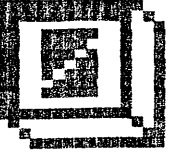
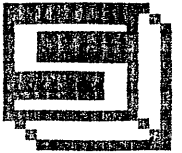
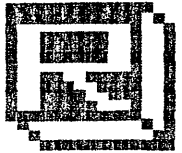
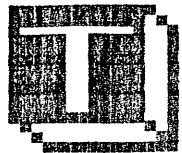




USER
CLUE

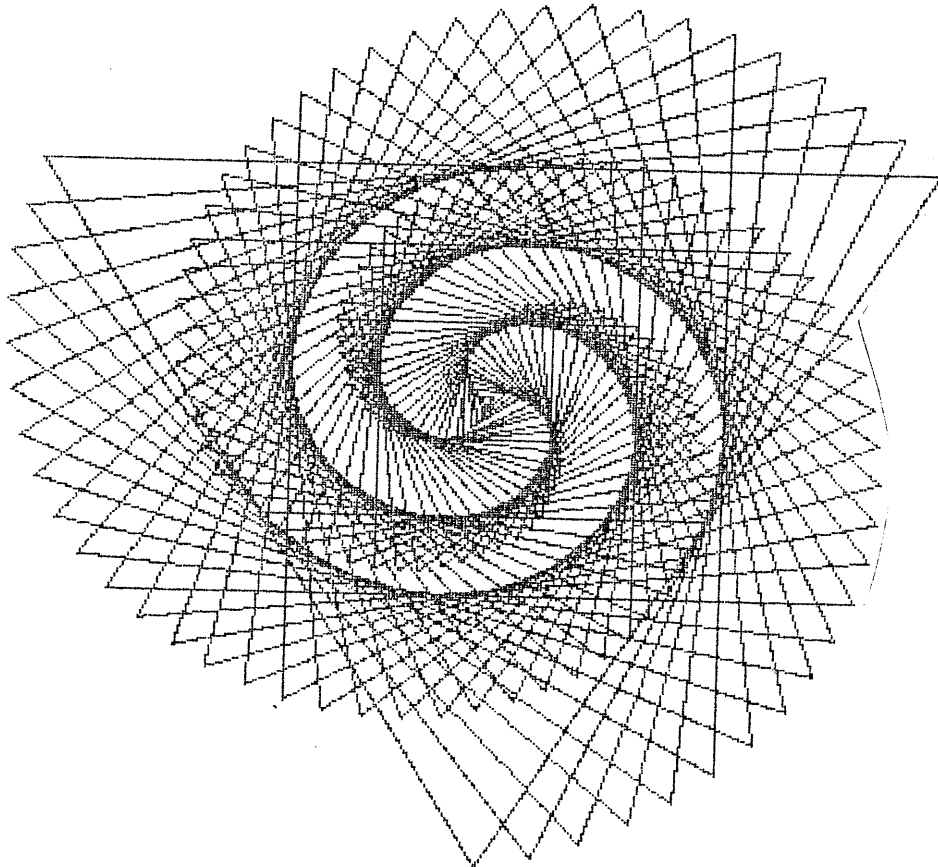
und Colourserie

USER
CLUE



E R E M E R H A F E N

CLUE-INFO FO
CLUE-INFO FO
CLUE-INFO FO



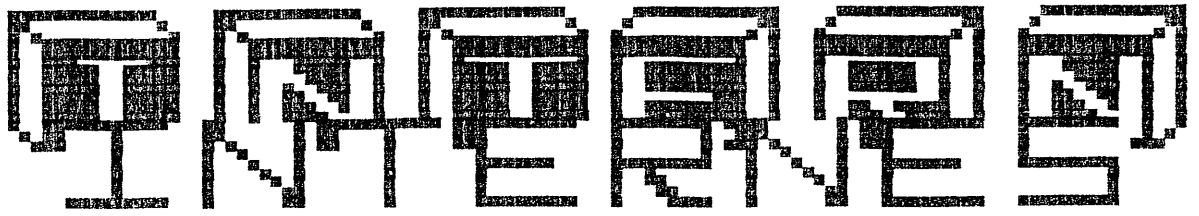
2. JAHRGANG | 4. AUSGABE

Betr.: Peter Spieß, Trugenhofenerstr. 27, 8859 Rennertshofen 1

INHALT

Seite	Thema
1	Internes vom Betreuer
2-4	Ein Brief
5-8	Die Schraube soll gleich rotieren v. A. Sopp
9	Computeritis
10-13	PUT TO - Ein neuer BASIC-Befehl v. A. Sopp
14-18	Uhr in der Interrupt-Routine v. K. v. d. Meulen
19	Nützliche PEEK's und POKE's für's Genie v. M. Jermann
20-22	Joystickinterface für's Genie v. M. Jermann
23	Adressenliste
24	Flohmarkt

