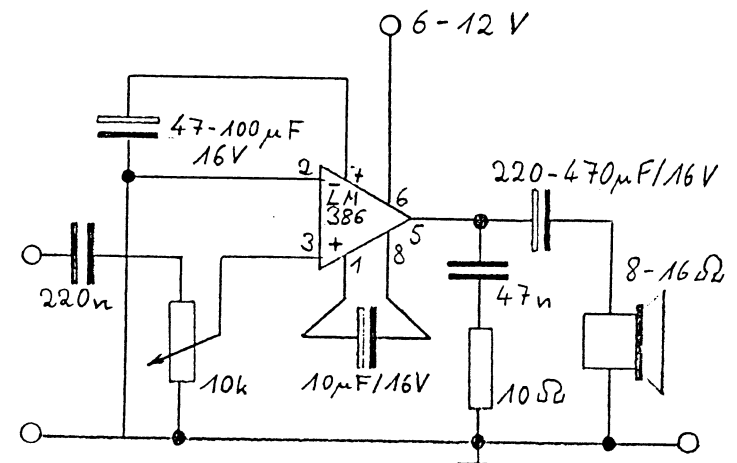
Ergänzend zu den Artikeln ueber die synchronisierte Taktumschaltung in den Clubinfo's 8 und 9 moechte ich hier eine Aenderung vorstellen in der die Anschlusspunkte fuer den TRS-80 Modell 1 angegeben sind. Weiterhin moechte ich noch hinzufuegen das die Leiterbahnen zwischen Z 56/8 und Z 72/12 und zwischen Z 69/5 und Z 69/12 zu trennen sind.

Hand-drawn schematic diagram of a digital circuit. The circuit includes a 16V battery, a 5V supply, and various logic gates and flip-flops. Inputs include Z 76/11, Z 52/10, A7, A6, A5, A4, A3, A2, A1, A0, and Z 22/3. Outputs include Z 72/12, Z 74/3, and Z 69/12. The circuit is powered by a 5V supply and a 16V battery.

Legend:

- X1 74LS00
- X2 74LS03
- X3 74LS04
- X4 74LS30
- X5 74LS74

Die nachfolgend aufgeführte Schaltung mit dem LM 386, einem NF-Verstärker-IC kleiner Leistung, kann direkt an den Tapeausgang des TRS 80 oder eines anderen Computers geschaltet werden. Sie macht den Anschluss eines grossen Verstärkers zur Tonwiedergabe (z.B.: Musikprog., Spiele) überflüssig. Der Ruhestromverbrauch liegt bei ca. 3 mA. Die Ausgangsleistung beträgt bei der angegebenen Beschaltung ca. 500 mW an einem 16 Ohm Lautsprecher. Es wird eine Versorgungsspannung von 5 - 12 Volt benötigt. Ich habe diese der Stromversorgung des Monitors entnommen in dem die Schaltung auch untergebracht ist. Die Wiedergabequalität ist zwar nicht überragend aber ausreichend.



Club 80  
INFO 45  
Sept. 94  
  
Seite 17

Club 80  
INFO 45  
Sept. 94  
  
Seite 18

Nachdem ich im Sommer des letzten Jahres einen gebrauchten TRS-80 Modell 1 gekauft hatte, der mit 16 k RAM bestueckt war, stand ich vor dem Problem diesen Rechner auf 48 k RAM aufzuruesten.

Zu diesem Zeitpunkt hatte ich schon einen Genie-I mit 48-k (64-k) RAM, und ich dachte das es kein groesseres Problem sein sollte auch den Tandy auf die gewuenschte Speicherkapazitaet zu bringen. Aber erstens kommt es anders, und zweitens als man denkt.

Da ich die technischen Handbuecher fuer beide Geraete besitze verglich ich erst einmal die Schaltunterlagen von beiden Rechnern. Auf den ersten Blick sahen auch beide Schaltplaene sehr aehnlich aus, aber gerade beim interessanten Teil, naemlich dem Adressdekoder fuer die RAM's und ROM's unterschieden sich beide Geraete erhaeblich. So musste ich feststellen das der Genie-Computer von seiner Grundauelegung her schon fuer den Umbau von 16-k auf 48-k mit den neueren 64-k-RAM Speicherchips ausgelegt war, waehrend der TRS-80 einen voellig anderen Adressdekoder hatte, der sich auf den ersten und zweiten Blick ueberhaupt nicht fuer diesen Umbau eignete.

Es musste also ein mehr oder minder neuer Adressdekoder eingebaut werden. Ich hatte zuerst vor, den Dekoder des Genie-I in den TRS-80 einzubauen. Aber auch dies erwies sich als nicht so einfach machbar, da sich beide Rechner in der Ansteuerung der ROM's und des Bildspeichers stark unterschieden.

Also was tun ?

Ich entschied mich dazu die ROM und Bildspeicheradressierung so zu lassen wie sie war und nur die Ansteuerung der RAM's zu aendern.

Zuerst zum Adressdekoder Z21. Dieser Baustein zerlegt die unteren 32K, die die ROM's, Bildspeicher, Tastatur und die alten 16 K RAM enthalten in Bloecke zu je 4K. Die Ansteuerung der alten RAM's konnte ja nun wegfallen, also machte ich folgende Ueberlegung, die neue RAM's sollen nicht angesprochen werden, wenn auf die unteren 16K zugegriffen wird, bei einer Adresse oberhalb von 3FFFH muss der RAM Bereich angesprochen werden.

Dies wird nun so erreicht: Wenn einer der ersten 4 Ausgaenge von Z21 LOW wird, so bedeutet dies, das auf die unteren 16K zugegriffen wird, also darf in diesem Fall der RAM Bereich nicht angesprochen werden.

Ich nahm also ein 4-fach-NAND und legte die Eingaaenge dieses Chips auf die 4 Ausgaenge des Z21 (Pin 9 - 12), den Ausgang des NAND legte ich auf X3 Pin 15 von wo aus die RAM-Select. Leitung abgeht (RAM \*). Ist diese Leitung LOW, so werden die RAM-Bausteine selectiert, und zwar geht das Signal ueber X71 Pin 12 - Pin 5 auf Z67 Pin 15, den Select-Eingang eines Buffers ueber den MCAS geschaltet wird.

Somit war das Ziel der Adressierung schon erreicht. Wird eine Adresse der ROM's des Bildspeichers oder der Tastatur (0000H - 3FFFH) angesprochen, so wird der RAM-Bereich nicht selectiert, wird aber eine Adresse ueber 3FFFH (4000H - FFFFH) angesprochen so wird der RAM-Bereich selectiert.

Nun aber zum praktischen Teil des Umbau's. Als erstes mussten die alten RAM's (4116) gegen die neuen RAM's (4164) ausgetauscht werden.

Zuerst musste die Spannungsversorgung der RAM's geaendert werden.

Tabelle 1:

4116 -> :	Pin 1	—	-5 V
	Pin 8	—	+12 V
	Pin 9	—	+5 V
	Pin 16	—	GND
4164 -> :	Pin 1	—	N.C.
	Pin 8	—	+5 V
	Pin 9	—	A 7
	Pin 16	—	GND

Wie aus dieser Tabelle zu ersehen muss der Pin 1 freigemacht werden. Auf der Platine sind die einzelnen Pins der 8 Speicher IC's durch Leiterbahnen miteinander verbunden. Also wurde festgestellt von welchem IC aus die Leiterbahn als naechstes musste aus dem Stromversorgungsanschluss Pin 9 der Adresseingang A 7 gemacht werden. Dies geschah aehnlich dem Umbau von Pin 1. Zuerst stellte ich fest, von wo aus die Versorgungsspannung aus dem Netzteil kam, dann trennte ich diese Leiterbahn auf, entfernte noch die Siebkondensatoren (100nF) die auf dieser Leiterbahn gegen Masse lagen und dann wurde der Pin 9 der Speicher IC's mit dem Adressmultiplexer Z51 Pin 2 verbunden. Damit der Adressmultiplexer auch die Adresse A 7 erzeugen kann muss er auch etwas modifiziert werden. Es muessen die Adressleitungen A 14 und A 15 auf die Pins 14 und 15 gelegt werden. Diese drei Anschluesse sind normalerweise nicht belegt.

Als letztes musste noch der Pin 8 der Speicher-IC's von der +12 V Versorgung getrennt, und mit der +5 V Versorgung verbunden werden. Dies war fast der schlimmste Teil des ganzen Umbaus, da an jedem einzelnen der 8 Speicher-IC's der Pin 8 von den Leiterbahnen getrennt werden musste und die neuen Anschluesse mit Schaltlitze auf der Platine nachverdrahtet werden musste.

Als letztes wurden dann die neuen Speicher-Bausteine in die Fassungen eingesetzt. Allerdings ist zu empfehlen, vor dem Einbau der IC's den Rechner schon einmal einzuschalten und die Spannungen an den IC-Sockeln zu pruefen, ob auch alle Aenderungen korrekt sind.

Dann koennen die neuen Speicherbausteine getestet werden. Also Rechner einschalten. Es sollte jetzt wie gewohnt 'MEM-SIZE' auf dem Bildschirm erscheinen. Nach dem Iruecken der 'ENTER'-Taste dauert der Speicherausbautest jetzt etwa dreimal solange wie bisher. Also keine Panik wenn es jetzt etwas laenger dauert bis 'READY' erscheint. Nachdem der Rechner sich mit 'READY' gemeldet hat, sollte man 'PRINT MEM' eingeben und die 'ENTER'-Taste druecken. Wenn alles richtig geklappt hat, sollte jetzt die Zahl '48340' auf dem Bildschirm erscheinen.

Jetzt noch ein paar Tips zu den Speicher-Bausteinen selbst :

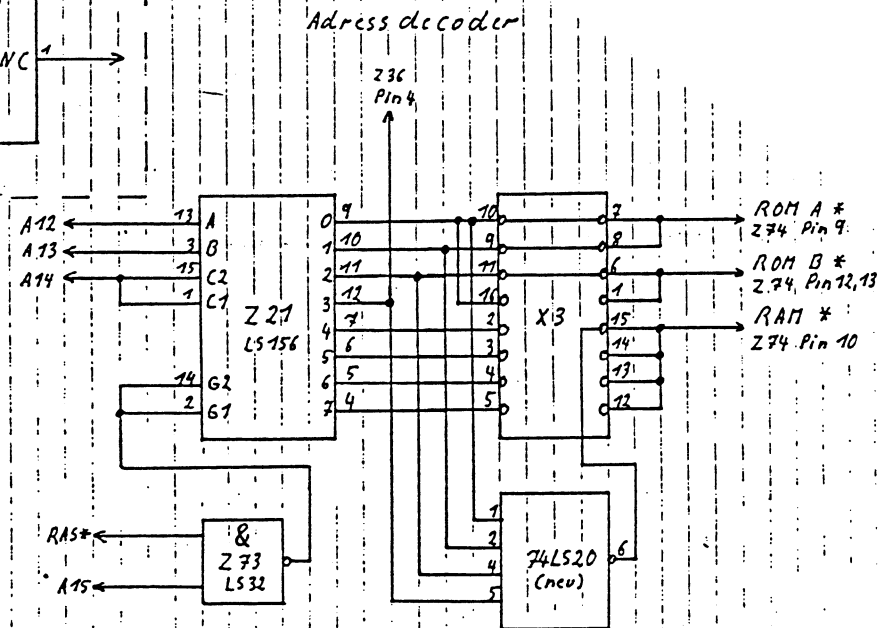
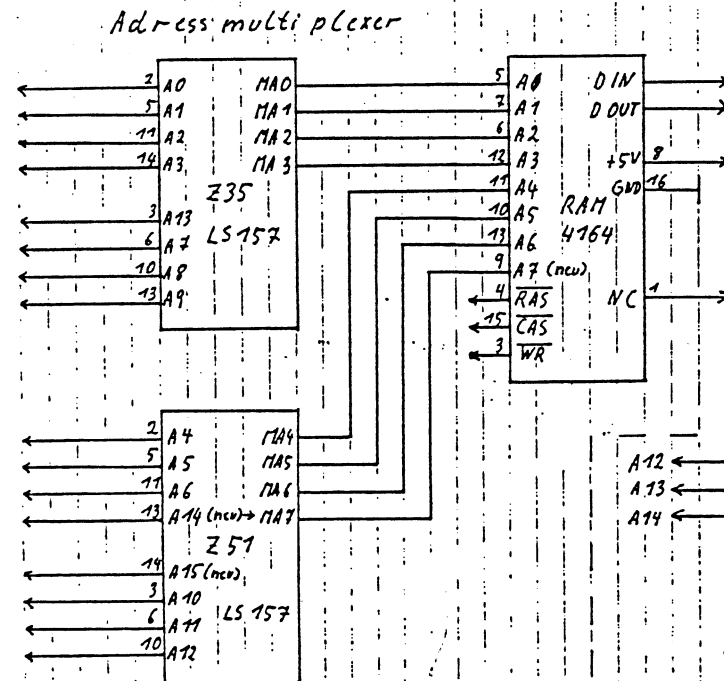
Bei meinem ersten Umbau klappte alles wunderbar, aber nach ein paar Tagen als ich ein ziemlich grosses Programm im Speicher hatte fing mein Rechner ploetzlich an zu Spinnen. Einer der auftretenden Effekte war ein sporadisch auftretender Kaltstart ab Adresse 0000H ('MEM-SIZE'), ein weiterer Fehler war, dass sich der Rechner irgendwo im Programm aufhaengte und nur durch ausschalten und wiedereinschalten erneut zum Arbeiten zu bewegen war. Diese und aehnliche Effekte traten aber nur auf, wenn die 'MEM-SIZE' Eingabe ueber 32767 lagen, also im neu dazugewonnenen Speicherbereich lag.

☹☹ Der Verstand ist unser größtes Vermögen, aber Armut schändet nicht. ☹☹

☹☹ Bildung ist die Fähigkeit, sich alles anzuhören und dabei weder die Selbstbeherrschung zu verlieren noch das Selbstvertrauen. ☹☹

Also fiel der Verdacht auf die neuen Speicher IC's. Da ich diese IC's mit einigen Arbeitskollegen in einer Sammelbestellung bezogen hatte, erkundigte ich mich beide diesen, ob auch sie Probleme mit den Speichern haetten. Und siehe da, einige hatten Probleme und andere nicht. Ein Vergleich der Hardware der einzelnen Rechner brachte uns dann auf den REFRESH. Und zwar wurden in einigen Rechnern die Speicher von der CPU aus REFRESH'd und in einigen anderen Rechner wurde dies von speziellen REFRESH-Controllern erledigt. Wir fanden also heraus, das alle Rechner, die den CPU-REFRESH benutzten Probleme mit den Speicher IC's hatten. Nach dem Durchsehen von diversen Datenblaettern fanden wir schliesslich heraus wo der Wurm zu suchen war. Die Z80-CPU erzeugt einen 7 Bit REFRESH. Nun gibt es allerdings einige Hersteller die einen 8 Bit REFRESH fuer ihre 64K dynamischen RAM's brauchen. Wenn man nun diese 8 Bit REFRESH IC's direkt an den Z80 anschliesst dann werden die Adressen 8000H -> FFFFH nicht refrescht, dass bedeutet das die Informationen, die auf diese Speicherplaetze geschrieben werden nach einer gewissen Zeit verschwinden. Also Leute, aufpassen beim Kauf von 4164 IC's, fuer den TRS-80, und den Genie I muessen es IC's mit 7 Bit REFRESH sein.

Bernd Retzlaff



## Computerurteil

Ein Mann besitzt zwei Uhren, die beide defekt sind. Die eine geht grundsätzlich 10 Minuten vor, das Laufwerk der anderen ist kaputt. Der Mann beschließt, das Elektro-nengehirn zu befragen.

Frage an den Computer:

»Welche Uhr soll ich behalten?«

Antwort des Computers: »DIE KAPUTTE UHR!«

Rückfrage: »Weshalb?«

Antwort des Computers: »DIE ERSTE UHR GEHT STÄNDIG FALSCH. DIE KAPUTTE UHR ZEIGT ZWEIMAL TÄGLICH DIE RICHTIGE ZEIT AN.«

# Ein sicheres Plätzchen

Maschinenroutinen geschützt im  
GENIE untergebracht

Helmut Bernhardt

Die Sicherung von Maschinenprogrammen im oberen Adreßbereich durch Befehle wie HIMEM oder MEMSIZE gewährt lediglich Schutz gegen Überschreiben durch BASIC-Programme. Der Computer legt Maschinenprogramme beim Laden immer in den Speicherbereich, für den sie vorgesehen sind. Die Pointer werden dabei nicht berücksichtigt, so daß es normalerweise keinen Speicherbereich gibt, der wirklich 'sicher' ist. Mit einer kleinen Zusatzschaltung kann man jedoch ein 'sicheres Plätzchen' schaffen.

Auf der Suche nach diesem Platz im Adreßraum des TRS-80 oder GENIE I, II findet man schnell einige 'Lücken'. So beginnt der Adreßbereich der Tastatur bei 3800h und endet bei 38FFh. Da aber die Dekodierung des Freigabesignals für die Tastatur-Lesetreiber unvollständig ausgeführt ist, wird auch der Bereich von 3900h bis 3BFFh blockiert.

Das muß man nicht so hinnehmen. Durch geringfügige Eingriffe in den Rechner läßt sich die Tastatur vollständig dekodieren und gleichzeitig ein Freigabesignal für einen zusätzlichen Speicher-Baustein im Bereich 3900h bis 3BFFh gewinnen. Die dazu notwendige Hilfsschaltung zeigt Bild 1. Der zusätzliche Speicher besteht aus zwei ICs 2114 (1 K x 4 Bit); die weitere Dekodierung geschieht mit zwei TTL-ICs.

In den so gewonnenen 768 Byte RAM sind eigene Routinen ziemlich sicher, da man normalerweise nicht auf diesen Speicherbereich zugreifen kann. Programme mit eigener Treibroutine konnten allerdings versuchen, die Tastatur im Bereich 3900h bis 3BFFh auszulösen. Solche Routinen würden laufend 'Eingaben' aus dem RAM erhalten, was zu Fehlfunktionen führt.

In solchen (seltenen) Fällen hilft dann nur noch der Einbau eines Zweifach-Umschalters, der entweder das unvollständig dekodierte oder das neu gewonnene Freigabesignal an die Tastatur-Lesetreiber legt und die

Freigabe der RAMs bei Benutzung des 'alten' Freigabesignals für die Tastatur abschaltet. Programme, die diesen Schalter benötigen, lassen sich dann nicht mit dem zusätzlichen RAM verwenden. Das Betriebssystem NEWDOS-80 gehört glücklicherweise nicht zu dieser Gruppe von Software.

## Eingriffe

Die vier ICs der Hilfsschaltung kann man auf einem Streifen Lochraster-Platine unterbringen. Die Versorgungsspannung (+5V) ist jeweils an Pin 14 der TTL-ICs und an Pin 18 der RAM-Bausteine anzuschließen. Der Masseanschluß ist bei den

RAMs Pin 9, bei den TTL-Bausteinen Pin 7. Zwischen Versorgungsspannung und Masse sollte man einen Kondensator (100 nF) schalten.

Beim Einbau der Karte in GENIE-Rechner kann man folgendermaßen vorgehen:

Zuerst ist das Gehäuse des Computers zu öffnen. Nachdem man die acht tiefer versenkten Schrauben in der Bodenwanne des Gerätes herausgedreht hat, läßt sich die obere Gehäuseschale abheben. Schraubt man noch die Tastatur ab, liegt links das CPU-Board, in der Mitte das Interface-Board und rechts das Netzteil sowie der Kassettencorder.

Die durchzutrennende Leitung (siehe Bild 1) ist die Leiterbahn, die vom hintersten IC (Z35, 74LS32) auf dem CPU-Board in der rechten Spalte nach 'hinten' verläuft (neben der 5-V- und Masse-Leitung). An die beiden Seiten der Trennstelle kann man den Ein- und Ausgang des 'OR'-Gatters der

Hilfsschaltung anschließen. Das IC 'Z12' findet man an zweiter Stelle (von vorne) in der linken Spalte auf dem CPU-Board. Der Baustein 'Z15' (74LS32) liegt in der zweiten Spalte (von links gezählt) als zweites IC.

Die Numerierung der Verbindungsleitungen zum Interface-Board, an denen die meisten Signale für die Zusatzschaltung abzunehmen sind, erfolgt von vorne nach hinten (Nummern 1 bis 32).

Beim Anschließen der Erweiterung sollte man unbedingt darauf achten, daß die Datenleitungen nicht an den ROMs, sondern an den Lötstellen der Stecker für die Verbindungsleitung zum Interface-Board abgenommen werden. Die an den ROM-Chips anliegenden Datenleitungen sind wegen der Freigabeschaltung der Treiber nur in Leserichtung verwendbar. An der Steckverbindung lassen sich außer CS und MWR alle für die RAMs nötigen Signale abnehmen.

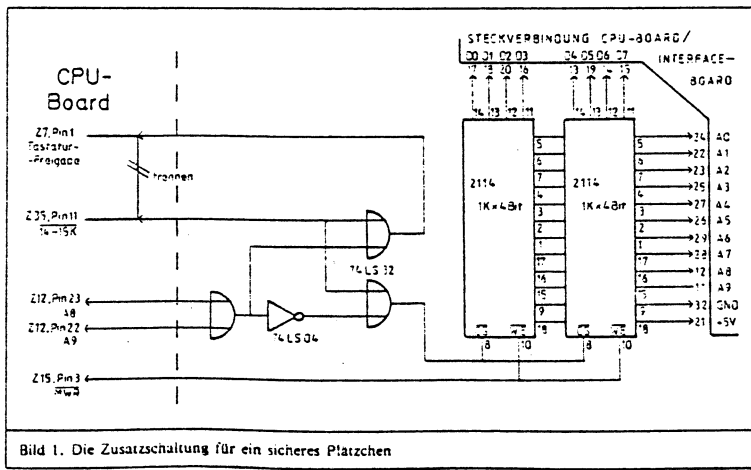


Bild 1. Die Zusatzschaltung für ein sicheres Plätzchen

Extra RAM oder Ein sicheres Plätzchen im TRS80 Modell

Artikel aus 80-U.S.Jurnal von Truman Krumholz  
sehr frei übersetzt von Hartmut Obermann

In diesem Artikel möchte ich den Aufbau eines Memory-Boards beschreiben, das den Zwischenraum zwischen ROM und Tastatur ausfüllt. Dieses zusätzliche RAM ist besonders zur geschützten Unterbringung von Maschinenroutinen geeignet, da es nicht vom System bzw. DOS benutzt wird. Das Board benutzt die Adressen 3000H-377FH und stellt 1920 Bytes freien Speicherplatz zur Verfügung.

Das Board läßt sich schnell und problemlos auf einer Lochrasterplatine in Fädertechnik aufbauen. Der Anschluß an das System erfolgt über den meist unbenutzten Screenprinteranschluß des Expansionsboards <sup>1</sup>. Dazu benötigt man entweder ein selbstgestricktes Kabel mit passendem Stecker oder man lötet die Leitungen direkt auf. Als Stromversorgung werden +5 Volt bei 0,5 Ampere benötigt <sup>2</sup>. Als Speicherchips werden zwei 2114s (1024x4 Bit Static RAM) verwendet.

Die Adressdecodierung wurde mit gängigen TTL-Bausteinen (7400 und 7410) aufgebaut. Alle Leitungen zwischen dem Board und dem Computer sind gebuffert (74LS367 und 8216) um Überlastung und Beeinflussung des Systems zu vermeiden. Die 8 "Pullup"-Widerstände (je 1 kΩ) werden benötigt, da ansonsten die High-Signale auf dem Datenbus manchmal nicht die geforderten Werte erreichen.

Das externe Powersupply <sup>2</sup> besteht aus einem 6-8 Volt Transformator (0,5-1 Ampere belastbar), einem Brückengleichrichter und einem 4700 µF Siebkondensator. Als Spannungsstabilisator dient ein IC 7805 Spannungskonstanter. Die Versorgungsspannung sollte an jedem Chip mit einem Keramik Kondensator (0,1 µF) von Störungen frei gehalten werden.

Nach dem Zusammenbau von Powersupply und Memory Board sollte als erstes eine genaue Sichtkontrolle der Schaltung erfolgen. Auf der Speicherplatine kann man außerdem mit einem Ohmmeter feststellen, ob die 5 Volt-Leitungen einen Kurzschluß haben. Erst nach diesen Kontrollen sollte man die Stromversorgung anschließen. Als letzte Überprüfung schaltet man die Versorgungsspannung ein und überprüft alle Leitungen, die später zum Computer führen darauf, daß keine mehr als 5 Volt führt.

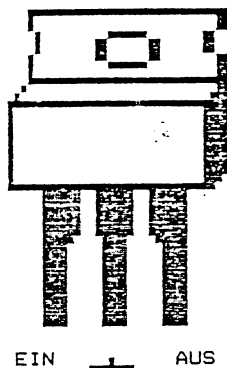
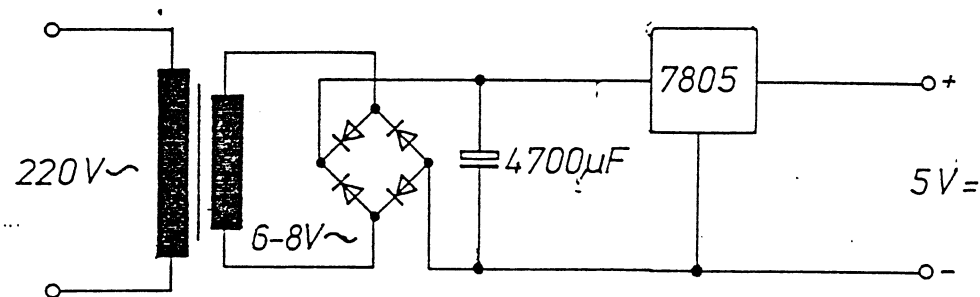
Nach dem endgültigen Anschluß der Erweiterung kann man das nun zusätzlich vorhandene RAM durch das unten stehende Programm auf seine Funktionsfähigkeit überprüfen.

<sup>1</sup> Natürlich ist auch der Anschluß über den geplanten ECB-Bus realisierbar

<sup>2</sup> Siehe Schaltbild Nr. 2

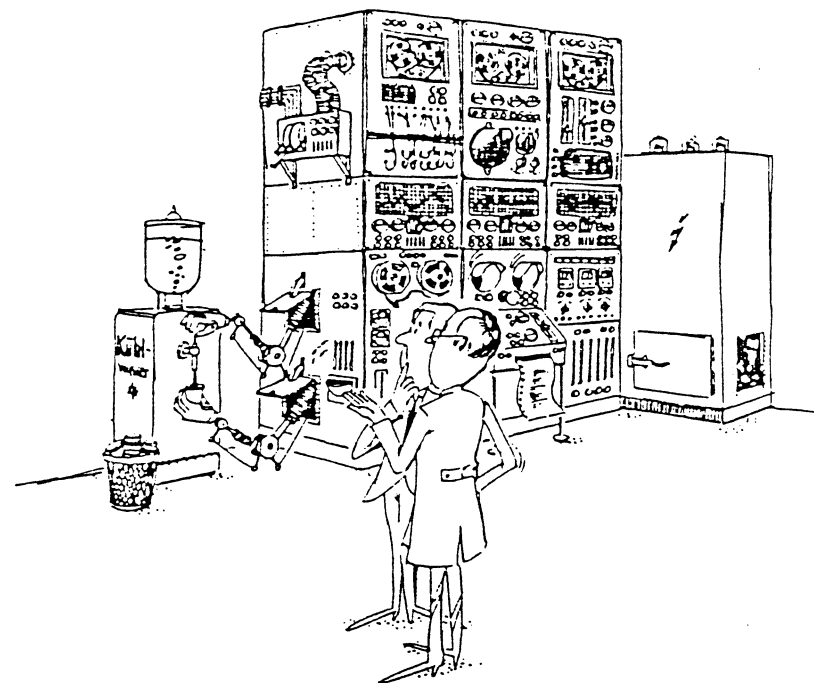
```
00 'MEMORY-TESTROUTINE FUER DAS ZUSATZ-RAM 3000H-377FH
    Langsam aber einfach! Am besten mit ZBASIC oder ACCEL
    Compilieren!!!
```

```
10 CLS
20 FOR X = 12288 TO 14207
30     FOR Y = 0 TO 255
40         POKE X,Y
50         IF PEEK (X) <> Y THEN GOSUB 100
60     NEXT Y
70 NEXT X
80 END
90 'FEHLERBEHANDLUNG
100 PRINT"SPEICHERFEHLER IN ZELLE "X;" SOLLWERT "Y;" ISTWERT "J
110 PRINT PEEK (X)
120 PRINT, "WEITER MIT <ENTER> !"
130 IF INKEY<CHR13> THEN GOTO 130 ELSE RETURN
```



# REGLER

UA 78 05  
UA 78 06  
UA 78 08  
UA 78 10  
UA 78 12  
UA 78 15  
UA 78 18  
UA 78 24

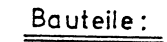


Die Originalzeichnung des Memory Board war wegen ihrer schlechten Qualität nicht zur Veröffentlichung geeignet. Ich bedanke mich vielmals bei Manfred Held, daß er eine reproduzierbare Zeichnung erstellte.  
Die Zeichnung des Netzteils stammt von mir. Wie man feststellen kann, sind meine Fähigkeiten auf dem Gebiet des technischen Zeichnens nicht allzusehr ausgeprägt. Ich hoffe, man kann erkennen was gemeint ist.

Karsten Obermann

»Donnerwetter! Das ist ja beinahe menschlich, nicht?«





zu IC 8216:

8216		
RD	CE	
0	0	Mem → Buss
1	0	Mem ← Buss
0	1	off
1	1	off

# SuperTape für TRS-80

Andreas Burgwitz, Andreas Stiller

Endlich schließt sich auch der TRS-80 der Gemeinde der SuperTape-Rechner an. Zuvor galt es allerdings eine Hürde zu nehmen, die die Entwickler des mittlerweile recht betagten Rechners errichteten: Vom SuperTape-Eingangssignal läßt die Schaltung des Kassettensystems nichts über, was mittels Software ausgewertet werden könnte. Die Lösung dieses Problems besteht aus einer kleinen Zusatzplatine, die dem TRS-80 SuperTape-Betrieb mit 7200 Baud ermöglicht.

Gerade die Anwender des TRS-80 werden ein schnelles und vor allem störstärkeres Kassettensystemverfahren begrüßen: sind sie doch vom Standard-Verfahren des Rechners Schlimmes gewohnt. Und gerade dieses Verfahren bescherte die Probleme, die den Einsatz einer Zusatzschaltung nötig machen. Aber wie fast alles, hat auch dies seine guten Seiten: Mit dem so notwendig gewordenen zusätzlichen Interface kann der TRS-80 SuperTape-Kassetten mit

7200 Baud beschreiben und lesen. Zusätzlich bietet die Schaltung je ein frei verwendbares 'Ein-/Ausgabebit'.

Die SuperTape-Software entspricht im wesentlichen dem in Heft 1/85 vorgestellten Programm für das Colour Genie, und damit auch dem SuperTape für den ZX-Spectrum. Allerdings sind die SuperTape-Ein-/Ausgaberroutinen an den Tandy angepaßt. Da der TRS-80 auch mit 7200 Baud lesen und schreiben soll, mußte die Laderoutine völlig neu erstellt werden.

Die TRS-80 SuperTape-Lösung kann man auf Geräten mit und ohne Diskettenlaufwerk betreiben. Sinnvollerweise sollte der Tandy aber mit minimal 16 KByte RAM ausgebaut sein. Das abgedruckte Programm liegt im Speicher ab der Adresse B900h. Natürlich kann man das Programm auch an die obere Grenze des Speichers legen, wo es aber oftmals mit einem Monitor oder anderen Utilities kollidiert.

Vor dem Start von SuperTape sollte man diesen Speicherbereich gegen Überschreiben durch BASIC-Programme schützen. Dies geschieht durch die Beantwortung der Frage 'MEM SIZE ?' mit 47300. Allerdings sichert man SuperTape damit nicht gegen Maschinenprogramme, die im Speicher ab B900h liegen.

## Input

Die Eingabe des Programms in den Rechner sollte nicht ohne die Hilfe eines Monitor-Programms geschehen, das das Abspeichern der eingegebenen Routinen als 'SYSTEM-File' erlaubt. Mehr Tipparbeit, aber wesentlich weniger Ärger bei Änderungen ergibt die Eingabe mit Hilfe eines Assemblers. Verfügt man über keins dieser Hilfsprogramme, sollte man SuperTape vom c't-Software-Service als SYSTEM-File beziehen.

## Bits schieben

Wie das SuperTape-Verfahren funktioniert, ist in mehreren vorausgegangenen c't-Ausgaben ausführlich erläutert worden. Die SuperTape-Programme aller Z80-Rechner orientieren sich dabei an den Routinen für den Spectrum aus c't 6/84.

Für 7200 Baud wird allerdings dabei eine einigermaßen 'flotte' CPU vorausgesetzt, die der Tandy mit seinen 1,774 MHz Taktfrequenz offensichtlich nicht aufweisen kann. Vor allen Dingen bei der Laderoutine wird es zeitlich etwas zu 'eng'.

Die optimale Prüfzeit liegt für 7200 Baud bei etwa 2/3 einer ganzen Periode für die Übertragung einer Null: also 93 Mikrosekunden. Davon ist sogar noch die mittlere Flankenenerkennungszeit abzuziehen, so daß dem Tandy also ungefähr 150 bis 160 Takte zur Verfügung stehen. Außerdem sollte man nach oben und nach unten noch etwas Spiel haben, um rein empirisch die beste Prüfzeitkonstante ermitteln zu können.

**Kommunizieren mit SuperTape**  
Das SuperTape-Verfahren ist ein 7200 Baud schnelles und ausfallsicheres Datenübertragungsverfahren zwischen Computern, das sich auf die Kompatibilität der TRS-80-Rechner ausrichtet. Es ermöglicht das Laden von Programmen von je einem SuperTape-Kassettensystem auf einen Rechner, für den das SuperTape-Verfahren gilt. SuperTape funktioniert mit handelsüblichen Kassettensystemen und benötigt keine besonderen Aufzeichnungen. Es kann die Daten von einem Rechner zum anderen übertragen werden. Für folgende Rechner ist SuperTape bisher vorgesehen: Spectrum, TRS-80, TRS-80C, TRS-80D, TRS-80E, TRS-80F, TRS-80G, TRS-80H, TRS-80I, TRS-80J, TRS-80K, TRS-80L, TRS-80M, TRS-80N, TRS-80O, TRS-80P, TRS-80Q, TRS-80R, TRS-80S, TRS-80T, TRS-80U, TRS-80V, TRS-80W, TRS-80X, TRS-80Y, TRS-80Z, TRS-80AA, TRS-80AB, TRS-80AC, TRS-80AD, TRS-80AE, TRS-80AF, TRS-80AG, TRS-80AH, TRS-80AI, TRS-80AJ, TRS-80AK, TRS-80AL, TRS-80AM, TRS-80AN, TRS-80AO, TRS-80AP, TRS-80AQ, TRS-80AR, TRS-80AS, TRS-80AT, TRS-80AU, TRS-80AV, TRS-80AW, TRS-80AX, TRS-80AY, TRS-80AZ, TRS-80BA, TRS-80BB, TRS-80BC, TRS-80BD, TRS-80BE, TRS-80BF, TRS-80BG, TRS-80BH, TRS-80BI, TRS-80BJ, TRS-80BK, TRS-80BL, TRS-80BM, TRS-80BN, TRS-80BO, TRS-80BP, TRS-80BQ, TRS-80BR, TRS-80BS, TRS-80BT, TRS-80BU, TRS-80BV, TRS-80BW, TRS-80BX, TRS-80BY, TRS-80BZ, TRS-80CA, TRS-80CB, TRS-80CC, TRS-80CD, TRS-80CE, TRS-80CF, TRS-80CG, TRS-80CH, TRS-80CI, TRS-80CJ, TRS-80CK, TRS-80CL, TRS-80CM, TRS-80CN, TRS-80CO, TRS-80CP, TRS-80CQ, TRS-80CR, TRS-80CS, TRS-80CT, TRS-80CU, TRS-80CV, TRS-80CW, TRS-80CX, TRS-80CY, TRS-80CZ, TRS-80DA, TRS-80DB, TRS-80DC, TRS-80DD, TRS-80DE, TRS-80DF, TRS-80DG, TRS-80DH, TRS-80DI, TRS-80DJ, TRS-80DK, TRS-80DL, TRS-80DM, TRS-80DN, TRS-80DO, TRS-80DP, TRS-80DQ, TRS-80DR, TRS-80DS, TRS-80DT, TRS-80DU, TRS-80DV, TRS-80DW, TRS-80DX, TRS-80DY, TRS-80DZ, TRS-80EA, TRS-80EB, TRS-80EC, TRS-80ED, TRS-80EE, TRS-80EF, TRS-80EG, TRS-80EH, TRS-80EI, TRS-80EJ, TRS-80EK, TRS-80EL, TRS-80EM, TRS-80EN, TRS-80EO, TRS-80EP, TRS-80EQ, TRS-80ER, TRS-80ES, TRS-80ET, TRS-80EU, TRS-80EV, TRS-80EW, TRS-80EX, TRS-80EY, TRS-80EZ, TRS-80FA, TRS-80FB, TRS-80FC, TRS-80FD, TRS-80FE, TRS-80FF, TRS-80FG, TRS-80FH, TRS-80FI, TRS-80FJ, TRS-80FK, TRS-80FL, TRS-80FM, TRS-80FN, TRS-80FO, TRS-80FP, TRS-80FQ, TRS-80FR, TRS-80FS, TRS-80FT, TRS-80FU, TRS-80FV, TRS-80FW, TRS-80FX, TRS-80FY, TRS-80FZ, TRS-80GA, TRS-80GB, TRS-80GC, TRS-80GD, TRS-80GE, TRS-80GF, TRS-80GG, TRS-80GH, TRS-80GI, TRS-80GJ, TRS-80GK, TRS-80GL, TRS-80GM, TRS-80GN, TRS-80GO, TRS-80GP, TRS-80GQ, TRS-80GR, TRS-80GS, TRS-80GT, TRS-80GU, TRS-80GV, TRS-80GW, TRS-80GX, TRS-80GY, TRS-80GZ, TRS-80HA, TRS-80HB, TRS-80HC, TRS-80HD, TRS-80HE, TRS-80HF, TRS-80HG, TRS-80HH, TRS-80HI, TRS-80HJ, TRS-80HK, TRS-80HL, TRS-80HM, TRS-80HN, TRS-80HO, TRS-80HP, TRS-80HQ, TRS-80HR, TRS-80HS, TRS-80HT, TRS-80HU, TRS-80HV, TRS-80HW, TRS-80HX, TRS-80HY, TRS-80HZ, TRS-80IA, TRS-80IB, TRS-80IC, TRS-80ID, TRS-80IE, TRS-80IF, TRS-80IG, TRS-80IH, TRS-80II, TRS-80IJ, TRS-80IK, TRS-80IL, TRS-80IM, TRS-80IN, TRS-80IO, TRS-80IP, TRS-80IQ, TRS-80IR, TRS-80IS, TRS-80IT, TRS-80IU, TRS-80IV, TRS-80IW, TRS-80IX, TRS-80IY, TRS-80IZ, TRS-80JA, TRS-80JB, TRS-80JC, TRS-80JD, TRS-80JE, TRS-80JF, TRS-80JG, TRS-80JH, TRS-80JI, TRS-80JJ, TRS-80JK, TRS-80JL, TRS-80JM, TRS-80JN, TRS-80JO, TRS-80JP, TRS-80JQ, TRS-80JR, TRS-80JS, TRS-80JT, TRS-80JU, TRS-80JV, TRS-80JW, TRS-80JX, TRS-80JY, TRS-80JZ, TRS-80KA, TRS-80KB, TRS-80KC, TRS-80KD, TRS-80KE, TRS-80KF, TRS-80KG, TRS-80KH, TRS-80KI, TRS-80KJ, TRS-80KL, TRS-80KM, TRS-80KN, TRS-80KO, TRS-80KP, TRS-80KQ, TRS-80KR, TRS-80KS, TRS-80KT, TRS-80KU, TRS-80KV, TRS-80KW, TRS-80KX, TRS-80KY, TRS-80KZ, TRS-80LA, TRS-80LB, TRS-80LC, TRS-80LD, TRS-80LE, TRS-80LF, TRS-80LG, TRS-80LH, TRS-80LI, TRS-80LJ, TRS-80LK, TRS-80LL, TRS-80LM, TRS-80LN, TRS-80LO, TRS-80LP, TRS-80LQ, TRS-80LR, TRS-80LS, TRS-80LT, TRS-80LU, TRS-80LV, TRS-80LW, TRS-80LX, TRS-80LY, TRS-80LZ, TRS-80MA, TRS-80MB, TRS-80MC, TRS-80MD, TRS-80ME, TRS-80MF, TRS-80MG, TRS-80MH, TRS-80MI, TRS-80MJ, TRS-80MK, TRS-80ML, TRS-80MM, TRS-80MN, TRS-80MO, TRS-80MP, TRS-80MQ, TRS-80MR, TRS-80MS, TRS-80MT, TRS-80MU, TRS-80MV, TRS-80MW, TRS-80MX, TRS-80MY, TRS-80MZ, TRS-80NA, TRS-80NB, TRS-80NC, TRS-80ND, TRS-80NE, TRS-80NF, TRS-80NG, TRS-80NH, TRS-80NI, TRS-80NJ, TRS-80NK, TRS-80NL, TRS-80NM, TRS-80NN, TRS-80NO, TRS-80NP, TRS-80NQ, TRS-80NR, TRS-80NS, TRS-80NT, TRS-80NU, TRS-80NV, TRS-80NW, TRS-80NX, TRS-80NY, TRS-80NZ, TRS-80OA, TRS-80OB, TRS-80OC, TRS-80OD, TRS-80OE, TRS-80OF, TRS-80OG, TRS-80OH, TRS-80OI, TRS-80OJ, TRS-80OK, TRS-80OL, TRS-80OM, TRS-80ON, TRS-80OO, TRS-80OP, TRS-80OQ, TRS-80OR, TRS-80OS, TRS-80OT, TRS-80OU, TRS-80OV, TRS-80OW, TRS-80OX, TRS-80OY, TRS-80OZ, TRS-80PA, TRS-80PB, TRS-80PC, TRS-80PD, TRS-80PE, TRS-80PF, TRS-80PG, TRS-80PH, TRS-80PI, TRS-80PJ, TRS-80PK, TRS-80PL, TRS-80PM, TRS-80PN, TRS-80PO, TRS-80PP, TRS-80PQ, TRS-80PR, TRS-80PS, TRS-80PT, TRS-80PU, TRS-80PV, TRS-80PW, TRS-80PX, TRS-80PY, TRS-80PZ, TRS-80QA, TRS-80QB, TRS-80QC, TRS-80QD, TRS-80QE, TRS-80QF, TRS-80QG, TRS-80QH, TRS-80QI, TRS-80QJ, TRS-80QK, TRS-80QL, TRS-80QM, TRS-80QN, TRS-80QO, TRS-80QP, TRS-80QQ, TRS-80QR, TRS-80QS, TRS-80QT, TRS-80QU, TRS-80QV, TRS-80QW, TRS-80QX, TRS-80QY, TRS-80QZ, TRS-80RA, TRS-80RB, TRS-80RC, TRS-80RD, TRS-80RE, TRS-80RF, TRS-80RG, TRS-80RH, TRS-80RI, TRS-80RJ, TRS-80RK, TRS-80RL, TRS-80RM, TRS-80RN, TRS-80RO, TRS-80RP, TRS-80RQ, TRS-80RR, TRS-80RS, TRS-80RT, TRS-80RU, TRS-80RV, TRS-80RW, TRS-80RX, TRS-80RY, TRS-80RZ, TRS-80SA, TRS-80SB, TRS-80SC, TRS-80SD, TRS-80SE, TRS-80SF, TRS-80SG, TRS-80SH, TRS-80SI, TRS-80SJ, TRS-80SK, TRS-80SL, TRS-80SM, TRS-80SN, TRS-80SO, TRS-80SP, TRS-80SQ, TRS-80SR, TRS-80SS, TRS-80ST, TRS-80SU, TRS-80SV, TRS-80SW, TRS-80SX, TRS-80SY, TRS-80SZ, TRS-80TA, TRS-80TB, TRS-80TC, TRS-80TD, TRS-80TE, TRS-80TF, TRS-80TG, TRS-80TH, TRS-80TI, TRS-80TJ, TRS-80TK, TRS-80TL, TRS-80TM, TRS-80TN, TRS-80TO, TRS-80TP, TRS-80TQ, TRS-80TR, TRS-80TS, TRS-80TT, TRS-80TU, TRS-80TV, TRS-80TW, TRS-80TX, TRS-80TY, TRS-80TZ, TRS-80UA, TRS-80UB, TRS-80UC, TRS-80UD, TRS-80UE, TRS-80UF, TRS-80UG, TRS-80UH, TRS-80UI, TRS-80UJ, TRS-80UK, TRS-80UL, TRS-80UM, TRS-80UN, TRS-80UO, TRS-80UP, TRS-80UQ, TRS-80UR, TRS-80US, TRS-80UT, TRS-80UU, TRS-80UV, TRS-80UW, TRS-80UX, TRS-80UY, TRS-80UZ, TRS-80VA, TRS-80VB, TRS-80VC, TRS-80VD, TRS-80VE, TRS-80VF, TRS-80VG, TRS-80VH, TRS-80VI, TRS-80VJ, TRS-80VK, TRS-80VL, TRS-80VM, TRS-80VN, TRS-80VO, TRS-80VP, TRS-80VQ, TRS-80VR, TRS-80VS, TRS-80VT, TRS-80VU, TRS-80VV, TRS-80VW, TRS-80VX, TRS-80VY, TRS-80VZ, TRS-80WA, TRS-80WB, TRS-80WC, TRS-80WD, TRS-80WE, TRS-80WF, TRS-80WG, TRS-80WH, TRS-80WI, TRS-80WJ, TRS-80WK, TRS-80WL, TRS-80WM, TRS-80WN, TRS-80WO, TRS-80WP, TRS-80WQ, TRS-80WR, TRS-80WS, TRS-80WT, TRS-80WU, TRS-80WV, TRS-80WW, TRS-80WX, TRS-80WY, TRS-80WZ, TRS-80XA, TRS-80XB, TRS-80XC, TRS-80XD, TRS-80XE, TRS-80XF, TRS-80XG, TRS-80XH, TRS-80XI, TRS-80XJ, TRS-80XK, TRS-80XL, TRS-80XM, TRS-80XN, TRS-80XO, TRS-80XP, TRS-80XQ, TRS-80XR, TRS-80XS, TRS-80XT, TRS-80XU, TRS-80XV, TRS-80XW, TRS-80XX, TRS-80XY, TRS-80XZ, TRS-80YA, TRS-80YB, TRS-80YC, TRS-80YD, TRS-80YE, TRS-80YF, TRS-80YG, TRS-80YH, TRS-80YI, TRS-80YJ, TRS-80YK, TRS-80YL, TRS-80YM, TRS-80YN, TRS-80YO, TRS-80YP, TRS-80YQ, TRS-80YR, TRS-80YS, TRS-80YT, TRS-80YU, TRS-80YV, TRS-80YW, TRS-80YX, TRS-80YY, TRS-80YZ, TRS-80ZA, TRS-80ZB, TRS-80ZC, TRS-80ZD, TRS-80ZE, TRS-80ZF, TRS-80ZG, TRS-80ZH, TRS-80ZI, TRS-80ZJ, TRS-80ZK, TRS-80ZL, TRS-80ZM, TRS-80ZN, TRS-80ZO, TRS-80ZP, TRS-80ZQ, TRS-80ZR, TRS-80ZS, TRS-80ZT, TRS-80ZU, TRS-80ZV, TRS-80ZW, TRS-80ZX, TRS-80ZY, TRS-80ZZ.