Internes
vom
Betreuer



INTERNES VOM BETREUER

Auf Grund einiger Anfragen, was mit dem Mitgliedsbeitrag geschieht, gebe ich Euch nachstehend eine kurze Aufstellung über den Verwendungszweck des Geldes.

monatlicher Beitrag pro Person : 3, -- DM
abzgl. Porto für Info : -,70 DM
abzgl. Druckkosten (-,15 DM pro
Seite) bei 20 Seiten (oft
sogar noch mehr Seiten) : 3, -- DM
abzgl. Versandtasche : -,25 DM

Ergebnis: minus -,95 DM

Ihr seht, ohne private Zuschüsse von mir oder Spenden von Euch ist es fast unmöglich, eine monatliche Clubzeitung auf die Beine zu stellen.

Anschließend ein Auszug aus einem Brief an mich:

"Ich vermisste im Club eine Hilfe für die Anfänger in Sachen Hardware. Könnte man vielleicht dieses Thema ansprechen? Ist es evtl. möglich, daß jemand einen Artikel in dieser Richtung schreibt und die Ausbau- und Erweiterungsmöglichkeiten gegenüberstellt? Ich möchte beim Ausbau nicht den Fehler begehen, mir die verschiedenen Arbeitsbereiche (z. B. CP/M) zu verbauen oder zu erschweren. Einiges kann man vielleicht von vornherein bei einer Erweiterung berücksichtigen. Ein allgemeiner Artikel zu diesem Thema, dürfte für die meisten der neuen Mitglieder von Interesse sein."

Ich bitte also alle Mitglieder um die Mithilfe bei der Lösung des o. a. Problems. Wie immer suche ich natürlich auch weiterhin andere interessante Artikel, Programme, Tips, Tricks, etc.

Zum Schluß: Den Leserbrief von mir in der letzten Computer Persönlich, habe ich ca. im November 1983 an die Redaktion abgeschickt. Geschwindigkeit ist alles!!!

Wir gratulieren zum Geburtstag:

Ralf Folkerts

Hans-Christian Meier

Martin Rüttgers

herzlichen Glückwunsch!

Neue Mitglieder:

Klaus van der Meulen

Othmar Stark, Österreich

Thomas Althaus

H. Omasreiter

Ralf M. Hübber

Verlag für Computertechnik

Ralf M. Hübber - Verlag für Computertechnik - D-5429 Marienfels

Herrn
Peter Spieß
Trugenhofenerstraße 27
8859 Rennertshofen 1

Ralf M. Hübber
Verlag für Computertechnik
Mühlbachstraße 2
D-5429 MARIENFELS
Telefon 067 72 - 12 61 und 78 28

Bank:
Nassauische Sparkasse Miehlen
(BLZ 510 500 15) Kto.-Nr. 707 009 586
Postscheck:
Ludwigshafen, Kto.-Nr. 723 16-678

Datum: 14.2.1984

Ihrem kurzen Anschreiben und dem Brief von Herrn Klaus Schmidt in Ihrem Info, Jahrg. 2, Ausg. 2, entnehme ich, daß Sie in Ihrem Club nicht nur eigene Programme tauschen, sondern daß Sie auch Programme von gewerblichen Anbietern Ihren Mitgliedern zugänglich machen.

Ich möchte nun in diesem Brief nicht nur von mir (ich bin ja gewerblicher Anbieter!) sondern auch von allen anderen Software-Anbietern sprechen.

Es wäre natürlich schön, wenn Sie diesen Brief auch den Mitgliedern Ihres Clubs zugänglich machen würden.

Also, ich möchte Ihnen hier einmal von den Sorgen der Software-Anbieter berichten, die wir mit Clubs haben, die unsere Programme tauschen:

Entwicklungen bei uns sind selten, da der Arbeitsanfall zur Entwicklung eines erstklassigen Programmes recht hoch ist. Wir kaufen also in den meisten Fällen, für recht hohe Preise, Programme mit Copyrights an. Das ist der erste Kostenfaktor für uns.

Bevor nun ein solches Programm verkauft wird, entstehen aber noch sehr viele weitere Kosten, die wir zu tragen haben. Als nächstes kommt nämlich die Gestaltung eines Labels für die Cassette oder die Diskette. Weiter geht es mit Werbekosten, zum einen sind das Inserate in Fachzeitschriften, zum anderen Rundschreiben an Händler und die Erstellung von Katalogen. Weiterhin möchte jeder Anbieter seinem Programm auch eine einwandfreie Bedienungs-Anleitung zulegen. Diese muß gesetzt und gedruckt werden!

Ist man schon einmal soweit mit einem Programm gekommen, hat man es fast geschafft. Jetzt entstehen nur noch die Kosten für Datenträger, hierbei kann ich nicht für andere Anbieter sprechen, wir aber benutzen z. B. Cassetten, mit hervorragenden Eigenschaften, die aber nicht billig sind. Jetzt muß man natürlich die Programme noch kopieren. Das wird bei uns mit dem Computer gemacht, um einen hohen Qualitäts-Standard zu halten. Die Recorder werden oft ausgetauscht, weil Sie ja verschleissen, und dadurch die Qualität der Kopien gemindert wird.

Wenn man jetzt soweit ist, daß man eine Anzahl von Kopien fertig hat, werden diese überprüft (Arbeitszeitaufwand).

So, jetzt kann man verkaufen. Der Verkaufspreis wird festgelegt, immer im Bewußtsein, daß von jedem verkauften Programm mindesten eine Kopie gemacht und getauscht wird. Man legt also den Preis direkt höher fest, um das auszugleichen.

Erste Erkenntnis: Programme wären wesentlich billiger, wenn nicht mehr getauscht werden würde!

BITTE WENDEN

GENIE DATA

Die Video Genie-, Colour Genie- und TRS 80 - Anwender Fachzeitschrift.

Wenn ein Programm kopiert und verschickt (Porto-Kosten), hat man es aber noch nicht ausgestanden, den es werden immer wieder Programme reklamiert, die man umtauschen muß.

Wenn man seine Programme an Händler verkauft, geht übrigens vom Preis nochmals ein Händler-Rabatt ab, den der Händler will auch etwas verdienen.

Wie Sie jetzt schon sehen, entsteht, bevor ein Programm verkauft wird, ein riesiger Aufwand an Arbeit und Kosten. Und man verkauft Programme ja nicht, weil man nichts besseres zu tun hat, sondern weil man damit Geld verdienen möchte.

Man verdient übrigens nicht an jedem Programm etwas, denn man kauft Flops, die man nicht mehr los wird, bei denen man also nur Geld zulegt!

Nur an wenigen Programmen (etwa 20%) die man anbietet, wird auch etwas verdient! Einen Großteil der Schuld an diesem Umstand tragen die "Tauscher". Wie schützt man sich jetzt also als Anbieter vor diesen "Tauschern"?

Man schützt seine Programme gegen kopieren. Das hilft etwas, obwohl man weiß, daß damit vielen Leuten die Möglichkeit genommen wird, sich von Ihrem Programm eine Sicherheitskopie zu machen.

Andere Anbieter gehen mit jedem Kopierer vor Gericht, und verlangen Schadensersatz. Bei den wenigen Verhandlungen, bei denen Sie Recht bekommen, hat der "Kopierer" dann das Nachsehen. Er bezahlt mehrer tausend Mark Schadensersatz, nur weil er einem guten Freund eine Kopie gemacht hat. Das ist wahrscheinlich nicht so schön für Ihn, aber was will man als Anbieter anderes tun, um sich vor dem großen Verdienstaustausfall zu schützen, der durch, nennen wir es doch beim Namen, Raubkopien entsteht. Wir haben bis jetzt noch nicht zu solchen Maßnahmen gegriffen, da wir immer noch glauben, daß die meisten Raubkopierer nicht wissen was Sie eigentlich tun!

Der Zeitpunkt aber, an dem auch wir mit Raubkopierern zu Gericht gehen, ist schon abzusehen. Wir tun das nicht, um wie J. R. einfach irgendjemanden fertig zu machen, sondern weil es leider nicht mehr anders geht!

Ein anderer Gesichtspunkt, unter dem man "Tauschaktionen" sehen sollte, ist der, daß viele gute Programme garnicht auf dem Markt erscheinen, weil man weiß, daß

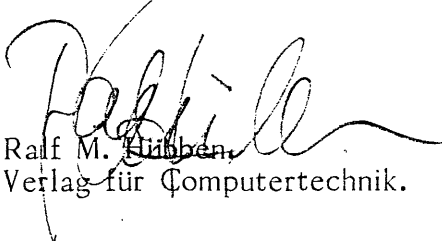
Sie zuwenig gekauft und zuviel getauscht werden! Programme die wir zum Beispiel deshalb nie angeboten haben sind ein Makro-Assembler, ein Toolkit und ein Bildschirmeditor für das Colour Genie.