Die Spectrum-Routine benötigt jedoch bereits 175 Takte, selbst wenn gar kein Sprung mehr in eine Warteroutine erfolgt. Daß so ein Sprung nicht ausgeführt werden soll, müßte aber das Programm auch noch feststellen, da ja für 3600 Baud auf jeden Fall eine zusätzliche Warteschleife nötig ist. Für diese Abfrage wäre erst recht keine Zeit mehr vorhanden, so daß die gesamte Laderoutine einschließlich Synchronisation zweimal geschrieben werden müßte, einmal für 7200 Baud und einmal für 3600 Baud.

Das Spectrum-Programm birgt auch noch eine Gefahr für andere Rechner: Beim Einlesen des Ports erstreckt sich die Abfrage über den gesamten Datenbus. Eine eventuell 'floatede' Datenleitung verfälscht das

Ergebnis völlig. Eine zusätzliche Ausmaskierung des einzulesenden Bits ist also vonnöten.

## Über alle Register

Um 7200 Baud auch auf dem Tandy — sogar mit einiger Reserve — lauffähig zu bekommen, muß man fast alle Register der Z80-CPU 'ziehen'. Lediglich das Interrupt- und Refresh-, das IY- und das DE-Register bleiben unberücksichtigt.

Das neue Interface legt die Signale vom Kassettenspeicher auf die Datenleitung D7. Das hat den Vorteil, daß hierbei bereits mit dem IN-Befehl das Bit abgefragt werden kann. Allerdings muß dafür das Register C die Portadresse enthalten, also IN A,(C). Die Abfrage JP M beziehungsweise JP P wertet dann unmittelbar das eingeseleste Bit auf D7 aus. Damit läßt sich die Flankenabfrage innerhalb von nur 22 Takten durchführen, was der Lesesicherheit sehr zugute kommt.

Der eingeseleste Wert wird als Carry-Flag in AF' zwischengespeichert. Falls der neu eingeseleste Wert mit dem alten übereinstimmt, erkennt das Programm eine Null; bei Ungleichheit eine Eins. Bei einer Eins wird das Prüfsummenregister HL inkrementiert. Das entsprechend gesetzte Carry-Flag gelangt dann in bewährter Manier per Rotation ins E-Register, das so Bit für Bit bis zu dem gewünschten Byte aufgefüllt wird. Das Programm erkennt die 'Fülle' an dem Endflag, das am Anfang (Adresse BCC9h) gesetzt wird und dann nach acht Rotationen ins Carry gelangt.

Beim bitweisen Einlesen innerhalb der ersten Synchronisationsstufe findet — bedingt durch den Befehl SET E,0 — nur eine Rotation statt.

Der indirekte Sprung über das IX-Register ermöglicht eine einfache und schnelle Verzweigung (8 Takte) in eine Warteschleife, die der gewählten Baudrate entspricht. Bei 7200 Baud läßt sich so die Wartezeit im 'Viertakterhythmus' durch Einsprung in die NOP-Reihe von LDW36 bis LDSAM verändern. Der Benutzer von SuperTape sollte sowohl diese Einsprungsadresse für 7200 Baud (in Adresse BC25h) als auch die Wartekonstante

LCON36 für 3600 Baud etwas variieren, um die für seinen Recorder besten Werte zu finden.

Immerhin 22 Takte ließen sich gegenüber den Spectrum-Routinen dadurch sparen, daß für die Blocklade- und Vergleichsoperationen die gestrichenen Register herangezogen wurden. Die zeitkritische Situation am Ende der Synchronisation konnte entschärft werden, da die Routine gleich in die LDBYT-Routine überläuft und somit bereits mit einem geladenen Byte zurückkehrt. Dieses eine Byte ist allerdings extra zu behandeln, bevor die eigentliche Lade- beziehungsweise Verify-Schleife beginnt.

Summa summarum kommt die Laderoutine einschließlich des JP (IX)-Befehls mit 139 Takten 'Eigenzeit' aus.

## Gut ge'toggle't

Beim SAVE liegen die Dinge wesentlich einfacher. Da das Interface auch noch die Arbeit des 'toggleins' übernimmt, also bei jedem Aufruf den Portzustand ändert, wird das sonst nötige Toggle-Register frei. Dieses findet im Programm Verwendung als Flagregister, das das Baudratenflag schnell abrufbar speichert. Bei 3600 Baud (Bit 7 von D = 0) wird dann eine zusätzliche Wartezeit von 124 Takten eingeschoben.

## Der Rahmen

Den Rahmen um die eigentlichen SuperTape-Routinen bildet ein menügesteuerter Programmteil, der die Anwendung von SuperTape komfortabel macht. Nach dem Laden von SuperTape meldet sich der TRS-80 mit:

SUPERTAPE FUER TRS-80. MODELL 1  
(SAVE) (OUIJK) (LOAD) (VERIFY) (EXIT)  
COMMAND:

Durch die Eingabe eines in Klammern stehenden Buchstabens verzweigt das Programm zu der entsprechenden Routine und fordert die eventuell noch benötigten Angaben an. Gibt man zum Beispiel 'S' für Speichern mit 3600 Baud (oder Q — Speichern mit 7200 Baud) ein, fragt das Programm nach dem Namen des abzuspeichernden Programms:

NAME: . . . TYP

Allerdings sollte man bei 7200 Baud einen besseren Recorder

verwenden als das Tandy-Gerät.

Nach der Eingabe des Namens gelangt man durch Berätigen der 'RETURN'-Taste zur TYP-Abfrage, zum Beispiel 'BAS' für BASIC-Programme oder 'HEX' für Speicherinhalte. Beim Typ 'HEX' fragt das Programm nach der Start- und Endadresse des Speicherbereichs. Diese Frage entfällt beim Typ 'BAS': das Programm schreibt ein BASIC-Programm vom Anfang bis zum Ende (im Rechner durch Zeiger markiert) auf Band.

Bei der folgenden Frage 'TAPE READY ?' hat man noch die Möglichkeit, sofort wieder zum Menü zurückzukehren, indem man diese Frage mit 'N' beantwortet. Andernfalls schreibt das Programm die Daten auf Band.

Alle Funktionen, die man schon einmal ausgeführt hat, kann man im Menü durch die Eingabe von 'Shift Buchstabe' erneut aufrufen, wobei die zuvor eingegebenen Parameter übernommen und angezeigt werden.

Soll die Aufzeichnung sogleich überprüft werden, genügt die Eingabe von 'V' (Shift), da die Parameter ja dieselben wie beim Abspeichern sind. Nach der Frage 'TAPE READY ?' erscheint in der untersten Bildschirmzeile ein Sternchen, das zu 'flackern' beginnt, sobald Signale vom Band kommen. Während das Sternchen 'blinkt', kann man den Ladevorgang jederzeit durch die 'BREAK'-Taste abbrechen und zurück zum Menü gelangen. Als Abschluß der Verify-Routine listet das Programm die Daten des 'geladenen' Programms und gibt eine Lade-/Prüfmeldung neben dem Sternchen aus:

\* Fehlerfrei  
B Blockfehler  
> Fehler bei Verify  
F Prüfsummenfehler

Beim Load-Befehl 'L' fragt das Programm nach der Eingabe des Namens noch zusätzlich nach einer neuen Startadresse. Somit kann man ein Programm an beliebiger Stelle im Speicher ablegen. Bei der Namensangabe erlaubt das Programm die Verwendung von 'Jokern': das Zeichen '\*' steht für einen beliebigen langen Namen mit beliebigen Buchstaben. Die einzige Ausnahme hiervon bilden BA-

SIC-Programme: Sie sollten immer mit der Typ-Angabe 'BAS' geladen werden, da nur dann die Zeiger für den späteren Start des BASIC-Programms richtig gesetzt werden. Die Zeiger 'stehen' zwar richtig, wenn man ein mit dem DISK-BASIC erstelltes Programm wieder in diesem BASIC starten beziehungsweise listen will; sie stimmen aber nicht, wenn ein ROM-BASIC-Programm unter DISK-BASIC laufen soll (oder umgekehrt).

Nach dem erfolgreichen Laden eines BASIC-Programms verzweigt das SuperTape-Programm in das BASIC: Hat man SuperTape aus dem ROM-BASIC gestartet, springt das Programm auch in dieses BASIC. Arbeitet man mit Disketten und hat SuperTape nicht aus dem DISK-BASIC auferufen, erscheint kurz die Fehlermeldung 'KEIN BASIC' und anschließend das Menü. Soll ein BASIC-Programm nach dem Laden ausgeführt werden, ist folgendermaßen vorzugehen:

Vom DOS aus mit dem Befehl 'LOAD SUTRS/CMD' SuperTape laden.

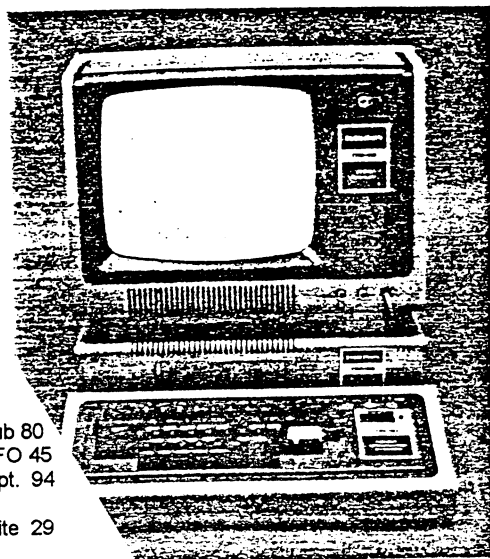
Anschließend muß man BASIC aufrufen.

Im BASIC den Befehl 'SYSTEM' eingeben und auf die Meldung '?\*' mit '/47360' antworten — SuperTape wird gestartet.

Einen direkten Rücksprung ins BASIC oder DOS kann man durch die Eingabe von 'E' im SuperTape-Programm erreichen. Arbeitet man mit Disketten und hatte vor dem Aufruf von SuperTape BASIC geladen, führt 'E' direkt ins DISK-BASIC. Ohne geladenes BASIC gelangt man durch die Eingabe von 'E' in die Befehlsreihe vom DOS, beim Arbeiten im ROM-BASIC erfolgt ein BASIC-Warmstart.

Die Unterscheidung zwischen diesen drei 'Betriebsystemen' erfolgt selbstständig durch das Programm. Dazu wertet es ein Byte aus dem 'Keyboard DCB' (Zeiger auf die Tastatur-Treiberoutine) aus.

Beim TRS-80, Modell 1 mit dem 'Level 2 ROM-BASIC' steht in dieser Speicherzeile (4017h) der Wert 03 beim aktivierten ROM-BASIC, beim Betrieb des TRS-80 unter NEW-DOS 40 erwartet das Pro-



## Interface

gramm das Byte 43h in der Speicherzelle. Ein Sprung in das DISK-BASIC erfolgt dann, wenn die zuvor genannten Werte nicht erkannt werden konnten.

Die Adreßdekodierung des Interfaces besteht aus IC1 und IC2. Die ausgewerteten Adressen schalten entweder ein Flipflop (IC3a) oder erlauben Ein-/Ausgaben über den Puffer IC4. Das Flipflop IC3a schaltet bei jedem Zugriff auf den Port 7Fh (127d) um, unabhängig von den ausgegebenen Daten. Über die drei Inverter (IC3b, IC3c) und das Filter (R2, R3, C1, C2) gelangt das Ausgangssignal des Flipflops an den Kassettencorder.

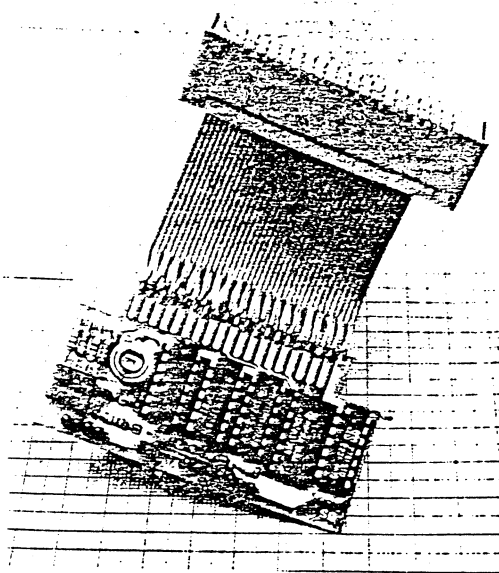
Die Datenleitungen D0, D1, D6 und D7 des TRS-80 liegen über den Puffer IC4 an IC3a und IC3b. Der logische Pegel der Datenleitung D0 gelangt über die Brücke J1 an den 'Clear'-Eingang von IC3a. Legt man diesen Eingang kurzfristig auf logisch 0, schaltet der Ausgang von IC3a auf logisch 1. Diese Funktion benötigt man aber nur dann, wenn mehr als zwei Rechner mit einer 'SuperTape-Ringleitung' verbunden werden sollen. Im Normalfall kann man die Brücke I offenlassen und an der Leiste X3 den Impuls von D0 abgreifen.

Mit dem Datenbit D1 kann man das Flipflop IC3b steuern. So setzt zum Beispiel der BASIC-Befehl 'OUT 126,2' den Ausgang von IC3b auf logisch 1. Schließt man an diesen Ausgang eine Treiberstufe und ein Relais an, kann man den Kassettencorder programmgesteuert ein- und ausschalten.

Das Signal vom Kassettencorder gelangt über die Triggerschaltung (IC5d, IC5e) an die Datenleitung D7. Der Pegel dieser Leitung entspricht nach einer Portabfrage dem Eingangssignal am Kassetteneingang. Die Leitung D6 liegt über den Puffer an der Anschlußleiste X3, Pin 2. Hier kann man ein Eingangssignal mit TTL-Pegel in den Rechner einlesen.

## Aufbau

Da die Leiterplatte sehr eng bestückt wird, sollte man beim Aufbau besonders sorgfältig vorgehen. Der Anschluß des Interfaces an den Rechner ze-



Das SuperTape-Interface für den TRS-80.

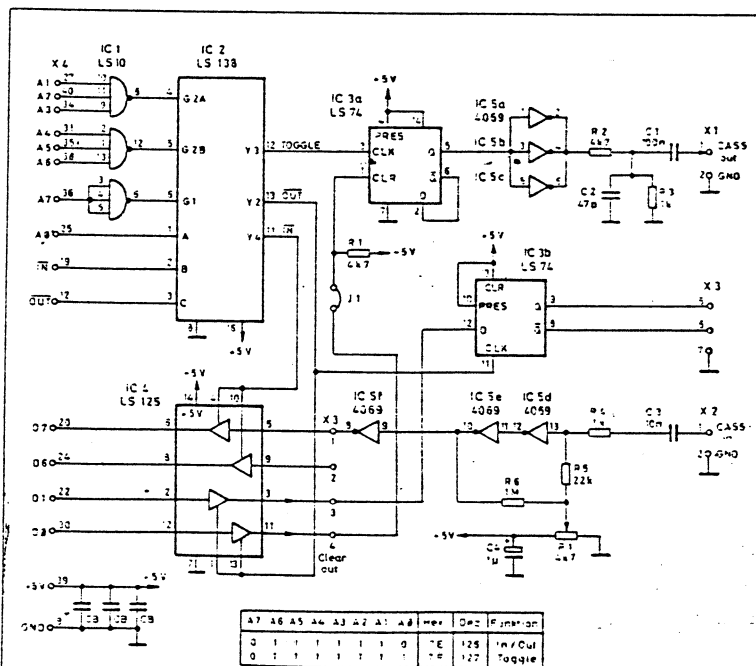


Bild 1. Schaltplan des TRS-80-Interfaces.

schiebt über die Steckerleiste X4. Die Platine ist so ausgelegt, daß man eine 40polige Steckerleiste direkt anlöten kann. Soll das Interface an den 'Interface-Bus' (J3) des Tandy-Expansion-Interface angeschlossen werden, hat diese Lösung allerdings den Nachteil, daß die Platine 'in der Luft schwebt'. Beim direkten Anschluß an den TRS-80-Bus kann man die Leiterplatte durch zwei Distanzrollen feststen Halt geben.

Die wohl bessere Lösung besteht darin, die Platine über ein Stück Flachbandkabel und einen 'Preßstecker' mit dem Rechner zu verbinden. Allerdings muß das Kabel so kurz wie möglich sein, da sonst der Tandy recht 'eigenwillig' wird.

## Test

Nach dem Bestücken sollte man die Platine sehr sorgfältig auf Kurzschlüsse und Unterbrechungen untersuchen. Anschließend kann man die Karte auf den Bus des Rechners oder des Expansion-Boards stecken

und den Computer einschalten: Das System sollte sich wie gewohnt verhalten. Andernfalls ist der Computer sofort abzuschalten und das Interface erneut zu überprüfen.

Tritt dieser Fall nicht ein, können die weiteren Tests des Interfaces mit BASIC-Befehlen und einem Voltmeter durchgeführt werden. Dazu sollte man die Brücke J1 schließen und ein Voltmeter an Pin 2, 4 oder 6 von IC5 anschließen. Nach jeder Ausführung des BASIC-Befehls 'OUT 127,0' muß der Pegel an IC5 wechseln; von circa 0 Volt auf etwa 5 Volt und umgekehrt. Für den nächsten Test sollte die Spannung an IC5 etwa 0 Volt betragen. Mit der Ausführung der Anweisung 'OUT 126,0' muß an IC5 der Pegel auf rund 5 Volt wechseln (Reset-Funktion).

Anschließend ist das Voltmeter an Pin 5 von der Steckerleiste X3 anzuschließen. Der Befehl 'OUT 126,2' bewirkt, daß der Ausgang von IC3b auf logisch 1 (etwa 3 Volt) springt. Mit der Anweisung 'OUT 126,0' ergibt sich eine Spannung von rund 0 Volt an dem Ausgang dieses ICs.

Für die Überprüfung des Kassetteneingangs gibt man folgendes Programm ein:

```
10 A=INP(126)
20 PRINT A;
30 GOTO 10
```

Nach dem Start des Programms wird ein Wert angezeigt, der sich ändern muß, wenn man P1 an den rechten oder linken Anschlag dreht. Die richtige Einstellung von P1 liegt an dem Punkt, wo die Anzeige zwischen den beiden Werten springt.

## Stückliste

Widerstände		IC2	74LS138
R1,2	4k7	IC3	74LS74
R3,4	1k	IC4	74LS125
R5	22k	IC5	CD 4069
R6	1M		
P1	4k7 Trimmer, liegend	Sonstiges	
		X1, X2, J1	Pfostenleiste, 2polig
Kondensatoren		X3	Pfostenleiste, 7polig
C1	100n	X4	40poliger 'Card-Edge'-Stecker, 2reihig, Raster 2,54 mm
C2	47p		
C3	10n		
C4	1µ, Tantal		
CB	3 Stützkondensatoren, 100n, ker.		
Halbleiter		Brückenstecker für Pfostenleiste; IC-Fassungen: 4 x 14polig; 1 x 16polig; Platine 'TRS-80 SuperTape'.	
IC1	74LS10		

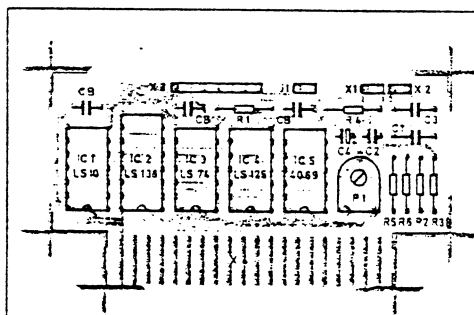


Bild 2. Der Bestückungsplan. Da die ICs sehr dicht beieinander liegen, müssen 'anreihbare' Fassungen verwendet werden.