All das sollten Sie sich einmal durch den Kopf gehen lassen, bevor Sie Tauschlisten herumreichen!

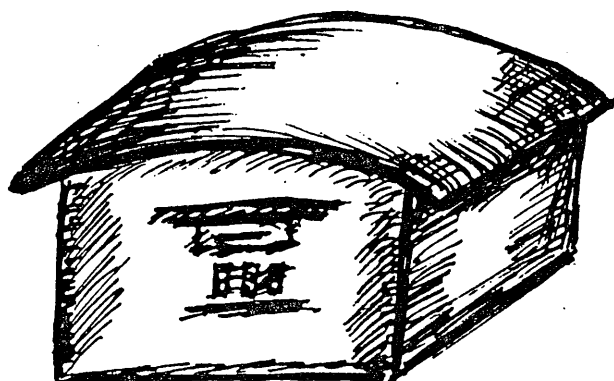
Wir hoffen immer noch, daß man den Leuten das nur klar zu machen braucht, um Ihnen zu zeigen, daß sie sich durch Kopien nur selbst schaden!

Denn, und hier noch einmal die Nachteile der Kopiererei, Programme sind so verschlüsselt, daß Sicherheitskopien für viele unmöglich werden, viele gute Programme erscheinen gar nicht erst, viele gute Anbieter werden sich vom Geschäft zurückziehen, weil sie nichts mehr verdienen können, die Ausstattung von Programmen wird immer schlechter (Dokumentation usw.) werden, weil man hohe Vorbereitungskosten scheut usw.

Mit freundlichem Gruß:



Ralf M. Hübben,
Verlag für Computertechnik.



Meine Antwort
auf den Brief:

LESER- BRIEFE

In einer Fachzeitung las ich vor einiger Zeit, daß es für das Colour Genie ein Interface zum Anschluß an einen Fernschreiber der Marke Siemens T1000 geben soll, kann die entsprechende Passage jedoch nicht mehr finden.

Nun meine Frage: Gibt es wirklich diese Anschlußmöglichkeit, und wo kann man sie erwerben (zu welchem Preis)?

Jörg Jacobsen
Hartwig-Hesse-Straße 9
2000 Hamburg 9

WER HILFT?

Seit Januar 1984 abonniere ich nun die Zeitschrift GENIE DATA und finde es gut, daß ab Heft Nr. 2/84 Teile der Erstausgabe beiliegen werden. Ich hätte aber gerne noch die anderen Ausgaben, die 1983 erschienen sind. Wieviele sind es, und wie kann ich sie Nachbestellen?

Verfügen Sie auch über eine Colour Genie Software-Liste?

Dirk Besemer

Lassen Sie mich zuerst einmal sagen, daß ich die GENIE DATA sehr interessant und überaus wichtig für Computer-Laien, wie ich einer bin finde. Dazu gehört, daß ich Ihnen bei der weiteren Gestaltung und Erhaltung der Zeitschrift viel Erfolg und gute Resonanz wünsche.

Nun zur GENIE DATA 2/83. Ich bin daran interessiert, auch diese Ausgabe und eventuell noch weitere zu erhalten.

Kristian Steffer

Von den zurückliegenden Ausgaben der GENIE DATA sind nur noch die Nummern 3+5/1983 erhältlich. Unser komplettes Software-Angebot finden Sie in diesem Heft.

GENIE DATA

Zu dem von Ihnen (übrigens aus meiner Sicht zurecht) verurteilten Programmaustausch von gewerblicher Software möchte ich wie folgt Stellung nehmen.

Ich habe den Club im November 1983 von Herrn Klaus Schmidt übernommen, da er selbst keine Zeit mehr hatte, um die Leitung durchzuführen. Zu dieser Zeit bestand die Möglichkeit, über den Club Software unter den Mitgliedern zu tauschen. Da mir diese Art der Betätigung auf Grund immer häufiger auftretender gewerblicher Software zu illegal erschien, habe ich in einem zurückliegenden Info die Mitglieder um passende Lösungsvorschläge zum Thema Softwaretausch gebeten und dabei auch auf die Probleme mit käuflicher Software hingewiesen. Der einzige Vorschlag kam von K. Schmidt, wie im letzten Info abgedruckt.

Ich möchte nochmals betonen, daß im Namen des Clubs **keine** Programme, gleichgültig ob privat oder gewerblich, getauscht werden. Die Veröffentlichung von Softwarelisten der Mitglieder wird ab sofort eingestellt.