8940	ORG	88780H
8943	DISPL	88780H
8944	POTAS	88780H
8945	READY	88780H
8946	CLS	88780H
8947	CURSYM	88780H
8948	READY	88780H
8949	READY	88780H
8950	READY	88780H
8951	READY	88780H
8952	READY	88780H
8953	READY	88780H
8954	READY	88780H
8955	READY	88780H
8956	READY	88780H
8957	READY	88780H
8958	READY	88780H
8959	READY	88780H
8960	READY	88780H
8961	READY	88780H
8962	READY	88780H
8963	READY	88780H
8964	READY	88780H
8965	READY	88780H
8966	READY	88780H
8967	READY	88780H
8968	READY	88780H
8969	READY	88780H
8970	READY	88780H
8971	READY	88780H
8972	READY	88780H
8973	READY	88780H
8974	READY	88780H
8975	READY	88780H
8976	READY	88780H
8977	READY	88780H
8978	READY	88780H
8979	READY	88780H
8980	READY	88780H
8981	READY	88780H
8982	READY	88780H
8983	READY	88780H
8984	READY	88780H
8985	READY	88780H
8986	READY	88780H
8987	READY	88780H
8988	READY	88780H
8989	READY	88780H
8990	READY	88780H
8991	READY	88780H
8992	READY	88780H
8993	READY	88780H
8994	READY	88780H
8995	READY	88780H
8996	READY	88780H
8997	READY	88780H
8998	READY	88780H
8999	READY	88780H

.....  
: SUPER TAPES TANDY TRS-80, MODELL 1  
: MENUE AUSWAHL  
: .....

8999	START:	LD	SP, STACK	
9000	START:	CALL	CLRSE	: CLEAR BILDSCHIRM
9001	START:	LD	HL, TEXT1	: TEXT
9002	START:	LD	HL, TEXT2	: TEXT
9003	TAST:	CALL	POTAS	: BILDSCHIRM
9004	LD	LD	C, A	
9005	LD	LD	SP, A	
9006	LD	LD	SP, A	
9007	LD	LD	SP, A	
9008	LD	LD	SP, A	
9009	LD	LD	SP, A	
9010	LD	LD	SP, A	
9011	LD	LD	SP, A	
9012	LD	LD	SP, A	
9013	LD	LD	SP, A	
9014	LD	LD	SP, A	
9015	LD	LD	SP, A	
9016	LD	LD	SP, A	
9017	LD	LD	SP, A	
9018	LD	LD	SP, A	
9019	LD	LD	SP, A	
9020	LD	LD	SP, A	
9021	LD	LD	SP, A	
9022	LD	LD	SP, A	
9023	LD	LD	SP, A	
9024	LD	LD	SP, A	
9025	LD	LD	SP, A	
9026	LD	LD	SP, A	
9027	LD	LD	SP, A	

.....  
: BILDSCHIRM : ZEICHEN UND CURSOR SETZEN  
: .....

9028	CLRSE:	CALL	CLS	: CLEAR SCREEN
9029	LD	LD	A, A20H	: CURSYM, A
9030	LD	LD	A, A20H	: CURSYM, A
9031	LD	LD	A, A20H	: CURSYM, A
9032	LD	LD	A, A20H	: CURSYM, A
9033	LD	LD	A, A20H	: CURSYM, A
9034	LD	LD	A, A20H	: CURSYM, A
9035	LD	LD	A, A20H	: CURSYM, A
9036	LD	LD	A, A20H	: CURSYM, A
9037	LD	LD	A, A20H	: CURSYM, A
9038	LD	LD	A, A20H	: CURSYM, A
9039	LD	LD	A, A20H	: CURSYM, A
9040	LD	LD	A, A20H	: CURSYM, A
9041	LD	LD	A, A20H	: CURSYM, A
9042	LD	LD	A, A20H	: CURSYM, A
9043	LD	LD	A, A20H	: CURSYM, A
9044	LD	LD	A, A20H	: CURSYM, A
9045	LD	LD	A, A20H	: CURSYM, A
9046	LD	LD	A, A20H	: CURSYM, A
9047	LD	LD	A, A20H	: CURSYM, A
9048	LD	LD	A, A20H	: CURSYM, A
9049	LD	LD	A, A20H	: CURSYM, A
9050	LD	LD	A, A20H	: CURSYM, A
9051	LD	LD	A, A20H	: CURSYM, A
9052	LD	LD	A, A20H	: CURSYM, A
9053	LD	LD	A, A20H	: CURSYM, A
9054	LD	LD	A, A20H	: CURSYM, A
9055	LD	LD	A, A20H	: CURSYM, A
9056	LD	LD	A, A20H	: CURSYM, A
9057	LD	LD	A, A20H	: CURSYM, A
9058	LD	LD	A, A20H	: CURSYM, A
9059	LD	LD	A, A20H	: CURSYM, A
9060	LD	LD	A, A20H	: CURSYM, A
9061	LD	LD	A, A20H	: CURSYM, A
9062	LD	LD	A, A20H	: CURSYM, A
9063	LD	LD	A, A20H	: CURSYM, A
9064	LD	LD	A, A20H	: CURSYM, A
9065	LD	LD	A, A20H	: CURSYM, A
9066	LD	LD	A, A20H	: CURSYM, A
9067	LD	LD	A, A20H	: CURSYM, A
9068	LD	LD	A, A20H	: CURSYM, A
9069	LD	LD	A, A20H	: CURSYM, A
9070	LD	LD	A, A20H	: CURSYM, A
9071	LD	LD	A, A20H	: CURSYM, A
9072	LD	LD	A, A20H	: CURSYM, A
9073	LD	LD	A, A20H	: CURSYM, A
9074	LD	LD	A, A20H	: CURSYM, A
9075	LD	LD	A, A20H	: CURSYM, A
9076	LD	LD	A, A20H	: CURSYM, A
9077	LD	LD	A, A20H	: CURSYM, A
9078	LD	LD	A, A20H	: CURSYM, A
9079	LD	LD	A, A20H	: CURSYM, A
9080	LD	LD	A, A20H	: CURSYM, A
9081	LD	LD	A, A20H	: CURSYM, A
9082	LD	LD	A, A20H	: CURSYM, A
9083	LD	LD	A, A20H	: CURSYM, A
9084	LD	LD	A, A20H	: CURSYM, A
9085	LD	LD	A, A20H	: CURSYM, A
9086	LD	LD	A, A20H	: CURSYM, A
9087	LD	LD	A, A20H	: CURSYM, A
9088	LD	LD	A, A20H	: CURSYM, A
9089	LD	LD	A, A20H	: CURSYM, A
9090	LD	LD	A, A20H	: CURSYM, A
9091	LD	LD	A, A20H	: CURSYM, A
9092	LD	LD	A, A20H	: CURSYM, A
9093	LD	LD	A, A20H	: CURSYM, A
9094	LD	LD	A, A20H	: CURSYM, A
9095	LD	LD	A, A20H	: CURSYM, A
9096	LD	LD	A, A20H	: CURSYM, A
9097	LD	LD	A, A20H	: CURSYM, A
9098	LD	LD	A, A20H	: CURSYM, A
9099	LD	LD	A, A20H	: CURSYM, A

.....  
: RETURN WURDE GEDRUECKT  
: .....

```

.....
: CURSOR ZURUECKSETZEN
.....
9958 28 CURS: DEC HL :
9959 28 DEC HL :CURSOR
9960 44 INC B :EIN ZEICHEN
9961 44 INC B :ZURUECKSETZEN
9962 2E19 LD A, #19H
9963 1BE7 JR PCONT

```



```

BC3F 47      LD      S,A          ;PRUEFSUMME NACH BC
BC48 D9      EXX      PUSH     HL          ;
BC41 E3      EXX      POP      HL          ;BESTIMMTE
BC42 D9      EXX      POP      HL          ;PRUEFSUMME NACH HL
BC43 E1      .....
;LDTEST: PRUEFSUMMENTEST
;.....
BC44 AF      XOR      SRC          ;VERGLEICHE HL MIT BC
BC45 E042    LD      R0,A,#23H          ;LADE AKKU MIT '8'
BC47 3E23    LD      RET          ;RET FALLS OK
BC48 C8      LD      A,#44H          ;SONST 'F'
BC49 3E44    LD      RET          ;UND RETURN
BC4C C9      .....
;LDLOWER: VERIFY
;.....
BC4D 147E    LD      L,PORTIN          ;PORTADRESSE
BC4F D9      EXX      CALL     LD5YN          ;SYNCHRONISIERT
BC78 C0B8BC  EXX      JX          ;VERGLEICH
BC79 E041    CP      JP          ;ERROR FALLS ( )
BC7A 2813    JP      NZ,ERRVE          ;SPRUNG FALLS ENDE
BC78 E2C838  EXX      PD,UTEST          ;
BC78 D9      VERRNT: EXX      LD      LD8YT          ;
BC79 C0C9BC  XOR      CALL     LD8YT          ;NACHST. BYTE
BC7F D9      EXX      CP          ;
BC88 E041    CP          ;VERGLEICH
BC82 E478BC  JP          PE,VERDUT          ;BEI PE WEITER
BC83 40      LD      C,D          ;SONST LADE PRUEF-
BC84 C0C9BC  ;TEST: LD      LD8YT          ;SUMME
BC89 18CC    JR      TEST          ;
BC88 3E3E    ERRTVE: LD      A,#3EH          ;FEHLER BEI VERIFY
BC8D C9      RET
;.....
;LD5YN: SYNCHRONISATION
;.....
BC8E 1E88    LD5YN: LD      E,888H          ;INIT LADEBYTE
BC8F 8E7E    LD      D,C,PORTIN          ;PORTADRESSE
BC92 F3      DI          ;INTERRUPTS AUS
BC93 57      LD      D,A          ;ICHSCHLUSSMARKE
;.....
BC94 88      ;WARTET AUF EIN SYCNH
BC95 A6      SYNINH: DEFB      88H          ;
BC96 54      LDINX: DEFB      88AH          ;WARTET 4 TAKTE
BC97 C399BC  SYNEU2: JP      SYMEI          ;WARTET 7 TAKTE
BC98 C8C3    SYMEI: SET      8,E          ;WARTET 18 TAKTE
BC99 C0C8BC  LD      LD8YT          ;LADE NUR EIN BIT
BC9E FE1a    CP          ;SYCNH ERKANNT?
BC98 5F      LD      E,A          ;
BCA1 2F22    JR      NZ,LDINX          ;
;TEST AUF WEITERE SYCNH
BCA3 47      DEFB      87H          ;WARTET 9 TAKTE
BCA4 D8      DEFB      8D8H          ;
BCA5 88A8    LD      D,88AH          ;ZAEHLER AUF 18
BCA6 C0C8BC  SYCNH: DEFB      87H          ;WARTET 7 TAKTE
BCA8 C0C9BC  CALL     LD8YT          ;LADE NACHSTES BYTE
BCAC FE1:4    CP          ;VERGL. MIT SYCNH
BCAD 28E5    JP      NZ,SYMEU1          ;UNGLEICH ? SYCNH NEU
BCAE 88F8    LD      8,SYCNH          ;WARTET 7TAKTE
BCB0 1F55    DMC2: SYCNH          ;WEITER BIS 18:SYCNH
;.....
BCB2 1884    JR      SYEIN          ;
BCB4 A6      SYION: DEFB      8AH          ;WARTET 22 TAKTE
BCB5 E041    DEFB      C8FH          ;
BCB6 D8      DEFB      8D8H          ;
BCB7 D8      DEFB      8D8H          ;
BCB8 C0C9BC  SYEIN: CALL     LD8YT          ;LADE NACHSTES BYTE
BCB9 C0C8BC  CP          ;VERGL. MIT SYCNH
BCBD 23F5    JR      8D8H          ;JA, DANN WEITER LADEN
;.....
BCBF BA      CP      D          ;ABSCHLUSS RICHTIG?
BCD0 2844    LD      NZ,SYNEU2          ;NEIN SYCNH, NEU
BCD1 219888  LD      HL,#9888H          ;INIT PRUEFSUMME
BCD3 A3      DEFB      8AH          ;WARTET 23 TAKTE
BCD4 54      DEFB      8AH          ;
BCD5 78      DEFB      8A7H          ;
BCD6 D8      DEFB      8D8H          ;
;TEST AUF ABSCHLUSSMARKE
BCD7 5A      CP      D          ;
BCD8 2844    LD      NZ,SYNEU2          ;
BCD9 219888  LD      HL,#9888H          ;
BCCE 88      LDW32: NOP          ;
BCD8 88      LDW32: NOP          ;
BCD8 88      LDW32: NOP          ;
BCD8 88      LDW32: NOP          ;
BCD1 88      LDW28: NOP          ;
BCD2 88      LDW18: NOP          ;
BCD3 88      LDW12: NOP          ;
BCD4 88      LDW8: NOP          ;
BCD5 88      LDW4: NOP          ;
BCD6 E078    LD8M1: IN      A,(C)          ;SAMPLER-WERT
BCD7 88      LDW1: LD      R,HIGH          ;WARTET AUF STEIGENDE
BCD8 88      LDW1: LD      A,(C)          ;FLANKEN
BCD9 F288BC  ZOR      A,P,LOW          ;
BCD8 88      ZOR      A,P,LOW          ;
BCD8 88      EX      AF,AF          ;
BCD8 88      LD      NOP          ;
BCD8 88      LD      NOP          ;
BCD3 5981    INC      NC,NULL          ;ICH DANN 8
BCD5 23      INC      HL          ;SONST INC PRUEFSUMME
BCD6 78      JALL: LD      A,E          ;
BCD7 78      LD      A,E          ;
BCD8 D8      RET      C          ;RET BEI ENDELAD
BCD9 5F      LD      E,H          ;SPEICHER BIT
BCD8 88      LD      8,H          ;
BCD8 88      LD      8,H          ;
BCD8 32F1    LD      8,BILDP2,A          ;STERICHEN BLINKT

```

```

BCF2 344038 BACM: LD A,(TAB2) ;ZEICHEN VON TAB.
BCF3 FE84 CP B84H ;BREAK?
BCF7 00 NOP
BCF8 C20B8C JP NZ,LD01T ;LADE NACHSTES BIT
BCF9 C3008F JP START ;
BCD0 E078 HIGH: LD A,(C) ;WARTEN AUF FALLENDE
BCD8 F4FEB2 JP N,HIGH ;PLANNE
BCD3 37 SCF ;SETZE CARRY
BCD4 80 EX AF,AF' ;NACH AF' UND MOLE ALTES
BCD5 3F CCF ;CARRY UND INDETERIERE
BCD6 3801 JP NC,NUL2 ;1048 ERGIBT BIT=0
BCD8 23 INC HL ;INC PRUEFSUMME
BCD9 78 NULL2: LD A,E ;ROTIERE CARRY IN E
BCDA 1F RET C ;RETURN BEI ENDFLAG
BCDB 08 LD E,A ;
BCDC 5F LD E,A ;
BCDD E1 DEFB 0E5H ;WARTEN 21 TAKTE
BCDE E1 DEFB 0E1H ;
BCDF 1DE1 JP BRMC ;
;
;.....
;LDW1: WARTEN FUER 3600 BAUD
;.....
BC11 3E00 LDW1: LD A,(CON36) ;WARTEN 172 TAKTE
BC13 30 LDWCON: DEC A
BC14 28FD JP NZ,LDWCON
BC16 C30AB0 JP LCON ;
;
;.....
;EXIT NACH DOS, DOS-BASIC ODER ROM-BASIC
;.....
BC1F 34748 BACM: LD A,(ECC0C)
BC2C FE43 CP 83 ;
BC2E CAC20A JP 2,ROMBAS ;INS ROM-BASIC SPRINGEN
BC2F FE43 CP 43H ;
BC32 C30D48 JP 2,DOS ;INS DOS SPRINGEN
BC34 C3C00B JP 2,DOSBAS ;INS DOS-BASIC SPRINGEN
;
;.....
;DER PUFFERBEREICH
;.....
BC81 ERROR: DEFS 1
BC81 ZFLAG: DEFS 1
BC8C SUMAME: DEFS 12
BC8D SUPLEN: DEFS 1
BC8E SUTYP: DEFS 3
BC8F SUTYP: DEFS 1
BC92 SUTAG: DEFS 2
BC92 SULEN: DEFS 2
BC94 SUPREF: DEFS 4
BC9C LNAME: DEFS 12
BCA1 LDPUN: DEFS 1
BCA3 LUTYP: DEFS 3
BCA1 LUTAG: DEFS 1
BCA2 LOSTA: DEFS 4
BC56 LOSTH EQU LDSTA+1
BCA2 LULEN EQU 2
BCA4 LLOREF: DEFS 4
BC5C LCGNE EQU LDLOREF+3
;
BC5D 80 TEXT1: DEFB 000H
BC5E 52 DEFB 'SUPERTAPE FUER TRS-80, MODELL 1'
BC5F 80 DEFB 000H
BC7E 80 DEFB 000H
BC7F 28 DEFB ' (S)AVE (D)UICK (L)OAD (V)ERIFY (E)XIT'
BC84 80 DEFB 000H
BC87 40 DEFB 'COMMAND:'
BC88 0E DEFB 000H
BC81 23 DEFB 0
BC82 80 TEXT2: DEFB 000H
BC83 5C DEFB 'NAME:'
BC87 30 DEFB 000H
BC88 4E DEFB 'NAME.....,TYP'
BC87 4E DEFB 000H
BC8C 23 DEFB 0
BC8C 30 TEXT3: DEFB 000H
BC8A 53 DEFB 'START ENDE'
BC8C 4E DEFB 000H
BC8A 80 TEXT4: DEFB 000H
BC87 54 DEFB 'TAPE READY ? '
BC8E 80 DEFB 000H
BC8A 54 TEXT5: DEFB 'VERIFY'
BC8C 80 DEFB 000H
BC8D 4E DEFB 'NAME.....,TYP'
BC8C 30 DEFB 000H
BC8E 80 TEXT6: DEFB 000H
BC8F 4C DEFB 'LOAD'
BC8D 80 DEFB 000H
BC8A 4E DEFB 'NAME.....,TYP'
BC8C 80 DEFB 000H
BC14 23 DEFB 0
BC13 80 TEXT7: DEFB 000H
BC14 4E DEFB 'NEUE ADRESSE ? '
BC13 80 TEXT8: DEFB 000H
BC14 4E DEFB 'ADRESSE'
BC13 80 DEFB 000H
BC1F 23 DEFB 0
BC1E 40 TEXT9: DEFB 000H
BC13 80 TEXT10: DEFB 000H
BC1F 42 DEFB 'BASIC-START'
BC1D 80 DEFB 000H
BC1E 23 DEFB 0
BC1F 80 TEXT11: DEFB 000H
BC1E 40 DEFB 'MAIN-BASIC'
BC1E 80 DEFB 0
BC1E 23 DEFB 0
BC1E 80 END START

```

Thema: SUPERTAPE

Das im Info Nr. 7 (Seite 44-50) vorgestellte Bandaufzeichnungsverfahren SUPERTAPE ist sicher für einige Mitglieder des CLUB 90 so interessant, daß sie sich entschlossen haben, das Programm abzutippen und die Platine nachzubauen. Dazu zwei Tips:

1. Das Programm wird vom Heise-Verlag, der auch die Zeitschrift *c't* herausgibt für 7.-- DM auf Kassette verschickt: Man kann sich also die Mühe des Eintippens und Fehlertuschens sparen. Wer zudem noch etwas Zeit net, kann ja warten bis das Programm in der Clubbibliothek auftaucht (was sicher nicht allzu lange dauert).
2. Die Platine für die benötigte Zusatzschaltung wird für unter 20.-- DM von verschiedenen Inserenten in der *c't* angeboten. Die Schaltung kann also auch von den in der Hardware nicht so erfahrenen Clubmitgliedern aufgebaut werden.

Einige Diskuser werden sich natürlich fragen, was man mit einem Tapeaufzeichnungsverfahren, sei es auch noch so schnell, soll. Die Antwort ist relativ einfach. Mit SUPERTAPE erhält man die Möglichkeit, Programme und Daten mit anderen (supertapefähigen) Computern auszutauschen! Weiterhin kann man sich damit vielleicht manche Backupdiskette sparen.

Ein sicheres Plätzchen (für TRS80)!

In letzten Clubinfo wurde eine Schaltung veröffentlicht, mit der sich die Besitzer von VideoGenie-Computern ein sicheres Plätzchen für Maschinenroutinen in ihrer Maschine sichern können. Die Schaltung ist für TRS80-User nur mit erheblichen Änderungen brauchbar und somit mit Vorsicht zu genießen. Wer sich trotzdem in seinen TRS80 etwas mehr RAM einbauen möchte, der möge noch etwas Geduld haben. Ich bin z.Z. dabei eine Schaltung aus einer älteren amerikanischen Zeitschrift (80 US) auszutesten, die speziell für TRS80-Computer 1920 Bytes zusätzlichen Speicher bringt. Die Schaltung und der zugehörige Artikel werden dann im nächsten Info veröffentlicht (wenn alles funktioniert!).

Hauptstadt Oshana

Bei den meisten heutigen Hobby-Computern wird die Programmiersprache BASIC verwendet. Das heißt aber (leider) nicht, daß ein BASIC-Programm zwischen zwei verschiedenen Computertypen direkt oder über Kassette ausgetauscht werden kann. Die BASIC-Befehle mögen dann wohl das gleiche bedeuten, aber die interne Verarbeitung dieser Befehle und die Speicherung auf der Kassette ist häufig ganz anders. Nichts geht mehr ... ? Doch: Mit Hilfe von BASICODE, einer Art universeller, aus BASIC abgeleiteter Computersprache, ist es möglich, BASIC-Programme zwischen verschiedenen Computertypen auszutauschen.