Genie TRS 80 User Club Bremerhaven
Peter Spieß
Trugenhofenerstraße 27
8859 Rennertshofen 1

Das dieser Club den Programmtausch einstellt, finden wir sehr nachahmenswert!

Wenn Sie, als Leser, Mitglied in einem User-Club werden wollen, können wir Ihnen zum Beispiel den oben genannten sehr empfehlen!

Wenden Sie sich dann bitte an die oben abgedruckte Adresse.

GENIE DATA

DIE SCHRAUBE SOLL GLEICH ROTIEREN

Im Clubinfo 12/83 wurde (leider ohne Angabe des Autors) das Programm "ROTIERENDE SCHRAUBE" veröffentlicht. Es ist wirklich witzig, der Schraube beim Rotieren zuzusehen. Nur dauert es leider sehr lang, bis die Graphik aufgebaut ist. Das geschieht nämlich aner kennenswerterweise mathematisch sauber mit einem kniffligen Algorithmus, der wohl auch kaum wesentlich zu beschleunigen ist.

Da hilft nur eins: Die Graphik muß fertig im Programm vorliegen. Zweckmäßigerweise packt man sie in Stringvariable. Da in eine solche aber nur max. 255 Zeichen passen, muß man sie stückeln, denn sie umfaßt insgesamt 1 KB. Ich entschied mich für acht Variable zu je 128 Zeichen, denn das geht ohne Rest auf.

Man könnte ohne weiteres mit der Befehlsfolge

```
FOR I=0T07: PRINT V$(I);: NEXT
```

die Graphik anzeigen. Dabei würde man aber noch mit dem Auge verfolgen können, wie sie aufgebaut wird. Es ist besser, wenn sie nach RUN schlagartig auf dem Screen steht. Das leistet ein kleines Maschinenprogramm (oberes Assemblerlisting). Auch dieses Unterprogramm steht in einer Stringvariablen (P1\$). Aber daraus ergibt sich ein Problem:

In einem BASIC-Programmtext dürfen keine logischen Nullen stehen, natürlich auch nicht innerhalb einer Zeichenkettenvariablen. Der Interpreter würde sie als EOL-Flag (end of line, Zeilenende) verstehen. Er würde demnach so tun, als begänne die nächste Zeile mitten im String. In diesem Falle habe ich den Akku mit XOR A auf 0 gesetzt. Jetzt kann jedes beliebige Register mit dem Befehl LD r,A mit logisch 0 geladen werden. Das klappt aber nur, weil mein Programm den Akku sonst nicht braucht. Wir werden später sehen, daß es auch anders geht. Zunächst aber zum Programmablauf:

Wie im BASIC MANUAL des Computers beschrieben, kann man einen Parameter in ein mit USR aufgerufenes Maschinen-Unterprogramm übergeben, indem man zunächst das ROM-Unterprogramm ab 0A7Fh aufruft. Für den Blocktransfer mit LDIR ist damit HL als Quelle geladen. DE als Ziel muß die Anfangsadresse des Bildschirms enthalten (3C00h). Der Zähler BC wird achtmal mit der Anzahl der Zeichen in einem String geladen (128d = 0080h). Nach der achten Zeichenübertragung wird BASIC wieder angesprungen.

So weit, so gut. Eine kleine zusätzliche Schwierigkeit ist jedoch noch zu bewältigen, die in der Zeilen- und Variablenorganisation begründet ist: Zwischen dem letzten Zeichen der soeben übertragenen Variablen und dem ersten der nächsten liegen der Reihe nach: Abführungszeichen (alte Var.), EOL (s. o.), Pointer auf übernächste Zeile (zwei Stellen), Zeilennummer (zwei Stellen), Text "V\$(n)=", Anführungszeichen (neue Var.). Das sind zusammen 13 (0Dh) Speicherstellen, die übersprungen werden müssen. Wie man sieht, ist das letztenendes aber auch kein Problem. Aus der Festlegung auf 13 folgt jedoch, daß zwischen die Zeilen mit den Variablen V\$(n) weiter nichts eingefügt werden darf. Ediert werden dürfen sie sowieso nicht, weil der Editor bei der Übernahme der veränderten Zeile in den Programmtext die Codes >7Fh nicht mehr als BASIC-Befehle speichert, sondern als Folge der Buchstaben, mit denen man sie schreibt. Das bedeutet, daß z. B. LET nicht mehr als ASCII 8Ch (das ist nämlich der Graphikblock, den wir an dieser Stelle haben wollen) gespeichert wird, sondern als L-E-T.