# BASICODE-2

Endlich ein  
"allgemeinver-  
ständliches"  
BASIC

Es ist ungefähr zwei Jahre her, daß die Hobbykoop-Redaktion von Radio Hülversum in den Niederlanden auf die Idee kam, einen universellen Code auszuarbeiten, mit dem man BASIC-Programme zwischen verschiedenen Computern austauschen kann. Denn daß BASIC nicht gleich BASIC ist, merkt man schnell, wenn man zum Beispiel ein Junior-Programm auf dem ZX-81 laufen lassen will. Obwohl es ein Standard-BASIC gibt, verwendet jeder Computer-Hersteller seinen eigenen "Dialekt" mit seinen eigenen (meist durchaus sinnvollen) Erweiterungen. Der Unterschied zwischen ostfriesischem Platt und Ur-Bajuwarisch ist nichts dagegen ... Dazu kommen Probleme mit Speicheradressen und der internen Programmabarbeitung und als eines der wesentlichsten Probleme das der Übertragung. Bei den meisten Hobbycomputern wird der Kassettenrekorder als externer Massenspeicher verwendet. Die Speichermethodik (das Datenformat), die Übertragungsgeschwindigkeit und die gebrauchten Frequenzen sind auch wieder bei jedem Computertyp anders. Da half bisher nur eins: sich hinzusetzen, das Programm umzuschreiben und dann mühsam Zeichen für Zeichen in den eigenen Computer einzutippen. Der BASICODE-Standard ist ein festgelegter Audio-Code, mit dem BASIC-Programme auf der Kassette gespeichert werden können. Dabei ist es im Prinzip unwichtig, ob diese Programme vom Computer oder vom Radio oder vom Fernseher angeliefert werden. Denn daß diese drei "Lieferanten" in Frage kommen, wissen aufmerksame Elektor-Leser aus dem September-Heft. Durch die Standardisierung sind Abspeichern und Wiedereinlesen unabhängig vom Computertyp. Die einzelnen Festlegungen, auf die wir unten noch eingehen, betreffen:

- Übertragungsformat, -geschwindigkeit, -frequenzen
- zulässiger BASIC-Befehlssatz
- Speicheraufteilung
- Variablenbenennung
- Schirmformat

Das neue, universellere BASICODE-2 enthält neben anderen Verbesserungen vor allem eine Anzahl Standard-Unterprogramme. Dieses BASICODE-2 werden wir hier besprechen.

## BASICODE auf dem Band

Beim BASICODE werden die beiden Frequenzen 1200 Hz und 2400 Hz verwendet. Eine volle Periode der 1200 Hz bedeutet eine logische 0, zwei volle Perioden der 2400 Hz eine logische 1. Die einzelnen Bytes werden mit einer Geschwindigkeit von 1200 Baud gesendet beziehungsweise empfangen, wobei die Bytes folgendermaßen aufgebaut sind (siehe auch Bild 1):

- 1 Startbit (logisch 0)
- 8 Datenbits, beginnend mit dem LSB
- 2 Stoppbits (logisch 1).

Das BASIC-Programm wird Zeichen für Zeichen kodiert, also so, wie es beim Kommando LIST zu sehen ist. Von den internen Op-Codes des Computers wird kein Gebrauch gemacht: Alle Zeichen werden im ASCII-Kode dargestellt. Jedem BASIC-Befehl muß eine Leertaste folgen, jede BASIC-Zeile muß mit CR (Wagenrücklauf, Hex 8D) abgeschlossen werden. Das MSB jedes ASCII-Zeichens muß logisch 1 gemacht werden.

Ein vollständiges Programm auf der Kassette besteht aus folgenden Teilen:

- Ein "leader", bestehend aus einem 5 Sekunden langen Ton von 2400 Hz
  - dem ASCII-Zeichen "start text" (Hex 82)
  - dem BASIC-Programm in ASCII-Kode
  - dem ASCII-Zeichen "end of text" (HEX 83)
  - einer "checksum" (Prüfsumme)
  - einem "trailer", bestehend aus einem 5 Sekunden dauernden Ton von 2400 Hz.
- Die Prüfsumme ermöglicht es dem Anwender zu kontrollieren, ob das Programm fehlerlos eingelesen wurde. Sie wird durch die bitweise Exklusiv-ODER-Verknüpfung aller ihr vorangehenden Bytes einschließlich des "start text"-Zeichens gewonnen. Daher hat sie selber auch die Länge von 1 Byte (8 bit).

## Das BASICODE-2-Protokoll

Allgemeine Absprachen  
Es dürfen nur die BASIC-Befehle eingesetzt werden, die alle Computer verwenden. Diese Befehle sind in Tabelle 1 aufgelistet, wir kommen noch darauf zurück. Eine bestimmte Anzahl Programmzeilen ist für vorgegebene Unterprogramme reserviert. Diese Routinen ermöglichen Programmab-

läufe, die im Standard-BASIC nicht ohne weiteres möglich sind. Die Arbeitsweise dieser Unterprogramme wird noch beschrieben. Sie werden nicht mit dem Programm mitgesendet, sondern müssen im BASICODE-Übersetzungsprogramm enthalten sein oder vor dem Programmablauf gesondert eingegeben werden. Das Schirmformat ist festgelegt auf 24 Zeilen zu je 40 Zeichen. Da manche Computer noch weniger Zeilen beziehungsweise Zeichen abbilden, wird empfohlen, nicht mehr als 16 Zeilen für ein Schirmbild zu verwenden und die Zeilen nicht länger als nötig zu machen. Ferner darf eine Programmzeile einschließlich Zeilennummer, Leerzeichen und CR maximal 60 Zeichen lang sein.

### Programm-Aufbau

Bei den Zeilennummern wird bei BASICODE-2 folgende Einteilung eingehalten: 0-999: Standard-Unterprogramme. Diese Routinen sind speziell für den betreffenden Computer entworfen und werden mit dem Übersetzungsprogramm oder gesondert eingelesen.

1000: Erste Zeile des Hauptprogramms. Es muß folgende Form eingehalten werden: 1000 A = (Wert): GOTO 20: REM Programmname. (Wert) ist die maximale Anzahl Zeichen, die von allen Strings zusammen benötigt werden. Durch den Sprung nach Zeile 20 reservieren die Computer, bei denen dies notwendig ist, entsprechenden Speicherplatz für die Strings.

1010: Erste frei zur Verfügung stehende Programmzeile.

1010-32767: Platz für die eigenen Programme. Eine bestimmte Systematik in diesem Programm ist zwar nicht Pflicht, wird aber von den BASICODE-Entwicklern empfohlen:

1000-19999: Hauptprogramm.  
20000-24999: Unterprogramme mit Befehlen, die im BASICODE-2 nicht zulässig sind.  
25000-29999: DATA-Statements.  
30000-32767: REM-Statements. Diese können eine Beschreibung des Programms, Literaturhinweise und als letztes Namen und Anschrift des Programmverfassers enthalten. Weiter wird empfohlen, die Zeilennummern in 10er-Schritten zu setzen. Über die Unterprogramme in den Reihen 20000-24999 muß man anmerken, daß sie natürlich nach Möglichkeit zu vermeiden sind. Wenn dies

nicht geht, muß man wenigstens eine gute Beschreibung mitliefern, was in diesen Unterprogrammen geschieht.

### Standard-Routinen in BASICODE-2

Wie diese Unterprogramme jeweils aussehen, ist stark vom jeweiligen Computertyp abhängig, so daß wir hier nur eine allgemeine Beschreibung geben.

GOSUB 100: Dieses Unterprogramm löscht den Schirm und setzt den Cursor auf Position 0,0 (links oben auf dem Schirm).

GOSUB 110: Setzt den Cursor auf eine bestimmte Stelle des Schirms. Die gewünschte Position wird durch die Variablen HO und VE festgelegt. VE gibt die Nummer der Bildschirmzeile an (die oberste Reihe hat die Nummer 0) und HO den Platz in der Reihe (außen links ist 0). Da das Schirmformat bei BASICODE-2 24 Reihen zu je 40 Zeichen ist, darf VE nicht größer als 23 und HO nicht größer als 39 werden. Die Variablen VE und HO dürfen durch den Aufruf dieses Unterprogramms ihren Wert nicht ändern.

GOSUB 120: Die Position des Cursors auf dem Schirm wird durch die Variablen VE und HO gespeichert. Hierbei ist VE = 0 die oberste Reihe und HO = 0 die erste Position in der Reihe (ganz links). Dieses Unterprogramm kann zusammen mit dem vorigen dazu eingesetzt werden, die Position des Cursors per Programm beliebig zu verändern (zum Beispiel bei selbstgestrickten Bildschirmspielen).

GOSUB 200: Prüft, ob eine Taste gedrückt ist, und setzt den zu dieser Taste gehörigen Inhalt in die Variable INS. Ist in diesem Moment keine Taste gedrückt, ist INS leer. Im Prinzip kann man hier auch Steuerzeichen einlesen, aber dabei muß man vorsichtig sein, da diese bei manchen Computern eine unterschiedliche Bedeutung und Wirkung haben! Eine Ausnahme dabei ist RETURN. Das ist bei allen Heimcomputern der ASCII-Code 13.

GOSUB 210: Dieses Unterprogramm wartet, bis eine Taste gedrückt wird, und setzt dann den zugehörigen Inhalt in INS. Vorsicht beim Programmieren, sonst wird das die berühmte Endlos-Warteschleife.

GOSUB 250: Erzeugt bei den Computern, bei denen dies möglich ist, einen Piepser. Tonhöhe und Tondauer sind hierbei nicht festgelegt.

GOSUB 260: Eine Zufallszahl zwischen 0 und 1 wird erzeugt und in die Variable RV eingeschrieben.

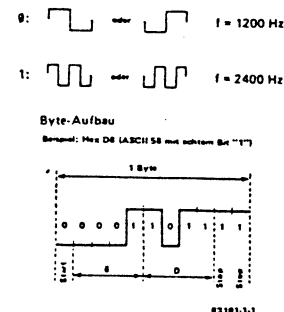
GOSUB 270: Der Variablenspeicher wird überprüft, und es wird festgestellt, wieviel Speicherplatz noch zur Verfügung steht. Die Variableninhalte werden dabei nicht verändert. Die Anzahl der freien Bytes wird in die Variable FR eingelesen.

GOSUB 300: Der Wert der Variablen SR wird als String in SR\$ eingelesen. Der String darf kein Leerzeichen am Anfang oder Ende der Zahl enthalten (im Gegensatz zum Statement STR\$, dieses Statement darf bei BASICODE-2 nicht verwendet werden).

GOSUB 310: Dieses Unterprogramm erzeugt einen String SR\$, der folgendermaßen aufgebaut ist:

Der Wert von SR\$ ist gleich dem Inhalt der Variablen SR und immer in Festkomma-

Bild 1. Der Aufbau des Übertragungsformates bei BASICODE. Achtung: Die Übertragung beginnt mit dem LSB (least significant bit = niederwertigstes Bit)!



Notation. Die Gesamtlänge von SR\$ beträgt CT Zeichen, die Anzahl der Zeichen nach dem Dezimalkomma wird durch CN festgelegt. Wenn die Zahl nicht in das vorgegebene Format paßt, besteht SR\$ aus CT Sternen. Wenn nötig, wird SR entsprechend gerundet. CT, CN und SR werden durch den Zugriff auf dieses Unterprogramm nicht verändert. Ein Beispiel: CT = 7, CN = 3 und SR = 0,6666 ergibt in SR\$ " 0,667".  
GOSUB 350: Printed SR\$, schließt aber die Zeile noch nicht ab. Dadurch ist es möglich, verschiedene Strings hintereinander in eine Zeile zu schreiben.  
GOSUB 360: Abschluß einer Schreibzeile mit einem Wagenrücklauf- und Neue-Zeile-Kommando.

#### Variable

Um die Austauschbarkeit der Programme zu garantieren, sind auch die Variablen bestimmten Festlegungen unterworfen.

- Numerische Variablen sind immer reell und mit "single precision".
- Der Name einer Variablen darf höchstens aus zwei Zeichen bestehen, wobei das erste Zeichen ein Buchstabe sein muß. Das zweite Zeichen kann, wenn nötig, aus einem Buchstaben oder einer Ziffer bestehen. Bei String-Variablen steht noch ein \$ nach dem Namen. Kleinbuchstaben dürfen in Variablennamen nicht verwendet werden.
- Bei logischen Variablen wird lediglich auf wahr oder unwahr geprüft. Eventuell vom Computer erzeugte Werte, zum Beispiel +1 für wahr und 0 für unwahr, dürfen nicht verwendet werden.
- Man darf nicht davon ausgehen, daß eine Variable durch den Programmstart automatisch auf 0 gesetzt wird.
- Stringvariablen dürfen maximal 255 Zeichen lang sein.
- Namen von Variablen dürfen nicht mit dem Buchstaben O beginnen, dieser ist reserviert durch die Standard-Unterprogramme.
- Weiter sind folgende Variablennamen nicht zulässig: AS, AT, FN, GR, IF, PI, ST, TI, TIS und TO.

Für den Datenaustausch zwischen dem BASIC-Programm und den Standard-Unterprogrammen werden die Variablen HO, VE, FR, SR, CN, CT, RV, INS und SR\$ verwendet.

#### BASIC-Abstrachen

Tabelle 1 gibt eine Übersicht über alle BASIC-Kommandos und -Operatoren, die verwendet werden dürfen. Auch in diesem Punkt sind strenge Absprachen nötig, da

es verschiedene BASIC-Varianten gibt. Die Bedeutung der Befehle stimmt mit der offiziellen BASIC-Version überein, so daß wir hierauf an dieser Stelle nicht eingehen. Einige spezielle Punkte müssen wir bei den BASIC-Befehlen noch besprechen. So daß man nach einem GOSUB oder GOTO kein Variablennamen verwenden. Die Form: A = 1000 : GOTO A darf also nicht verwendet werden. Nach IF muß immer ein THEN folgen. Also: IF ... THEN A = 5 IF ... THEN 1000 oder IF ... THEN GOSUB 20000. Die Form IF ... THEN ... ELSE ist verboten. Kommentare oder mehrere Variablen nach einer INPUT-Anweisung sind nicht zugelassen. Die Form INPUT "Der Wert = "; A\$ ist verboten. Eine Zeilennummer nach RUN ist nicht zulässig. Beim TAB-Befehl muß man berücksichtigen, daß der eine Computer bei Null und der andere bei Eins zu zählen beginnt.

#### Die Praxis

So, das waren die wichtigsten BASICODE Absprachen. Um damit zu arbeiten, braucht man jetzt noch das Übersetzungs-Programm und die Standard-Unterprogramme. Dazu müssen wir für die "Allgemeinheit" an die Stelle passen: Denn hier gilt (hoffentlich zum letzten Mal), daß halt jeder Computer etwas anders ist. Glückliche Junior-Computer-Besitzer: Sie finden das Übersetzungsprogramm und die Standard-Unterprogramme sowohl für den erweiterten Junior mit Microsoft-Basic als auch für den DOS-Junior an anderer Stelle in diesem Heft. Und wenn sie die Sendung "Know-how-Special" empfangen können, bekommen sie das auch noch vom Fernsehen auf Kassette geliefert (siehe Elektor im September-Heft). Und die Besitzer der vielen anderen Computer? Wenn alles gut gegangen ist, ist inzwischen das BASICODE Buch Nummer 2 erschienen – allerdings in niederländischer Sprache. Neben einer ausführlichen Beschreibung von BASICODE enthält es auch die Übersetzungsprogramme für diverse Computertypen. Sobald das Buch der Redaktion vorliegt, werden wir unsere Leser noch genauer informieren. Müssen nun diejenigen, denen weder Rad Hülversum noch ein Fernsehprogramm noch dieses Elektorheft ein Übersetzungsprogramm liefert, etwa aufgeben? Nun, es gibt schließlich viele, auf einen bestimmten Computertyp spezialisierte Computerclubs bei denen man einmal anfragen kann. Natürlich stellt sich jetzt mancher Leser die Frage: Kann ich denn, wo ich als Hamburg schon den WDR nicht empfangen kann, wenigstens Hülversum II empfangen? Leid können wir da auch keine eindeutige Antwort geben, weil dies von verschiedenen "funktchnischen" Bedingungen abhängt. Es kann vorkommen, daß ein Elektor-Leser in Bayern einen guten Empfang hat, während ein Leser im viel näher gelegenen Hessen leer ausgeht. Aber was spricht dagegen, durch Leserbriefe an Fernseh-Intendanten dazu beizutragen, daß Know-how-Special in alle 3ten Fernseh-Programme kommt?

## Für TRS-80 und Video-Genie

# BASICODE-Interface

Um Basicode-Programme von der Kassette in den TRS-80 oder den Video-Genie laden zu können, benötigt man ein spezielles Interface, mit dem die FSK-Signale in Rechteckimpulse umgewandelt werden. Die Schaltung (Bild 1) besteht aus einem Impulsaufbereiter-Teil (A1, A2 und "Umgebung"), einem Adreßdekoder (N1... N9 und N17) und einem Indikator (A3, D3). Sie wird an der Erweiterungsanschlußleiste des Computers angeschlossen, deren Belegung aus dem jeweiligen Handbuch hervorgeht.

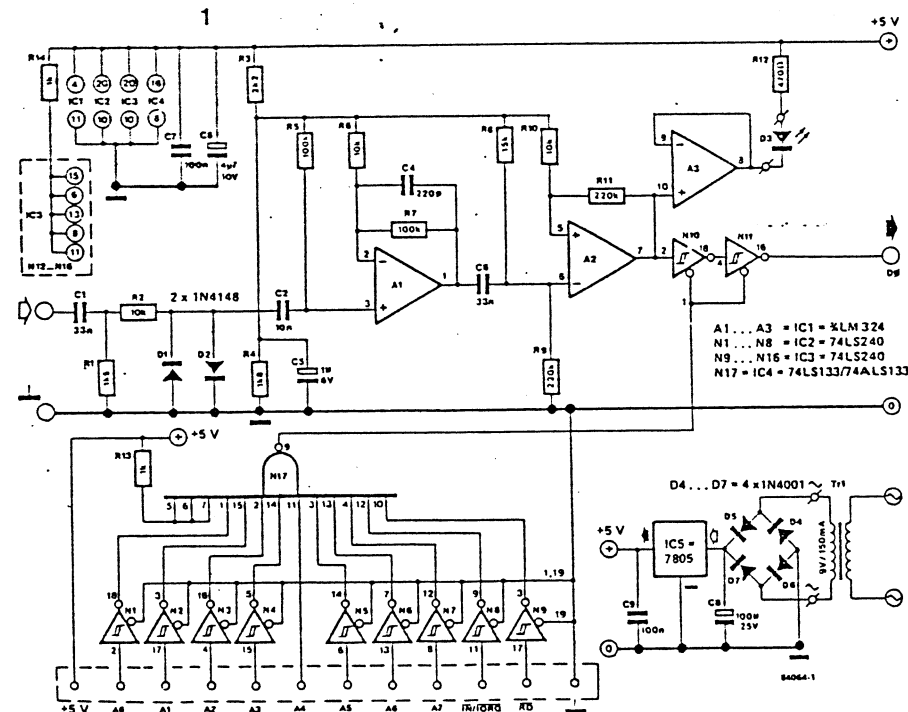
#### Die Schaltung

Wenn der Ausgang von N17 auf "0" ist, wird das Signal auf die Datenbuslinie D0 durchgegeben. Das Übersetzungsprogramm – es muß vorher eingegeben beziehungsweise über den normalen Kassettenschluß eingelesen werden – sorgt hier auch für die Umwandlung der seriellen Eingangssignale in "normale Bytes" und deren Abspeicherung. Der Adreßdekoder wird durch das Übersetzungsprogramm so angesteuert, daß beim Einlesen N10 und N11 freigegeben sind. Die Dioden D1 und D2 begrenzen das Signal vom Kassetteneinkoder. Mit A1 ist eine Verstärkerstufe aufgebaut, mit A2 ein Schmitt-Trigger. Der OP A3 dient als Treiber für die Indikator-LED. Ist kein Eingangssignal vorhanden, sorgt R9 dafür, daß der Ausgang von A2 auf "1" liegt. Die LED D3 leuchtet dann nicht. Im Signalweg liegen die beiden invertierenden Treiber N10 und N11, sie werden vom Ausgang des Adreßdekoders gesteuert.

#### Aufbau

Auf der Platine (Bild 2) finden – mit Ausnahme des Trafos – auch noch die Bauteile für ein kleines Netzteil Platz. Allerdings braucht man diese nur, wenn das

Bild 1. Die Schaltung des BASICODE-Interface für TRS-80 und Video-Genie.





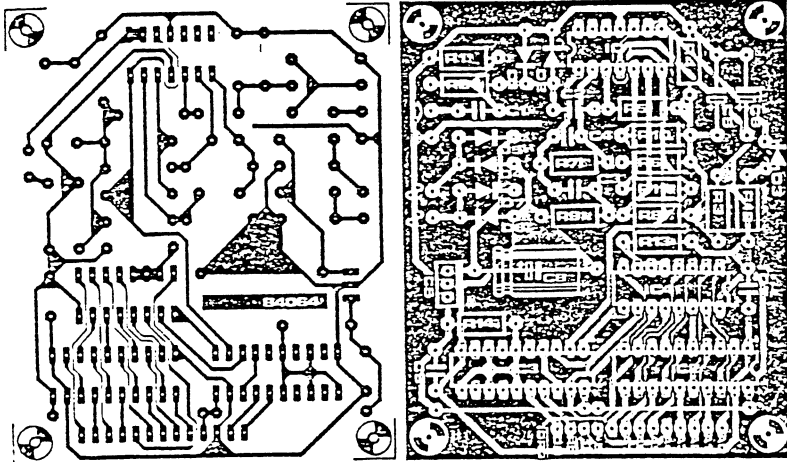


Bild 2. Mit der Platine fällt der Aufbau der Interface-Schaltung leicht.

Netzteil des Computers bereits "anderweitig" zusätzlich belastet ist: Die Schaltung des Interface benötigt lediglich 50 mA. In den meisten Fällen kann man also den 5-V-Anschluß auf der Erweiterungs-Anschlußleiste als "Kraftwerk" verwenden.

Auf der Platine gibt es eine Leiterbahn, die "im Nichts endet". Das ist kein Fehler, diese Leiterbahn dient als einfache Abschirmung des Eingangs gegen die direkt daneben anliegende Netzteil-Wechselspannung.

Abschließend noch ein Hinweis: Über das BASICODE-Verfahren ist ausführlich im Oktober-Heft 1983 berichtet worden.

#### Neues von BASICODE

Über das BASICODE-Verfahren haben wir ausführlich in Elektor vom Oktober 1983 berichtet. Das BASICODE-2-Buch (in Niederländisch und Englisch) von Radio Hilversum liegt der Redaktion inzwischen vor. Es enthält die Anpassungen und Übersetzungsprogramme für folgende Computer beziehungsweise Betriebssysteme:

Apple 2, BBC-Computer Modell A und B, Commodore (Pet und VC), CP/M-Systeme, DAI-Computer, Exidy Sorcerer, OSI-IP-Computer, Philips P-2000, Sharp MZ80 A und MZ80 K, SWTPC-6800, TRS-80 (I und III) und Video-Genie. Weitere Anpassungen sind bei Radio Hilversum — genauer: bei der Hobbyskop-Redaktion — in Vorbereitung. Außerdem soll in Kürze bei einem deutschen Verlag ein Sonderheft über BASICODE — ebenfalls mit Anpassungen — erscheinen, hier wissen wir aber noch nicht, für welche Computer (Rubrik Büchermarkt beachten).