Im Programm unseres unbekanntenen Freundes war ebenfalls ein Maschinensprache-Unterprogramm enthalten. Seine einzelnen Codes waren in den DATA-Zeilen gespeichert. Es wurde in die Stringvariable B\$ eingelesen und dort aufgerufen. Aber auch die READ-Schleife kostet Zeit, die man einsparen kann. Deshalb habe ich das fertige Programm in die

Variable P2# geladen. Hierbei ergab sich dasselbe EOL-Problem. Da der Akku hier als Zähler verwendet wird, muß man nun anders vorgehen:

Wenn z. B. in Zeile 190 des unteren Assemblerlistings DE mit dem Beginn des Video-RAM geladen werden soll, so kann man einfach eine Stelle höher gehen (3C01h) und anschließend das LSB in E dekrementieren. (Die dergestalt veränderten Assemblerzeilen sind mit ";***" gekennzeichnet). In vielen Fällen ist bei dieser Methode allerdings Vorsicht geboten, weil beim Dekrementieren auf Null das Z-Flag gesetzt wird. Das könnte einige Befehle später verhängnisvoll werden.

Von Fall zu Fall ist es machbar, das EOL selbst für eine unvermeidbare Null zu mißbrauchen. Das trifft z. B. zu, wenn eine im Maschinenprogramm enthaltene Zeichenkette angezeigt werden soll, auf die HL zeigt, und die mit logisch 0 abgeschlossen sein muß. Man packt sie kurzerhand an das Ende des Unterprogramms und läßt die Abführungszeichen des Programmstrings einfach weg.

Das modifizierte BASIC-Listing der "SCHRAUBE" kann in dieser Form natürlich nicht mehr abgetippt werden, denn Stringvariable akzeptieren in der direkten Eingabe (fast) nur die Codes >ASCII 1Fh und <ASCII 80h. Der hier gelistete BASIC-Text hat eine weitere Besonderheit, auf die ich noch eingehen möchte:

Die beiden Stellen unmittelbar hinter dem Variablenzeiger (VARPTR = variable pointer) zeigen durchaus nicht immer auf den Beginn des Variablentextes. Das tun sie nur bei nicht dimensionierten Strings. In meinem Fall mit einer Dimension von V*(n) mußte ich daher 8 bzw. 9 Stellen nach dem VARPTR PEEKen (BASIC-Zeile 120).

Arnulf Sopp, Wakenitzstr. 8, 2400 Lübeck 1, ☎ 0451-791926


```

00100 :ROTIERENDE SCHRAUBE: GRAPHIK AUF BILDSCHIRM LADEN
00110
0000      00120      ORG      0000H      ;ABHAENGIG VON P1#
00130
0000 CD7F0A 00140      CALL     0A7FH      ;HL MIT $--ANFANG (UV) LADEN
0003 AF      00150      XOR      A          ;A=0, WEIL ASCII 0 IN BASIC = EOL
0004 0608    00160      LD       B,08H      ;B STRINGVARIABLE
0006 163C    00170      LD       D,3CH      ;SCREEN-ADR., MSB
0008 5F      00180      LD       E,A        ;D0., LSB (DE=3C00H)
0009 C5      00190 LOOP    PUSH     BC        ;WIRD NOCH GEBRAUCHT
000A 47      00200      LD       B,A        ;ZAEHLER MSB = 0
000B 0E80    00210      LD       C,80H      ;LSB; BC=0080H (128 Z. IM STRING)
000D EDB0    00220      LDIR     ;BLOCKTRANSFER, STRING -> SCREEN
000F 0E0D    00230      LD       C,0DH      ;13 STELLEN UEBERSPRINGEN
0011 09      00240      ADD     HL,BC       ;AB (HL) WEITERLADEN
0012 C1      00250      POP     BC         ;VARIABLENZAEHLER
0013 10F4    00260      DJNZ    LOOP       ;FALLS NOCH NICHT ALLE ANGEZEIGT
0015 C9      00270      RET      ;SONST ZURUECK NACH BASIC
00280
0000      00290      END
00000 TOTAL ERRORS
34455 TEXT AREA BYTES LEFT

```

```

00100 :ROTIERENDE SCHRAUBE: ROTATIONSPROGRAMM
00110
0000      00120      ORG      0000H      ;ABHAENGIG VON P2#
00130
0000 21FE3F 00140      LD       HL,3FFE3F
0003 11FF3F 00150      LD       DE,3FFF3F
0006 01FF03 00160      LD       BC,03FF03
0009 EDB8    00170      LDDR
000B 0E10    00180      LD       C,10H
000D 11013C 00190      LD       DE,3C01H   ;***
0010 1D      00200      DEC     E          ;***
0011 213F01 00210 LOOP1   LD       HL,013FH   ;***
0014 25      00220      DEC     H          ;***
0015 19      00230      ADD     HL,DE
0016 EDA0    00240      LDI
0018 E5      00250      PUSH    HL         ;SHORTCUT FUER
0019 D1      00260      POP     DE         ;      HL->DE
001A 79      00270      LD       A,C
001B B7      00280      OR      A
001C 20F3    00290      JR      NZ,LOOP1
001E 3E10    00300      LD       A,10H
0020 21013C 00310      LD       HL,3C01H   ;***
0023 2D      00320      DEC     L          ;***
0024 E5      00330 LOOP2   PUSH    HL
0025 D1      00340      POP     DE
0026 13      00350      INC     DE
0027 0E04    00360      LD       C,04H
0029 EDB0    00370      LDIR
002B 113C01 00380      LD       DE,013CH   ;***
002E 15      00390      DEC     D          ;***
002F 19      00400      ADD     HL,DE
0030 3D      00410      DEC     A
0031 B7      00420      OR      A
0032 20F0    00430      JR      NZ,LOOP2
0034 C9      00440      RET
00450
0000      00460      END
00000 TOTAL ERRORS
34535 TEXT AREA BYTES LEFT

```


Computeritis

Mein Freund Willi leidet seit drei Monaten an einer unheilbaren Krankheit. Die Symptome sind vielfältig: Schlaflosigkeit, verbunden mit Nachtschwärmerei (sitzt noch morgens um 4 Uhr vorm Monitor), Vereinsamung (kein Skatabend, kein Schachtreff) mehr, beängstigende Essensunlust (dabei ist er schon so schlank wie eine Bohnenstange), rote Augen (vom Bildschirm-Anstarren). Am schlimmsten scheint mir eine ausgeprägte Sprachverwirrung zu sein: "15040 FORI = 32807T032927 STEP 40: POKEI,32: NEXT", Jubelte er mir bei meinem jüngsten Kurzbesuch entgegen. Und etwas deutlicher: "Hast Du's kapiert, ich programmiere 'Formel 1'?" Nichts hab' ich kapiert und auch die arme Wilma nicht, seine geplagte Frau. "Lassen Sie ihn sich austoben", hat man ihr als einzige Therapie geraten, ansonsten Achselzucken.

Die Diagnose ist klar: **Computeritis**.

Angefangen hatte ja alles mit dem neuen Heimcomputer, den sich Willi so sehnlich gewünscht hatte. "Der Fehler meines Lebens" sagt Wilma. "Nach der Arbeit kaum die Tür drin, sitzt er schon vor seinem Kasten." Sie hat schon alles mögliche ausprobiert, ihren Willi aus der Computerfessel zu befreien. Sie hat sein Lieblingsgericht gekocht, den Tisch fein gedeckt, ihr bestes Kleid angezogen, Theaterkarten gekauft. Aber gegen "18001 ZE=500-PH*45: PR=PR+100: RETURN" kommt sie einfach nicht an!

Bis gestern - denn da ging ihr ein Licht auf. Sie machte ihm dunkel! Richtig, sie drehte schlicht und wirkungsvoll die entsprechende Sicherung heraus. Und jetzt hat sie das Heft wieder in der Hand. Gibt ihm sozusagen als Heilbehandlung nur stundenweise Strom für seinen Computer. Ob die Entwöhnungskur hilft? Fragt sich.

Namen von der Redaktion geändert. Ähnlichkeiten mit lebenden Personen wären rein zufällig. Aus: AOK Magazin 2/84

PUT TO adresse - ein neuer BASIC-Befehl

Nein, es ist durchaus nicht mein Lieblingsthema, BASIC im RAM auf- und abzuschubsen. Hat man aber einmal über ein derartiges Thema nachgedacht, um ein paar Seiten für das Clubinfo daraus zu machen, kommen in der Folge ständig neue Ideen zum gleichen Problemkreis.

Heute soll der BASIC-Programmierer etwas davon haben: Der neue Befehl PUT TO geht noch ein Stück weiter als die bisherigen Vorschläge. Er verschiebt ebenfalls BASIC, aber im Speicher befindliche Programme bleiben erhalten. Sie rutschen kurzerhand an die neue Stelle mit und tun dort so, als sei nichts gewesen. In BASIC merkt der User nur dann etwas, wenn er mit PRINT MEM fragt, wieviel Platz er noch im RAM hat.