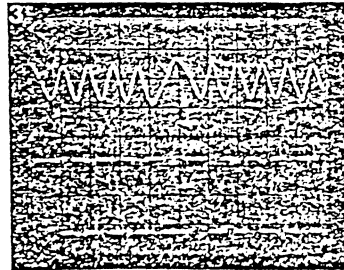


Bild 3. Auf dem Foto erkennt man oben das Eingangssignal und unten das Ausgangssignal der Interface-Schaltung.

#### Stückliste

##### Widerstände:

R1 = 1k5  
R2, R6, R10 = 10 k  
R3 = 2k2  
R4 = 1k8  
R5, R7 = 100 k  
R8 = 15 k  
R9, R11 = 220 k  
R12 = 470 Ω  
R13, R14 = 1 k

##### Kondensatoren:

C1, C3 = 33 n  
C2 = 10 n  
C4 = 220 p  
C5 = 1 µ/6 V  
C6 = 4µ7/10 V  
C7, C9 = 100 n  
C8 = 100 µ/25 V

##### Halbleiter:

D1, D2 = 1N4148  
D3 = LED  
D4, D5, D6, D7 = 1N4001  
IC1 = LM 324  
IC2, IC3 = 74LS240  
IC4 = 74LS133 (oder 74ALS133)  
IC5 = 7805

außerdem:  
Tr1 = Netztrafo 9 V/  
150 mA sek.  
Platine 84064

Abschließend nochmals die Adresse von Radio Hilversum:  
NOS-Hobbyscoop  
postbus 1200  
1200 BE Hilversum  
Niederland

Jan Lichtermann

## Allerlei Nützliches für den TRS-80

TRS-80-Besitzer müssen manche Kleinigkeit bei ihrem Computer in Kauf nehmen, die auf die Dauer ärgerlich ist. In den meisten Fällen kann man mit wenig Aufwand Abhilfe schaffen. Die im folgenden angesprochenen Tips und Kniffe haben sich in der Praxis vielfältig bewährt.

Die Dauer eines Tastendrucks wird festgestellt

Die INKEY-Funktion hat den Nachteil, daß nach einmaligem Tastendruck die Dauer der Betätigung nicht ermittelt werden kann. Möchte man mit einer Taste, z. B. in einem Grafikprogramm, eine Dauerfunktion erreichen, so kann man die Tastatur mit der PEEK-Funktion direkt abfragen und in einer Programmschleife einen Befehl so lange ausführen, wie die Taste betätigt wird. Die Tabelle gibt Aufschluß darüber, welches Argument zur Abfrage einer bestimmten Taste erforderlich ist. Wird die Taste betätigt, so erhält man den zugehörigen Wert als Resultat der Abfrage. Ist keine Taste betätigt, ergibt die PEEK-Abfrage den Wert 0. Bild 1 verdeutlicht das Prinzip der Tastaturabfrage durch PEEK. Soll aus einem bestimmten Grund, etwa zur Erhöhung der Aufmerksamkeit, eine Stringvariable durch einen blinkenden Cursor eingelesen werden, so empfiehlt sich die Nachbildung des INPUT-Befehls durch die INKEY-Funktion. Durch die Wahl der Schleifenlänge kann man die Blinkfrequenz des Cursors beeinflussen. Bild 2 zeigt, wie das Einlesen eines Strings mit der INKEY-Funktion bewerkstelligt wird.

#### Ausgabe im technischen Format

Obwohl das TRS-80-Basic die komfortable PRINT-USING-Anweisung beinhaltet, ergeben sich bei der Ausgabe von

Zahlen manchmal Schwierigkeiten. Von der Technik her schätzt man das Arbeiten mit den Kürzeln plko ( $10^{-12}$ ), nano ( $10^{-9}$ ), Mega ( $10^6$ ), Tera ( $10^9$ ), Giga ( $10^{12}$ ). Von Kürzel zu Kürzel unterscheidet sich der Exponent um jeweils 3. Eine automatische Anpassung wäre deshalb in vielen Fällen wünschenswert. Das in Bild 3 dargestellte Unterprogramm erledigt das. Übergeben wird die auszugebende Zahl Z, ebenfalls verwendet werden die Variablen ZZ und V. Letzteres kann eine Ganzzahlvariable sein, lediglich ZZ und Z müssen vom selben Typ sein, also entweder von einfacher oder doppelter Genauigkeit.

#### Tastendrücken und Ladeprobleme

Vielen mag das Tastendrücken beim TRS-80 schon Kopfzerbrechen bereitet haben. Sind die Tastenkontakte verschmutzt, so werden Buchstaben zum Teil doppelt geschrieben. Tandy empfiehlt dagegen das Reinigen der Tastaturkontakte oder die Verwendung eines kleinen Maschinenspracheprogramms KBFIX. Dieses Programm konnte ich von Tandy kostenlos erhalten, und es arbeitet zu meiner vollen Zufriedenheit. Ist das Programm geladen, wird das Tastaturdrücken wirkungsvoll per Software eliminiert.

Ein anderes beinahe noch kritischeres Problem ist das Laden der Programme vom Kassettenrecorder. Dabei ist die Lautstärke am Recorder peinlich genau

Tabelle zur Tastaturabfrage

Zeichen	PEEK()	Wert
@	14337	1
A	14337	2
B	14337	4
C	14337	8
D	14337	16
E	14337	32
F	14337	64
G	14337	128
H	14338	1
I	14338	2
J	14338	4
K	14338	8
L	14338	16
M	14338	32
N	14338	64
O	14338	128
P	14340	1
Q	14340	2
R	14340	4
S	14340	8
T	14340	16
U	14340	32
V	14340	64
W	14340	128
X	14344	1
Y	14344	2
Z	14344	4
frei	14344	8
frei	14344	16
frei	14344	32
frei	14344	64
frei	14344	128
0	14352	1
1	14352	2
2	14352	4
3	14352	8
4	14352	16
5	14352	32
6	14352	64
7	14352	128
8	14368	1
9	14368	2
:	14368	4
+	14368	8
<	14368	16
=	14368	32
>	14368	64
/	14368	128
ENTER	14400	1
CLEAR	14400	2
BREAK	14400	4
↑	14400	8
↓	14400	16
←	14400	32
→	14400	64
SPACE	14400	128
SHIFT	14464	1
frei	14464	2
frei	14464	4
frei	14464	8
frei	14464	16
frei	14464	32
frei	14464	64
frei	14464	128

```
10 A=PEEK(14337):IF A=2 THEN PRINT"A";GOTO 10
20 IF A=4 THEN PRINT"B";GOTO 10
30 IF A=8 THEN PRINT"C";GOTO 10
40 GOTO 10
```

Bild 1. Das Programm gibt so lange einen der Buchstaben A, B oder C aus, wie die entsprechende Taste gedrückt ist

```
10 B$=""
20 PRINT CHR$(14);IFOR N=1TO50:AS=INKEY$:IF AS=""THEN NEXT ELSE GOSUB 100:NEXT
30 PRINT CHR$(15);IFOR N=1TO50:AS=INKEY$:IF AS=""THEN NEXT ELSE GOSUB 100:NEXT
40 GOTO 20
100 IF ASC(A$)=13 THEN PRINT CHR$(14);A$;GOTO 130
110 IF ASC(A$)=8 AND LEN(B$)>0 THEN B$=LEFT$(B$,LEN(B$)-1):PRINTA$;RETURN
120 IF ASC(A$)>8 THEN B$=B$+A$:PRINTA$;
125 RETURN
130 REM STRING STEHT ALS B$ BEREIT, NACH BEARBEITUNG MIT RETURN ZURUECKKEHREN !
```

Bild 2. Das Programm liest einen String ein, dabei wird ein blinkender Cursor erzeugt - Enter schließt den String ab

```
60000 IF Z=0 THEN V=1:GOTO60010 ELSE ZZ=ABS(Z):V=FIX(LOG(ZZ)/LOG(10))-FIX(LOG
(ZZ)/LOG(10)/3)+3+2:IF ZZ<1 THEN V=V+2
60010 PRINTUSING STRING$(V,"");"*****";Z:RETURN
```

Bild 3. Das Unterprogramm gibt die Zahl Z im technischen Format aus

einzustellen, da sonst immer wieder Ladefehler auftreten. Auch ist die Lautstärke je nach Bandsorte verschieden zu wählen. Um so erfreuter war ich, als ich las, daß Tandy eine Zusatzplatine einbaut. Der kostenlose Einbau (er erforderlich war einige Schreiarbeit meinerseits, da die Firma Tandy nicht sofort reagierte) hat sich durchaus gelohnt. Ladefehler treten jetzt nur noch in Ausnahmefällen auf.

Weitere Hardwareverbesserungen einfachster Art sind der zusätzliche Einbau zweier Tastschalter rechts neben der Tastatur. In der Klappe zur Erweiterung des TRS-80 befindet sich ein Taster, der beim Schieflaufen des Ladevorgangs bzw. eines Maschinenspracheprogramms betätigt werden kann und einen nichtmaskierbaren Interrupt am Prozessor auslöst. Die Betätigung dieses Tasters ist recht umständlich: Man muß jedesmal die Klappe öffnen und ihn auf der Rückseite mühsam ertasten. Deshalb kann man einen einfachen Taster parallel dazu legen und neben der Tastatur montieren. Dies empfiehlt sich sehr für permanente mit Maschinensprache.

Der Prozessor kann damit jederzeit ohne Programmverlust unterbrochen werden. Bekannt ist jedem TRS-80-Anwender die nach dem Einschalten des TRS-80 erscheinende Frage: MEMORY SIZE?, die zur Reservierung eines vor Basic geschützten Arbeitsbereichs für Maschinensprache dient. Ausgelöst wird dieser Vorgang hardwaremäßig nach dem Einschalten durch den sogenannten Power-On-Reset. Arbeitet man viel mit Maschinensprache, so sind manchmal Umreservierungen nötig. Dies würde ein Aus- und Einschalten der Anlage erforderlich machen, wobei die Anlage einige Sekunden ausgeschaltet bleiben muß. Hier kann man einen weiteren Taster einbauen, der diesen Power-On-Reset simuliert. Dadurch braucht man die Anlage nicht an der Rückseite ein- und auszuschalten.

Betrachten wir einmal den Schaltungsausschnitt in Bild 4: Wird das Gerät ein-

geschaltet, so ist C42 entladen; am Eingang von Z53 liegt also L-Pegel an, damit auch am Ausgang von Z52. Dies veranlaßt den Prozessor dazu, mit der Befehlsausführung bei Adresse 0000 zu beginnen, sobald der Reset-Eingang wieder H-Pegel erhält. Dies geschieht dadurch, daß C42 über R47 langsam aufgeladen wird. Im Betriebszustand ist C42 also geladen. Legt man jetzt einen Taster parallel zu C42, kann man ihn auch während des Betriebs entladen und damit den Einschaltvorgang simulieren. Dazu muß man C42 auf der Platine suchen und eine entsprechende Leitung anlöten. Ich verwende hierzu eine abgeschirmte Leitung, damit im Videoteil keine wilden Schwingungen ausgelöst werden. Doch Achtung! Versehentliches Betätigen dieses Tasters löscht ein Basic-Programm im Speicher. Auch verliert man durch einen solchen Eingriff die Garantie des Herstellers.

#### Der Speicher wird erweitert

Bekanntlich kann ein 4-KByte-Modell auf 16 K erweitert werden. Dazu werden die Speicherchips ausgetauscht. Zusätzlich müssen zwei Shunts ausgetauscht werden. Führt man diese Arbeiten selbst aus, kann man eine Menge Geld sparen! Benötigt werden acht Chips vom Typ 4116 (450 ns oder schneller). Es handelt sich um dynamische NMOS-RAMs. Beim Einbau sind unbedingt die Handhabungsregeln von MOS-Bausteinen zu

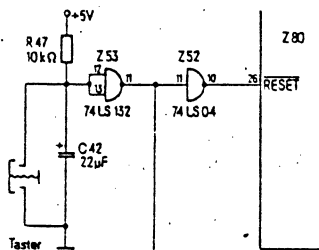


Bild 4. So wird der Power-On-Reset beim TRS-80 erzeugt

beachten, d. h. auf jeden Fall ist statische Aufladung zu vermeiden. Man entfernt die alten Speicherchips Z13 bis Z20 und setzt unter Beachtung der Richtung (Kerbe) die neuen ICs ein. Dann wechselt man die DIP-Shunts A3 und A71 aus. Das folgende gilt nur für die D-Version des TRS-80 (letzter Buchstabe der Nr. an Unterseite des Gehäuses) und „BASIC LEVEL II“: Bei A3 sind sämtliche Kontakte zu überbrücken, bei A71 sind Brücken gemäß Bild 5 herzustellen. Möchte man keine Shunts kaufen, so kann man die Verbindung mit Schweißdraht herstellen. Korrekte Verbindungen sind die erste Voraussetzung für den Betrieb, sonst läuft der Computer nicht mehr. Hat man eine andere Platinenversion oder kein Level-2-Basic, so sind die Brücken anders herzustellen! Die Funktion des eingebauten neuen Speichers läßt sich durch die Eingabe von PRINT MEM grob überprüfen: Bei leerem Arbeitsspeicher muß die Zahl 15572 erscheinen.

#### Literatur

- [1] TRS-80 micro computer technical reference handbook. Radio Shack.
- [2] Stübs, Martin: Programmieren mit TRS-80. Ing. W. Hofacker GmbH.

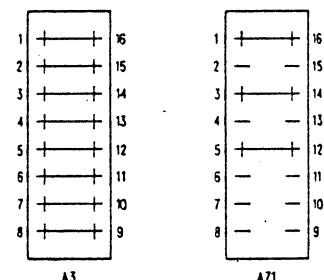


Bild 5. Erweitert man den Speicher des TRS-80 von 4 KByte auf 16 KByte, dann müssen bei der D-Version (letzter Buchstabe der Nummer auf der Unterseite des Gehäuses) die eingezeichneten Verbindungen hergestellt werden

## Billiges CP/M von Bill Brewer

übersetzt von G. Schröder

Das Modell III kann in seinem ursprünglichen Zustand keine CP/M-Programme laufen lassen. Aber dieser Bericht zeigt Ihnen, wie Sie eine einfache Modifikation mit zwei Chips für fünf Dollar bauen können.

Das ROM belegt im Modell III die Adressen 8000-37FF hexadezimal. Außerdem liegen die Tastaturadressen und der Bildschirmspeicher in 37FF-3FFF. Nur der Bereich von 4000 bis FFFF kann im Modell III als Lese/Schreib-RAM benutzt werden. Die Aufteilung des Speichers wird in Bild 1 gezeigt.

Das ROM liegt im Modell III leider genau dort, wo auch die Programme unter CP/M stehen würden. CP/M würde Daten an der gleichen Stelle speichern, an der das Modell III-ROM ab 8000 liegt, und CP/M lädt alle Programme, die unter ihm laufen, in den Speicher ab 180 hexadezimal. CP/M und alle Programme, die unter ihm laufen, brauchen einen durchgehenden Speicher von 8000 bis zum Ende des Speichers, wo das Betriebssystem steht.

Abgesehen von dieser Speicherabweichung ist die Modell III Hardware durchaus dazu geeignet, leicht modifizierte 8000 oder 280 Betriebssysteme laufen zu lassen; auch CP/M.

Diese Hardware-Begrenzung im Modell III wird durch nur sehr wenige der Schaltkreise dieses Computers verursacht. Wenn man sich diese Schaltkreise ansieht, wird rasch ein schneller und leichter Weg sichtbar, um diese Hindernisse aufzuheben.

Die Modell III CPU adressiert Signale über 16 Drähte in seinem Adressbus. Ein unmodifiziertes Modell III führt seine beiden höchstwertigen Adressdrähte in die eine Hälfte eines 74LS139 Multiplexers (U58), der einen Decoder Schaltkreis darstellt. Dieser Decoder gibt Signale aus, um eine der vier 16K-Speicherbänke einzuschalten (Bild 2).

Wenn beide der höchstwertigen Adressdrähte (A14 und A15) auf einem niedrigen logischen Level sind, schaltet der Decoder die 16K-Speicherbank ein, die das Modell III ROM sowie die Tastatur und den Bildschirmspeicher enthält. Abhängig von den binären Werten dieser beiden Adressdrähte wird eine der vier 16K-Speicherbänke im Modell III Adressenraum ausgewählt (Tabelle 1).

Ein zweiter Decoder, der die untere Hälfte desselben Multiplexers benutzt, produziert Einschalt-Signale in der 16K-Bank, die das ROM enthält. Indem er Adressdraht A14 und den 8000-3FFF-Einschalter (ROMENABLEX) benutzt, schaltet dieser Decoder entweder den 8K ROM-Chip ein, der das meiste des Modell III Basic enthält, oder eine andere Gruppe von Schaltkreisen mit einem 4K-ROM, einem 2K-ROM, der Tastatur und dem Bildschirmspeicher. Dieser Segment-Decoder wählt diese verschiedenen Routinen zur Benutzung in ihrem jeweiligen Speicherplatz wie in Tabelle 2 gezeigt.

### SIXTEEN KILOBYTE ADDRESS BANK SELECTED BY MRQ\*, A16, AND A15

MRQ*	A16	A15	BANK SELECTED
HIGH	EITHER	EITHER	NONE
LOW	LOW	LOW	0000-3FFF
LOW	LOW	HIGH	4000-7FFF
LOW	HIGH	LOW	8000-BFFF
LOW	HIGH	HIGH	C000-FFFF

Table 1. 16K Bank Selection by U58

**Table 2. Segment Selection in Bank Containing ROM**

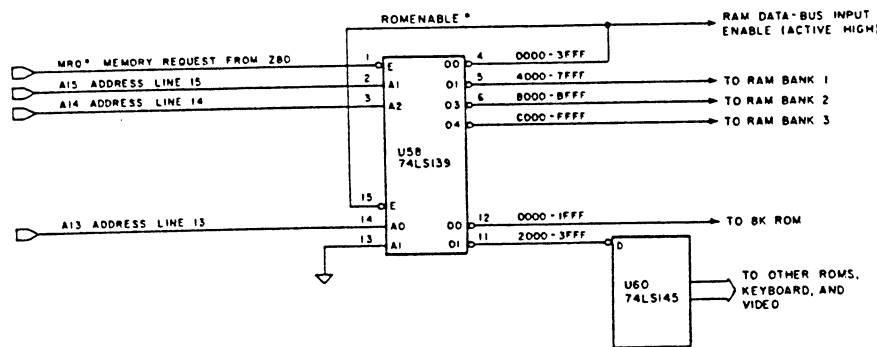
Die einfachste Lösung, um das Modell III zu einer CP/M-fähigen Maschine zu machen, ist, das ROM aus dem Adressenbereich auszuschalten, den die meisten Programme benutzen, die nicht für das Modell III geschrieben wurden, und dafür das RAM einzuschalten. Das sieht zunächst sehr einfach aus. Man kann dies erreichen, indem man die Leiterbahnen, die im Bild 2 mit 0000-3FFF und C000-FFFF bezeichnet sind, vertauscht. Somit wäre das ROM, die Tastatur und der Bildschirm in C000-FFFF und die 16K-RAM-Bank der Bank 3 würde von 0000 bis 3FFF adressiert werden.

Dieser einfache Ansatz bringt aber zwei Nachteile mit sich. Erstens können keine Programme mehr laufen, die für das Modell III geschrieben wurden, denn ROM, Tastatur und Bildschirm würden ständig aus dem Speicherbereich herausgeschaltet werden, in dem sie laufen müssen.

Zweitens würde der Prozessor - ohne entsprechende Änderungen seines Kontroll-Schaltkreises - bei jedem RESET versuchen, ein Programm an der Adresse 0000 im RAM auszuführen; es gibt keine Möglichkeit, ein Programm in das RAM zu laden, bevor der Prozessor beginnt, von ihm Instruktionen zu holen.

Sie können die ursprünglichen Fähigkeiten bewahren, wenn Sie elektronische Schalter benutzen. Elektronische Schalter können die Adressierung so ändern, daß 0000 die Startadresse des ROMs ist, wenn Programme unter TRSDOS etc. laufen und daß 0000 die Startadresse des ROMs ist, wenn Sie Programme unter CP/M laufen lassen.

Elektronische Schaltungen achten auch auf das Problem mit dem Laden des Programms: Mit einem elektronischen Schalter können Sie Programme mit Routinen aus dem ROM laden, bevor der Schalter die ROM- und RAM-Adressen tauscht. Somit versorgt ein elektronischer Schalter das Programm mit einer dauernden Kontrolle durch den Prozessor.



Ein Weg, um die elektronische Speicher-Umschaltung zu erreichen, ist, die Adressensignale des Z80 zu bearbeiten, bevor sie den Decoder erreichen. Während sich der Speicher im Standard Modell III-Modus befindet, kann der Z80 Instruktionen im Adressbereich von 4000 bis BFFF ausführen, um die Signale von A15 und A14 zu invertieren, wenn beide WAHR oder beide FALSCH sind. Somit wird der Decoder genarrt, denn er denkt, daß eine vom Prozessor gelieferte Adresse im Bereich 0000-3FFF in Wirklichkeit im Bereich C000-FFFF liegt, und der Prozessor hat benutzbares RAM bei 0000.

Es gibt einige kommerzielle Modifikationen, die den Speicher in dieser Weise umschalten. Diese Methode hat einen Vorteil und einen Nachteil. Das Vorbereiten der Adressensignale ist mechanisch bequem; es kann durch eine einzelne Huckepack-Platine im Z80-Sockel erreicht werden, welche natürlich wieder den Z80 in einem Sockel aufnehmen muß.

Der Vorteil ist, daß der Computer nicht physisch verändert werden muß; die Huckepack-Platine ist keine unwiederrufliche Veränderung, denn der Z80 ist einer der wenigen gesockelten Chips im Modell III. Der Nachteil ist, daß dieses Vorbearbeiten der Signale natürlich eine Verzögerung vor dem Erreichen des Decoders und der RAM-Bänke bewirkt, wodurch RAM-Operationen unzuverlässig werden. Diese Verzögerung ist im Modell III kritischer als im Modell I, denn die Speicher-Zyklus-Zeit (memory-cycle time) ist im Modell III um annähernd 20% kürzer als im Modell I.

Eine bessere Lösung stellt der Ersatz des Multiplexer-Schaltkreises (U58) durch einen neuen Decoder-Schaltkreis dar, wobei dieser die Adresszuweisungen für ROM und RAM umschalten kann und trotzdem nicht laeseramen im Anprechen der Adressen ist als das Originalteil.

Der neue Decoder, der den 16K Bank-Wahl Decoder in der oberen Hälfte von U58 ersetzt, besteht aus zwei anwählbaren Decodern in einem 74LS156 (Demultiplexer mit offenem Collector). Einer der neuen Decoder kann das ROM mit Tastatur und Bildschirm von 0000-3FFF und die RAM-Bank 3 von 0000-FFFF einschalten (wie der Original Modell III Decoder). Der andere neue Decoder schaltet das ROM + Tastatur-Matrix + Bildschirmspeicher in C000-FFFF und RAM-Bank 3 in 0000-3FFF ein. Dieser Schaltkreis, um den 74LS156 Demultiplexer (genannt U58-1) herum aufgebaut, wird in Bild 3 gezeigt.

Es ist jederzeit immer nur einer der Decoder aktiv. Ein aktiv-HIGH Signal (REMAP), extern an die Decoder herangeführt, bestimmt, welcher der beiden Decoder die Speicherbank-Auswahl vornimmt. Weil die aktiv-LOW Ausgänge der Decoder als verdrahtete OR-Glieder (wired OR) geschaltet sind und der nicht angewählte Decoder HIGH (inaktive) Ausgänge für alle hat, kann nur der gewählte Decoder ein aktiv-LOW Signal über die Bank-Wahl-Anschlüsse senden.

Signal über die Bank-Wahl-Anschlüsse senden.  
Wenn REMAP logisch LOW (inaktiv) ist, wird der Decoder, der in Bild 4 oben gezeigt ist, eingeschaltet; er gibt Schaltsignale aus, die mit der Original Speicheraufteilung des Modell III übereinstimmen. Ist REMAP logisch HIGH (aktiv), wird der untere Decoder eingeschaltet; er gibt Schaltsignale aus, die das ROM mit Tastatur und Bildschirm in die obersten 16K des Speichers verlegen. RAM-Bank 3 wird in die untersten 16K des Speichers gelegt.

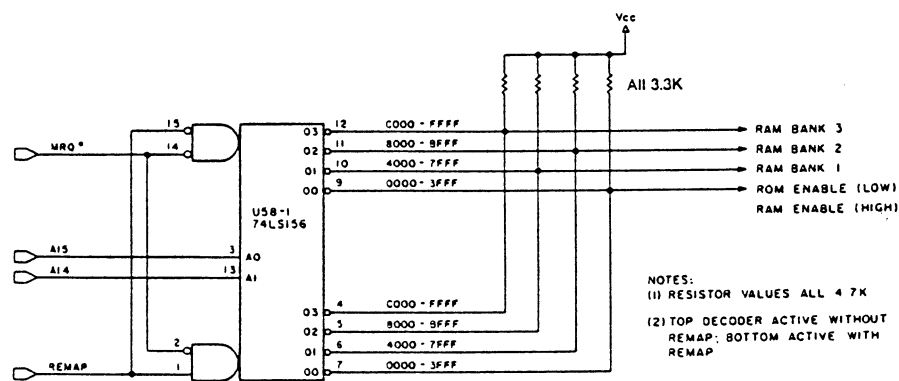


Fig. 3. Bank Decoder with External Request Line

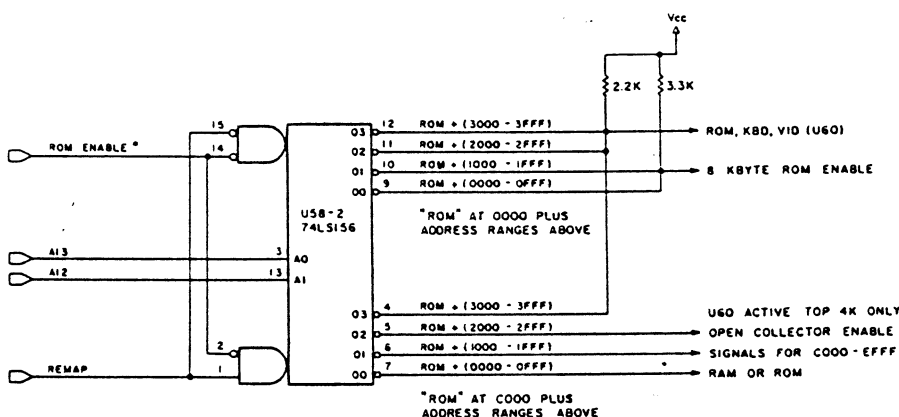


Fig. 4. Decoder for Bank Containing ROM, Keyboard, and Video

## Treffen der Roboter

Chicago – Eine Baufirma schrieb einer Fabrik für elektronische Geräte:

»Unser Elektronengehirn hat ausgerechnet, daß sich die Kosten des von Ihnen geplanten Umbaus auf 125 000 Dollar belaufen.«

Die Firma erhielt umgehend die Antwort ihres Kunden:

»Unser Elektronengehirn hält Ihren Kostenanschlag für viel zu hoch. Wir schlagen daher sobald wie möglich ein Treffen der beiden Roboter in unserem Büro vor.«

## Der zusätzliche Decoder

Ein zusätzlicher Decoder muß außerhalb von U58-1 benutzt werden, um die 8K Segment-Wahl Funktionen zu übernehmen, die vorher der untere Teil von U58 durchführte. Wie in Bild 4 gezeigt, besorgt der obere Decoder in einem zweiten 74LS156 Chip (genannt U58-2) die 8K Schaltungen für die unter Normalaufteilung zum ROM gehörenden 16K Speicher-Segmente. Der andere Decoder versorgt vier 4K Schalter im Adressbereich C000-DFFF unter der Neuauflistung des Speichers.

Die Segment-Decoder werden folgendermaßen durch REMAP angesprochen. Ist REMAP LOW, wird das ROM mit Tastatur und Bildschirm wie im Original-Decoder geschaltet. Ist REMAP HIGH, produziert der untere Decoder vier Einschaltsignale in Übereinstimmung mit Adressleitung A12. Eine dieser beiden Möglichkeiten ist immer aktiv, wenn der Prozessor C000-FFFF adressiert.

Der Draht, der "ROM + (3000-3FFF)" genannt wurde, schaltet in aktivem Zustand U60 ein, um das 4K Segment anzuwählen, welches das 2K ROM, die Tastaturmatrix und den Bildschirmspeicher enthält. Die drei zusätzlichen Signale sind Ersatz, und sie sind für die drei anderen 4K Segmente in C000-FFFF aktiv. Sie können zusammen benutzt werden, um 12K RAM außer der Reihe einzuschalten (womit wir insgesamt 60K hätten). Die Original 8K- und 4K-ROMs, die nutzlos sind, wenn sie von 0000 weggeschaltet sind, tauchen nicht mehr in der Speicheraufteilung auf. Dieser Raum ist frei für eine Speichererweiterung.

Der Ersatz von U58 durch die beiden 74LS156 Demultiplexer-Chips versorgt das Modell III mit seiner Original-Speicher-aufteilung und einer von CP/M nutzbaren Speicheraufteilung (siehe Bild 5). Außerdem erzeugen die neuen Decoder keine längere Verzögerung als die Decoder, die sie in dem jeweiligen Speicherzustand ersetzen. Statt Unzuverlässigkeit im Speicherzugriff bewirken sie sogar eine Speichererweiterung.

## Das Signal, das den Speicher umorganisiert (REMAP)

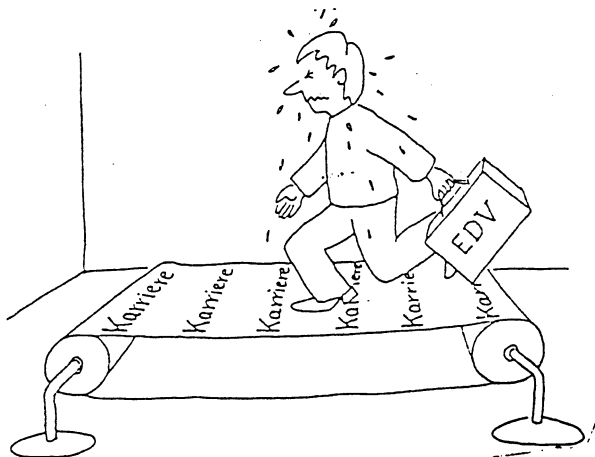
Das REMAP-Signal, das zwischen den beiden Speicher Konfigurationen umschaltet, könnte von einem Schalter außerhalb des Modell III-Gehäuses ausgehen. Aber der Schalter kann nur betätigt werden, wenn der Z80 ein Programm ausführt, das in den Speicher 4000-BFFF geladen wurde (oder im ROM ist). Sonst "vergißt" der Programmzähler, welche Instruktionen er gerade ausführte.

Diese Einschränkung erfordert also, daß Sie ein REMAP nur dann ausführen, wenn die Kontrolle bei einem Programm liegt, das in den richtigen Bereich geladen wurde.

Das Signal kann von einem unbenutzten Ausgabe-Port des Modell III Logic-Boards versorgt werden. Der Ausgabebaustein U98, normalerweise benutzt, um Kassettenrekorder-Levels und ein Video Zeichengröße Signal zu senden, beinhaltet ein unbenutztes D-Flip-Flop.

Indem der unbenutzte Eingang dieses Bausteins mit Bit 0 des Data-Bus verbunden wird, kann die Ausgabe des freien D-Flip-Flop durch das niederwertigste Bit des Port EC kontrolliert werden. Ist dieses Bit LOW (inaktiv), geht der Ausgang des Flip-Flop (REMAPX) auf LOW, und die Speicher-aufteilung des Modell III ist für TRSDOS richtig. Ist dieses Bit HIGH (aktiv), wird der Remap-Befehl ausgegeben, welcher den Speicher für CP/M umorganisiert.

Das RESETX Signal von U98 ist normalerweise an Vcc (+ 5 Volt) gebunden. Die Information der beinhalteten Flip-Flops wird nicht unbedingt gelöscht, wenn Sie den Computer einschalten; auf jeden Fall wird es nie gelöscht, wenn Sie Reset von der Tastatur aus eingeben. Um sicher zu gehen, daß der Computer mit dem ROM an 0000 (Startadresse des Prozessors) startet, muß das RESETX Signal zu U98 von der Unterstützungseinheit des Prozessors kommen und zwar genau von U57 Pin 13.



## Physische Modifikationen

Die Decoder im Modell III zu ersetzen ist weitaus komplizierter als das Aufstecken einer Huckepack-Platine in den 280-Sockel, obwohl an der CPU keine größeren Änderungen vorgenommen werden. Der neue Decoder-Schaltkreis wird auf einer separaten Platine platziert, welche in einen Sockel gesteckt wird, der wiederum den Platz des ursprünglichen Decoders (U58) auf der Hauptplatine einnimmt (Fotos 1a und 1b).

Teile des Decoders, bestehend aus IC58-1, IC58-2, zwei 16-Pin-Sockeln, verschiedenen Widerständen und einem DIP-Stecker (der in den oben erwähnten, U58 ersetzenden DIP-Sockel gesteckt wird) werden auf einer passenden Lochraster-Platine aufgebaut. Die Sockel für die Chips sind auf der Komponenten-Seite der Platine eingesteckt und die Verdrahtung wird auf der Lötseite mit ummanteltem Draht (Wrap-Draht) vorgenommen. Der DIP-Stecker ist auf die Platinenbahnen auf der Lötseite der Platine gelötet. Die Fotos 2a und 2b zeigen die Komponenten- und die Lötseite der Platine.

Die schwierigste Änderung ist das Entfernen des Original-Decoder-Chips (U58) und die Installation eines DIP-Sockels. Das Abschneiden jedes Drahtes des Chips und das Entfernen je eines Drahtes mit einem nicht zu stark erhitzten Lötkolben ist am wenigsten riskant. Ein Sockel, der den Decoder-Board-Stecker aufnehmen kann, wird dann anstelle des IC58 eingelötet. Dies kann ein konventioneller IC-Sockel sein, doch ein aus einzelnen Pins (IC-Kontakte Meterware) individuell aufgebauter Sockel ist erheblich sicherer und besser.

Die Reset-Leitung des Ausgabe-Latch (U98) muß auf dem Logic-Board gekappt werden. Mit isoliertem Draht, der direkt an die IC-Pins gelötet wird, werden folgende Verbindungen hergestellt:

-Data-Bus Leitung 0 von U100, Pin 14, zu dem Ersatz Port EC Latch Input bei U98, Pin 4.

-Port EC Latch Output von U100, Pin 5, zu einer Klemme auf der Decoder-Platte, welche REMAP an U58-1 und U58-2 weiterleitet.

-/RESET (RESETX) von U57, Pin 13, zum Port EC Latch Reset bei U98, Pin 1.

-Adressleitung U12 von U60, Pin 13, zur Decoder-Platte.

Ich habe die oben beschriebene Modifikation über hundert Stunden lang getestet und es ist kein Fehler aufgetreten. Die 8K und 4K Modell III ROMs sind ausgeschaltet und drei 4K Speichererweiterung Einschalt-Signale funktionieren wie beschrieben.

VIDEO RAM	FFFF
PRIMARYLY UNUSED	FC00
KEYBOARD SWITCHES	F900
2K MODEL III ROM	F800
	F000
SPACE FOR 12K RAM EXPANSION	
	C000
"MIDDLE" 16K RAM BANK	
	8000
"LOW" 16K RAM BANK	
	2000
"TOP" 16K RAM BANK	
	0000

Fig. 5. CP/M-Compatible Model III Memory Map.

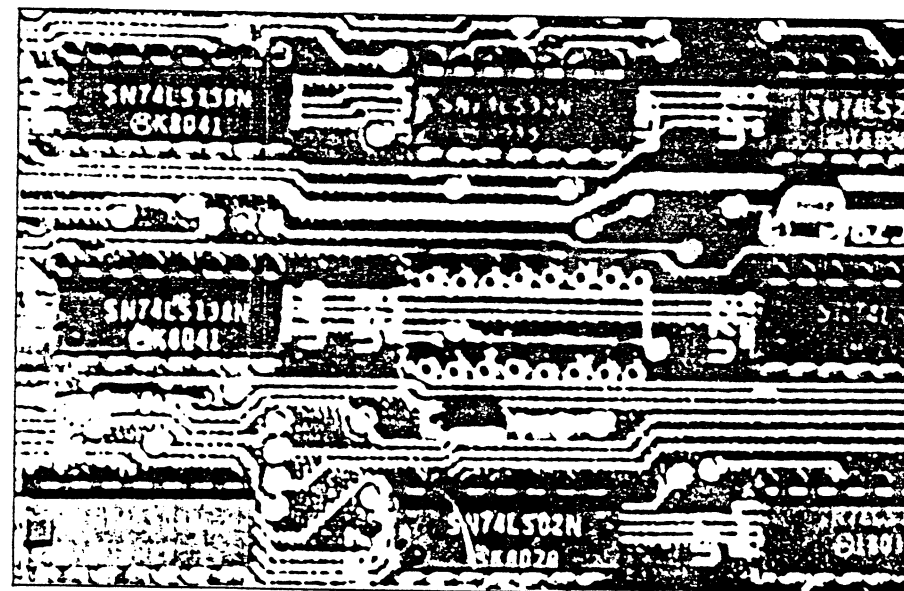


Photo 1a. Socket for New Decoder

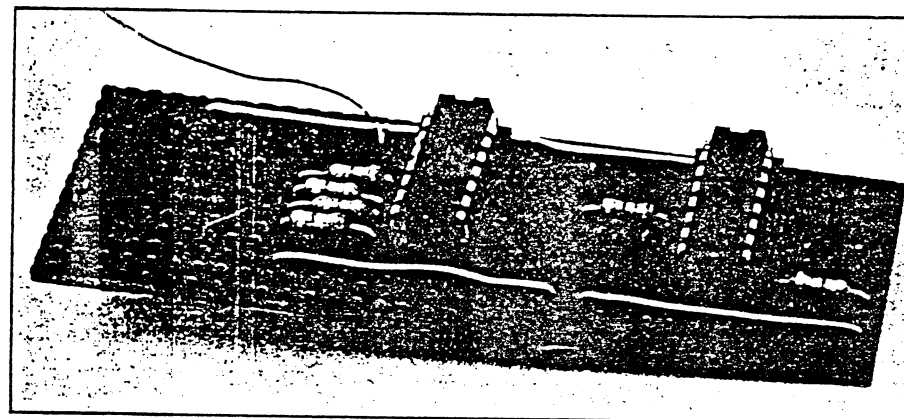


Photo 2a. Component Side of Decoder Board

Das Assemblerlisting für den Bootstrap Loader ist als "Paperware-Kopie" von Hartmut Obermann erhältlich

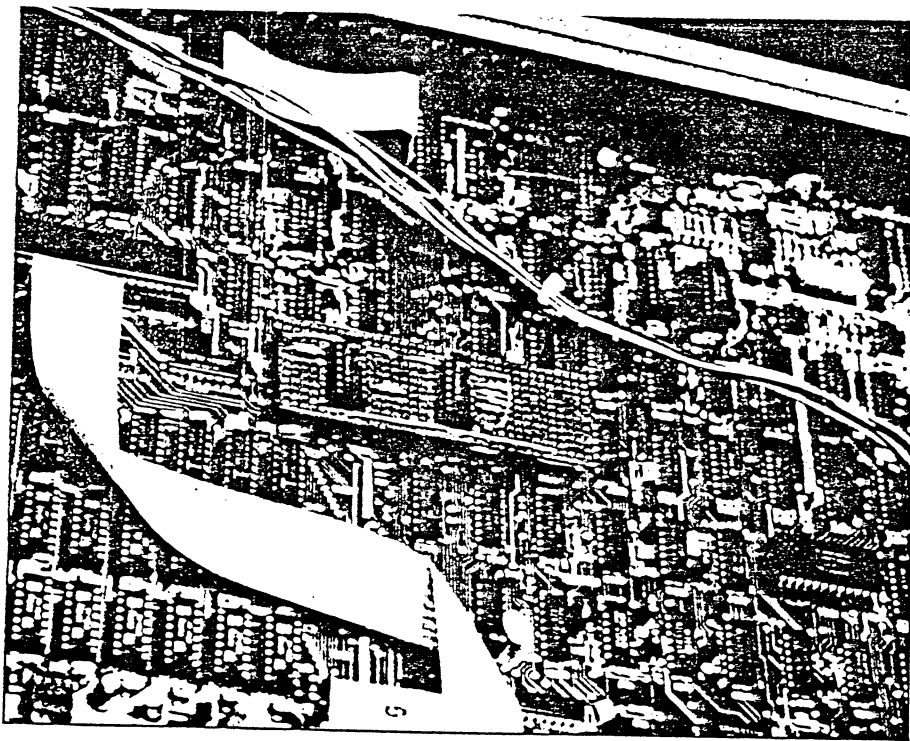


Photo 1b. New Decoder Installed

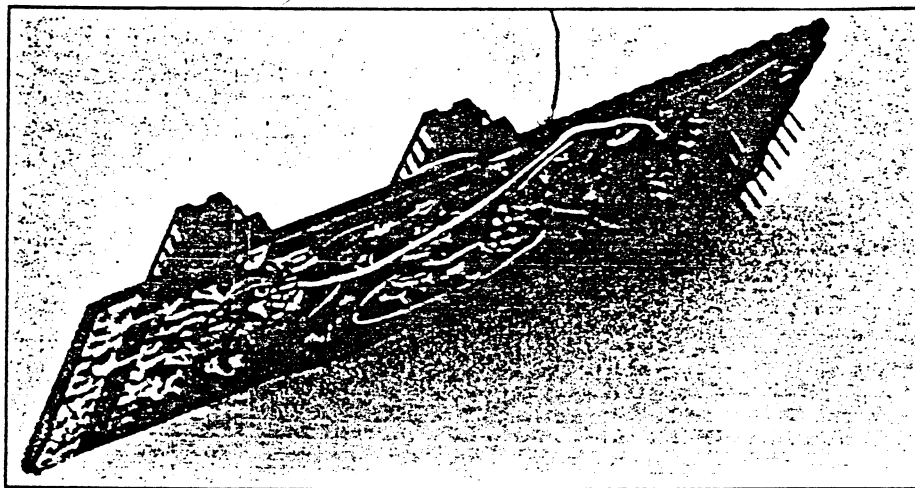


Photo 2b. Solder Side of Controller Board

## Vorläufiges Design zur Speichererweiterung

Es wurden keine Tests mit einem Modell III durchgeführt, das zusätzlichen Speicher hatte. Die Speichererweiterung ist eine kostenlose Nebenerscheinung der Wahl unserer Chips für die Speicherneuaufteilungsmodifikation.

Speichererweiterungen können in derselben Technik ausgeführt werden wie die beschriebene Modifikation zur Neuaufteilung des Speichers. Stecken Sie eine Platine, die aus 8 DIP-Steckern und 16 Sockeln konstruiert ist, in die Sockel für eine der 16K RAM Banken. Verbinden Sie 8 der Sockel parallel mit entsprechenden Headern und stecken Sie die 4116 RAM Chips in sie. Sie funktionieren genau wie bei einem nagelneuen Computer.

Verbinden Sie alle Pins bis auf das CAS\*-Signal (CAS\* - Column Address Strobe pin) der acht übrigen Sockel (für die acht zusätzlichen 4116s) parallel mit entsprechenden Steckern. Eine CAS\*-Leitung, die mit allen CAS\* Pins der neuen RAM Chips verbunden ist, kann hergestellt werden, indem die drei 4K Einschaltleitungen des U58-2 als wired OR verdrahtet werden. Um das neue RAM Bank-Wahlsignal CAS\* zu erhalten, wird das daraus resultierende Signal mit dem Modell III CAS\* über ein OR-Glied verbunden (siehe Bild 6).

Nur weiteres Konstruieren und Austesten kann sicherstellen, daß diese Speichererweiterungsmodifikation über längere Zeiträume brauchbar ist. Auf jeden Fall sollte die Stromversorgung, die mit einer 50-Hertz Leitung arbeiten kann, die neuen RAMs mitversorgen können. Außerdem dürfte in diesem gut durchlüfteten Teil des Gehäuses kein Hitzestau auftreten.

Äußerste Vorsicht ist die einzige Garantie für einen Erfolg bei der Arbeit mit der CPU-Platine. Alle Verbindungskabel und Verbindungen, die in der Platine stecken, sind sehr dünn und sehr zerbrechlich.

Versuchen Sie sich nicht an dieser Modifikation, wenn Sie ein Neuling im Arbeiten mit Lötzinn sind.

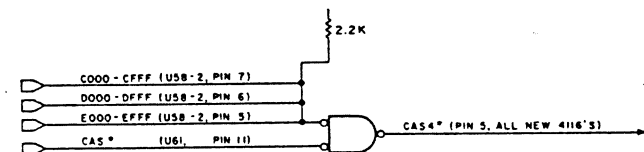


Fig. 6. Preliminary Logic for a New RAM Bank-Select Signal

## Software Kontrolle

Die Software, die die Speicherumstellung schaltet, ist einfach. Um den Speicher zu re-organisieren, schreiben Sie die folgenden Instruktionen in Ihr Bootstrap Loader Programm:

```
LD  A,9
OUT (0ECH),A
```

Dementsprechend benutzen Sie folgende Zeilen, um den Originalzustand Ihres Computers wiederherzustellen:

```
LD  A,8
OUT (0ECH),A
```

Das am wenigsten signifikante Bit des Ausgabe-Bytes schaltet den Speicher um. Andere Bits in diesem Byte betreffen Video Zeilenlänge und Zeichensatz, Video Warte-Status, Kassettenmotor Schaltung und externe I/O Bus Operationen. Sind diese Bits aus (off), ist das Modell III in folgendem Zustand:

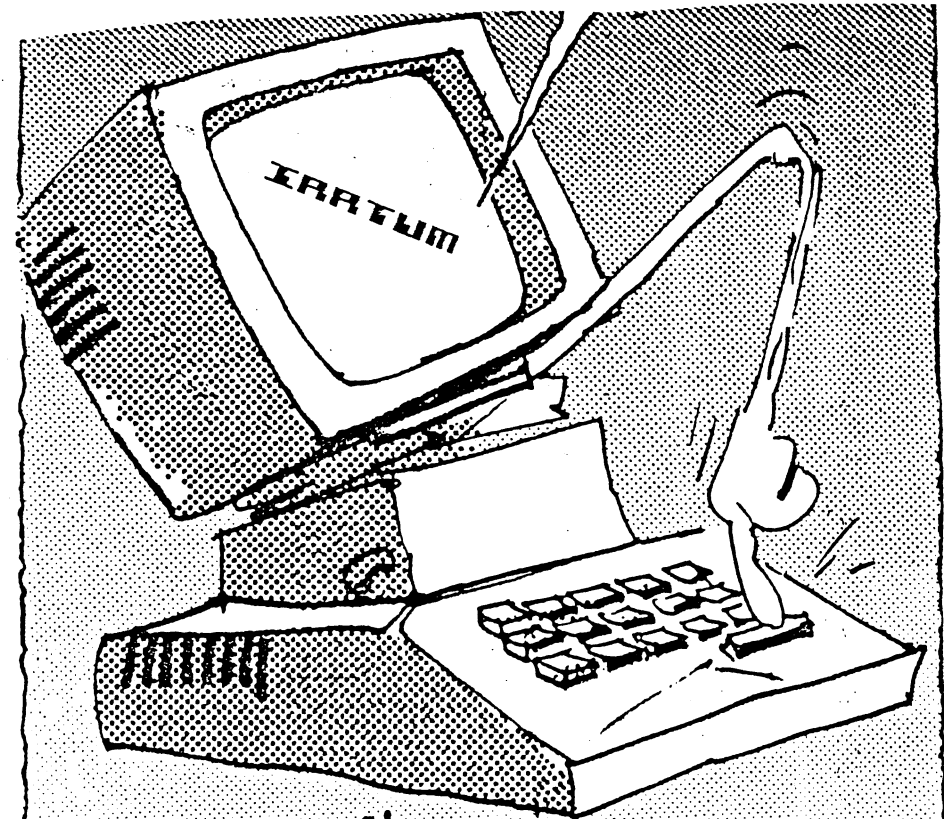
64 Zeichen/Zeile, Japanischen kana Spezialzeichensatz, kein Video Warte Status und Kassettenmotor sowie I/O Bus sind ausgeschaltet. Mit Bit 3 an wird der griechische Zeichensatz ausgewählt.

Die meisten mit kommerziellen Modifikationen gebrauchten CP/M-Versionen können leicht an diese Modifikation angepaßt werden. Durchsuchen Sie nur den Bootstrap Loader und das Basic Input/Output System (BIOS, das maschinenabhängige Modul in CP/M) nach den Speicherumstellungs-Routinen, die für die jeweilige Modifikation gelten und ersetzen Sie diese durch die oben angeführten.

Als Alternative können Sie ein neues BOOT (s. unten) und BIOS für die Standard CP/M-Version schreiben. Der unten gelistete Bootstraploader kann in ein vorhandenen BIOS eingebaut werden.

Ein Bootstrap Loader für CP/M oder jedes andere Programm, das Sie auf der Diskette plaziert haben, ist im Listing gezeigt. Der Loader muß auf Track 0, Sektor 1 einer Diskette im Modell III Standard Format von 18 256-Byte Sektoren per Track stehen. Das zu ladende Programm muß auf Track 0, Sektor 2 beginnen.

Die Speicheradressen, in die das Programm geladen wird, sind immer am Anfang des Programms angegeben und dort können Sie diese leicht verändern, um das Programm anzupassen. Wenn Ihre Diskette eingelegt ist, bringt das Modell III ROM bei einem RESET den Loader in das RAM bei 4300 hexadezimal und übergibt ihm die Kontrolle. Er lädt dann Ihr Programm und dieses Programm gibt Ihnen Möglichkeiten, die Sie vorher nicht hatten.



sagte  
der  
Computer  
und  
knipste  
sich aus.

Bild 1: Modell III Speicher Aufbau

Bild 2: Modell III Adressen Decoder Schaltkreis

Tabelle 1: 16K Bank-Wahl beim USB

Tabelle 2: Segment-Wahl in ROM-Bank

Bild 3: Bank Decoder mit externer Wahlleitung

Bild 4: Decoder für Bank mit ROM, Tastatur und Bildschirm

Bild 5: CP/M-kompatibler Modell III Speicher Aufbau

Bild 6: Vorläufige Logik für ein neues RAM Bank-Wahl Signal

Foto 1a: Sockel für den neuen Decoder

Foto 1b: Neuer Decoder installiert

Foto 2a: Komponenten Seite des Decoder Boards

Foto 2b: Lötseite des Controller (Decoder) Boards



## Impressum

1. Vorsitzender: **Hartmut Obermann** Tel.: 07 21/ 85 40 68  
Wilhelm-Baur-Straße 8 BTX: 07 21/  
FAX: 07 21/ 85 40 68

**76135 Karlsruhe**

2. Vorsitzender: **Gerald Schröder** Tel.: 0 41 05/ 26 02  
Am Schützenplatz 14

**21218 Seevetal**

Hardwarekoordinator: **Andreas Magnus** Tel.: 02 09/ 87 02 30  
Bismarckstraße 29

**45879 Gelsenkirchen**

NewDOS-Diskothekar: **Oliver Volz** Tel.: 07 11/ 74 40 51  
Am Ochsenwald 37A  
**70565 Stuttgart (Rohrerhöhe)**

CP/M-Diskothekar: **Fritz Chwolka** Tel.: 0 24 64/ 89 20  
Saarstraße 34  
**52457 Aldenhoven**

C-128-Diskothekar: **Günther W. Braun**  
Postfach 80 02 26  
**81602 München**

Clubbücherei: **Kurt Müller** Tel.: 0 41 52/ 7 06 43  
Sophie-Scholl-Ring 3b  
**21502 Geesthacht**

Redaktion: **Jens Neueder** Tel.: 07 91/ 4 28 77  
Gechlachtenbretzingen BTX: 07 91/ 44 47 22  
Rudolf-Then-Straße 32 FAX: 0 79 71/ 2 50 55  
**74544 Michelbach/ Bilz**

Bankverbindung: **Club 80** Postgiroamt Frankfurt  
Postgiro Sonderkonto CLUB 80 BLZ: 500 100 60  
Obermann H., 8870 Günzburg Kto.Nr.: 496 071 - 605

Autoren: Die Redaktion bedankt sich bei den im  
**Inhaltsverzeichnis** genannten Autoren  
für die Mitarbeit an der Club-INFO.  
Eine Zensur oder Kontrolle der INFO-Beiträge  
erfolgt nicht.

## Schluß

Hallo Club 80'er,

wenn schon mal alles schief geht, dann auch beim INFO. Ihr haltet heute INFO 44 und 45 in den Händen, da bei der Produktion des 44. INFO's der Kopierer sich für eine Reparatur (natürlich mit notwendiger Ersatzteilbestellung) entschlossen hatte. Nach erfolgreicher Reparatur war dann auch schon der nächste Info-Termin erreicht und so gibt es heute mal **"Zwei auf Einmal"**.

Da ich wenig Material für dieses INFO hatte, habe ich mich entschlossen, zum Hardwarethema passend, einmal die Hardwarebeiträge der ersten zehn INFO's hier nochmals zu veröffentlichen. Vielleicht sind ein paar neue/alte brauchbare Tips für Euch dabei. Wenn nicht, ist es sicher einmal wieder schön in der Hardwarenostalgie der ersten Jahre zu schwelgen. Bei den heutigen **"Multi-Power-User-Chips"** ist man meist recht schnell nur noch als **Softi** und/oder **Power-User** der vorhandenen Software gefordert.

Nachdem es mit dem September-Redaktionsschluß nicht klappen konnte, wünsch ich mir für den Dezember natürlich rege Aktivität bei den Artikelschreibern, denen ich hierbei mal wieder meinen Dank für die Mitarbeit im Club 80 aussprechen möchte. Denkt bitte alle auch an den **DFÜ-Fragebogen**, der jetzt leider knapp terminiert ist.

Viel Spaß beim Schmökern in der Hardwarevergangenheit.  
Bis zum nächsten INFO grüßt Euch Euer



lfd. Nr.	Nachname	Vorname	Straße	PLZ	Ort	Telefon privat Tel. geschäftl.	Telefax privat FAX geschäftl.	■BTX ■FIDO-Node	■Mailboxname ■Mailboxnummer
1	Barendt	Harry	Hermann-Löns-Straße 7	GER 50181	Breedburg (Erft)	02272/ 7168	-	-	-
2	Berndt-Jochum	Ilse	Stachelsgut 24	GER 51427	Bergisch Gladbach	02204/ 65254	-	-	-
	IBM, GENIE III, GENIE IIIs, SHARP Pocket 160, Scanner					02204/ 65254	-	-	-
3	Bernhardt	Helmut	Preetzer Straße 75	GER 24143	Kiel	-	-	-	-
	IBM, Prof 180, CPU 280, diverse PCs, Novellite-Netz, 1496E, Mustek105+					0431/ 77578-20	0431/ 77578-99	2:242/262.26	-
4	Bielenberg	Georg	Erikaweg 1	GER 24568	Kaltenkirchen	04191/ 3751	-	-	-
	Atari 260ST, C128, Schneider Joyce, Modem: LC2496 Digitech, Scanner: Supersc.III					04193/ 90430	-	-	-
5	Brans	Jörg	Tieloh 55	GER 22307	Hamburg	040/ 6906531	-	-	-
	IBM, CD-Rom, Streamer, Soundkarte					-	-	-	-
6	Braun	Günther W.	Postfach 80 02 26	GER 81602	München	-	-	-	Discovery
	Commodore 128 D, Akustikkoppler					-	-	-	07127/ 70107
7	Braun	Harald	Postfach 8011	GER 24154	Kiel	0431/35139	-	-	-
8	Böckling	Ulrich	Juchaczstraße 61	GER 56203	Höhr-Grenzhausen	02624/ 4861	-	-	-
	IBM 386DX33, 1040ST, VC20, C64, TRS80 M I, ZX81, Modem, Videodat-, Videotextdecoder					02631/ 895168	-	-	-
9	Chwolka	Fritz	Saarstraße 34	GER 52457	Aldenhoven	02464/ 8920	-	-	-
	IBM 386+Co, Commodore, Apple, Z 280, Modem 2.4					-	-	2:248/242:8	-
10	Dose	Volker	Dorfstraße 10	GER 24235	Brodersdorf	04343/ 1357	-	-	-
	GENIE IIIs mit Z180, EPROMer, Modem					-	-	-	-
11	Halgasch	Gert	Großschönauer Straße 26	GER 02796	Jonsdorf	035844/ 636	-	-	-
	IBM 386DX					-	-	-	-
12	Hartmann	Hans-Günther	Möwenstraße 9	GER 27804	Berne	04406/ 6911	04406/ 1071	-	-
	IBM 386SX, TANDY M4p, Z280-Kartenrechner, Real Time Clock, Speed Up Kit 6,3MHz, 40MB-P					0421/ 248-2419	-	2:240/300.24	-
13	Hebecker	Ulrich	Büsnauer Straße 15	GER 70563	Stuttgart	0711/ 734800	-	-	-
	IBM 286+386, 128D, 1581, Kaypro484					-	-	-	-
14	Held	Manfred	Stirner Straße 22	GER 91785	Pleinfeld	09144/ 6563	09144/ 8514	-	-
	IBM, Modem ZyXEL 1496+, CD-ROM					0911/ 219-2245	-	2:2400/10.10	-
15	Hermann	Klaus	Forchenstraße 8	GER 72124	Pliezhausen	07127/ 71945	-	-	Discovery
	IBM 386, ET 4000, Modem					-	-	2:2407/70.740	07127/70107
16	Hürdler	Manfred	Niederhoferstraße 29	GER 97222	Rimpar	09365/ 4235	-	-	-
	Victor Sirius 1 (IBM), CPC 6128					-	-	-	-
17	Johnen	Willi	Hansemannstraße 1	GER 52351	Düren	02421/ 501305	-	-	-
	GENIE IIIs					02421/ 33064	-	-	-
18	Kalb	Michael	Königstraße 10	GER 96242	Sonnefeld	09562/ 6387	-	-	-
	Schneider CPC, Joyce, Modem, Drucker					-	-	-	-
19	Kauka	Dietmar	Straße des Friedens 37	GER 04552	Neukirchen (Borna/Leipz)	03433/851019	-	-	-
20	Kemmer	Jürgen	Dorfberg 7	GER 97232	Sulzdorf	09334/ 1050	-	-	-
	IBM 386, UltraSound, ZyXEL U1496E, DCF-77-Empfänger parallel, IR-Sender seriell					-	-	2:247/2086.1	-
21	Kleespies	Andreas	Mainzer-Land-Straße 765	GER 65934	Frankfurt /Main	069/ 387432	-	-	-
22	Kuhn	Eckehard	Im Dorf 14	GER 72636	Frickenhausen	07022/ 45417	-	-	-
	Atari ST 1040, TRS80 M I					-	-	-	-
23	Linder	Jörg	Küstriner Str. 68	GER 15306	Seelow	03346/ 520	-	-	-
	KC 85/4 mit Floppy					-	-	-	-
24	Lorenz	Walter	Mahrackerstraße 9	GER 60431	Frankfurt /Main	069/ 531656	-	-	-
	IBM 286/486, Z80-, HD 64180-Eigebau, Z280-T, Scanner, Soundkarte, ET 4000					-	-	-	-
25	Magnus	Andreas	Bismarckstraße 29	GER 45879	Gelsenkirchen	0209/ 144029	-	-	-
	IBM 386Dx, GENIE IIIs, Modem 2400					-	-	-	-

1fd.

Nr.	Nachname	Vorname	Straße	PLZ	Ort	Telefon privat Tel. geschäftl.	Telefax privat FAX geschäftl.	■BTX ■FIDO-Node	■Mailboxname ■Mailboxnummer
26	Mahlert	Herbert	Hohenbudbergerstraße 112 A	GER 47229	Duisburg	02065/ 47217	-	-	-
	IBM, GENIE I ■ c't Videotext-Karte, Vobis Videodat-Decoder					02065/ 902592	-	-	-
27	Messerschmidt	Kurt	Hanns-Eisler-Straße 54	GER 10409	Berlin	-	-	-	-
28	Müller	Kurt	Sophie-Scholl-Ring 38	GER 21502	Geesthacht	04152/ 70643	-	-	-
	IBM, Atari Mega ST4 ■ ATONCE 386, OverScan, Scanner, HBS640-T36, Modem					040/ 89983403	-	-	-
29	Mössel	Franz	Schafferstraße 12	I 39012	Meran	0039-473/34178	-	-	-
	IBM, Joyce, Workmate-Bullet ■ Modem					0039-471/980496	-	2:333/400	-
30	Neueder	Jens	Rudolf-Then-Straße 32	GER 74544	Michelbach / Bilz	0791/ 42877	-	791444722	CCWN
	IBM 286/386/486, Atari ST 1040, TRS80 MI ■ Modem, Sound Galaxi NX, Scanner, Tape, Borsu					07971/ 250-50	07971/ 250-55	1001667	0715168434
31	Neumann	Christof	Zeitblomstraße 22/2	GER 89077	Ulm /Donau	0731/ 6022568	-	-	-
	IBM, Tandy MII, Tandy M4p ■ Novell					0731/ 9749720	-	-	-
32	Nitschke	Stefan	Germenenstraße 5	GER 75045	Walzbachtal 1	07203 /452	-	-	-
	CPU 280 ■ NEC 7220 1024*1024 Grafikkarte					-	-	-	-
33	Obermann	Hartmut	Wilhelm-Baur-Straße 8	GER 76135	Karlsruhe	0721/ 854068	-	-	-
	IBM486, Tandy M4p, Epson PX-8 ■ Modem, Scanner					-	-	2:241/7922.10	SYNREL
34	Peters	Jürgen	Heukoppel 14	GER 22179	Hamburg	040/ 6412371	-	-	08282/ 4311
	IBM, Z280, MZ821, SHARP ■ Modem, Soundkarte, Video, Drucker					-	-	-	-
35	Retzlaff	Bernd	Kleiner Sand 98	GER 25436	Uetersen	04122/ 43551	-	-	-
	IBM 386, C64, GENIE I ■					04103/ 605310	-	-	-
36	Rinio	Gerd	Rennbahnstraße 9	GER 22111	Hamburg	040/6552630	-	-	-
	IBM 486DX66/2, RTS 80, TRS 80 RS ■ Modem					-	-	-	-
37	Ruschinski	Claus	Pommernstraße 21	GER 45770	Marl	02365/ 34646	-	-	-
	IBM 386, TRS80 M I ■ Highscreen-Scanner, CoProz IIT387					-	-	-	-
38	Schimmer	Jörg	Stettinerstraße 28	GER 60388	Frankfurt	06109/ 35336	-	-	-
	IBM 486, Schneider CPC ■ Modem CSR2400					069/ 3800-2385	-	2:249/70.9	-
39	Schmid	Alexander	Entmannsdorf 5	GER 96317	Kronach /Gehülz	09261/ 53496	-	-	-
	GENIE IIIs, GENIE IIIs, CPU 280 ■ Modem 2400, Prommer80, Ramdisk, Club80Terminal					-	-	2:2400/830	-
40	Schmitz	Rainer	Küferweg 12/1	GER 73099	Adelberg	07166/ 1397	-	-	-
	IBM, Portfolio, NCR Decision Mate V, Joyce, ... ■ Modem 1200, Märklin Digital Interface 6050					07161/ 608-475	-	-	Discovery
41	Schoberth	Uwe	Petrus-Waldus-Straße 14	GER 75443	Oetisheim	07041/ 7254	-	-	07127/ 70107
	Alphatronic P3 ■					0711/ 89394500	0711/ 89394513	-	Discovery
42	Scholz	Hans-Werner	Spitalstraße 54	GER 41334	Nettetal	02157/ 3613	-	-	07127/ 70107
	IBM 386, Prof 80, ITT 3030 ■ Prommer80					-	-	-	-
43	Schroers	Horst-Dieter	Breslauer Straße 9	GER 85622	Feldkirchen	089/ 9032615	089/ 9043413	-	-
	IBM ■ Modem, Scanner					-	-	-	-
44	Schröder	Gerald	Arminiusstraße 2	GER 22525	Hamburg	040/ 8507131	-	-	-
	IBM 386SX, Atari 1040STF, Z280 ■					040/ 54715334	-	-	-
45	Schröder	Egbert	Joachimstraße 18	GER 46284	Dorsten	02362/ 75311	-	-	-
	Portfolio, TRS80 MI, GENIE I, GENIE IIIs ■					02362/ 49-9649	-	-	-
46	Schulte	Hartmut	Entenschnabel 8	GER 31311	Uetze	05173/ 1248	05173/ 24631	-	-
	IBM, Z280 u.a. ■ Scanner					-	-	-	-
47	Sonnemann	Harald	In den Eckwiesen 9	GER 64405	Fischbachtal	06166/ 8512	-	-	-
	NDR-Klein ■ parallel/seriell, EPROMer					06151/ 92-1265	-	-	-
48	Staimer	Ulrich	Dekan-Mayer-Straße 15	GER 86199	Augsburg	0821/ 993104	-	-	-
	IBM (Siemens), C128, Reh Z280 ■ Video, Panasonic-Drucker					0821/ 2574-107	-	-	-
49	Stumpferl	Stefan	Hasenbergstraße 57	GER 80933	München	089/ 3138193	//0893144001//	-	-
	Amstrad CPC 6128+, GENIE IIIs ■ Modem, 20MB-Wechselplatte, PSG&PIO, SIO, ...					-	-	-	-
50	Sörensen	Rüdiger	Wiesbadener Str. 28B	GER 55252	Mainz-Kastel	06134/65342	-	-	-
						-	-	-	-

1fd. Nr.	Nachname	Vorname	Straße	PLZ	Ort	Telefon privat Tel. geschäftl.	Telefax privat FAX geschäftl.	BTX FIDO-Node	Mailboxname Mailboxnummer
51	Tornow	Wilhelm	Elbblick 46	GER 21629	Neu Wulmstorf	040/ 7007280	040/ 7003854	-	-
	IBM 386DX, Atari Mega ST4, Tandy M4p, Streamer, Soundblaster, Modem 2400, F.A.K.S. 91					-	-	-	-
52	Vogl	Michael	Postfach 10 06 10	GER 41490	Grevenbroich /Laach	02181/45112	-	0218145112-0001	-
	IBM, Amstard CPC 464/6128, Modem Digitek 2400, Scanner Dart F.CPC					-	-	2:2433/27.2	-
53	Volkmer	Richart	Am Spörkel 69	GER 44227	Dortmund	0231/ 752574	-	-	-
	IBM XT, Apple LC, CPC 6128, Osborne, TandyMI, Modem 2400, ScanMan-Scanner					-	-	-	-
54	Volz	Oliver	Am Ochsenwald 37A	GER 70565	Stuttgart (Rohrerhöhe)	0711/ 744051	-	-	-
	IBM, GENIE IIs, Modem Avantec					0711/ 685-3013	-	-	-
55	Walter	Hans-Jürgen	Elssholzstraße 9	GER 10781	Berlin	-	-	-	-
						-	-	-	-
56	Werner	Heiko	Reichenberger Straße 5	GER 01129	Dresden	0351/ 4608612	-	-	-
	IBM 286/486, Modem 2400, Scanner					-	-	-	-