Die Befehlsyntax lautet wie in der Überschrift. Dabei können die Blanks entfallen, die Adresse muß dezimal eingegeben werden. Mit dieser Adresse ist die BASIC-Untergrenze gemeint. Es ist die Stelle, auf die der DATA-Zeiger deutet, bevor RUN eingegeben wurde. In diesem Vor-RUN-Zustand befindet sich übrigens auch das Programm nach der Verschiebung mit PUT TO: Variable sind gelöscht, der DATA-Zeiger weist unmittelbar vor den Programmtext.

Die einzugebende Adresse muß ohne den zuvor besprochenen EG 64 MBA mindestens 17128 betragen, nach oben sind nur die physikalischen Grenzen des RAM gesetzt. Es obliegt dem Anwender, darauf zu achten, daß von dieser Adresse an nach oben für das Programm und später hinzukommende Variable genügend Platz ist. Diese Eigenverantwortung des Users macht deutlich, daß gewisse Grundkenntnisse über die Speicher-verwaltung von BASIC nötig sind wie etwa auch bei den Befehlen VARPTR und USR.

In diesem Zusammenhang ist auch der Grund zu sehen, weshalb ich einerseits eine BASIC-Utility vorstelle, andererseits aber nur ein Assembler-Listing anbiete. Der BASIC-Programmierer ohne Maschinensprachkenntnisse wird den Befehl PUT TO wohl nie brauchen, der Z80-Freak hat aber einen Assembler, mit dem er PUT TO seinen eigenen Bedürfnissen anpassen kann (z. B. ORG verlegen).

Das Programm läuft nicht unter Disk-BASIC, weil der PUT-Befehl zweckentfremdet wird. Mit einer kleinen Änderung ist aber auch das möglich: Die Befehle

	RST	08H	
	DEFB	TO	
werden ersetzt durch	DEC	HL	;als Zeiger für RST 10H
	RST	10H	;nächstes Zeichen laden
	CP	TO	;TO-Token?
	JP	NZ,627BH	;PUT-Routine Disk-BASIC
	INC	HL	;war in RST 8 enthalten

(Die Adresse 627BH kann von DOS zu DOS variieren. Schaut euch in 4183/4184H den genauen Vektor an und ermittelt seinen dezimalen Wert mit PRINT PEEK(&h4183)+PEEK(&h4184)*256.)

Und was das Ganze soll? Das Himem wird für tausend kleine Popelprogramme gebraucht; man setzt pausenlos die Memsize für alles Mögliche. Mit PUT TO wird stattdessen die BASIC-Untergrenze bestimmt, so daß

man z. B. ganz oben einen Monitor fahren kann und unten die Tasten entprellt oder (nein, es ist wirklich nicht mein Lieblingsthema) für EDTASM Platz hat. Irgendwo lümmelt sich dann BASIC beliebig dazwischen.

Auch dies ist ein solches Popelprogramm, denn man wird es recht selten brauchen. Immerhin, es kann im Gegensatz zum Meisten dieser Art mitten in der BASIC-Arbeit nachgeladen werden. So war für mich an der ganzen Geschichte hauptsächlich die Frage interessant, wie man überhaupt einen neuen BASIC-Befehl implementiert. Es gibt einige Möglichkeiten:

Zum gleichen Zweck hätte auch das Befehlswort MOVE BASIC dienen können. Dabei würde zunächst ein Syntax Error erkannt werden. Im Verlauf der Fehlerbehandlungsroutine wird auch ins RAM verzweigt, das wir für diesen Zweck umprogrammieren können. Da steht jetzt ein Jump zu unserer neuen Verb Action-Routine. Dort wird der Stack nach der Herkunft des Fehlers durchpflügt. Kommt er aus der Variable Assignment-Routine, checken wir, ob das fehlerhafte Statement genau MOVE BASIC oder MOVEBASIC und nicht anders lautete. Ist das der Fall, ist es kein Fehler und der Spaß geht los.

Zu umständlich? Einverstanden. Da der BASIC-Verschiebebefehl wohl hauptsächlich nicht programmiert, sondern als direktes Statement eingetippt wird, vereinfacht sich die Sache: Der hierfür zuständige DOS-Exit ist 41B2H (JP 6033H, je nach DOS). Wir verbiegen ihn auf unsere Routine und können den Stack Stack sein lassen. Jetzt wird wieder untersucht, ob die folgenden Zeichen genau M-O-V-E-(-)B-A-S-I-C hießen.

Immer noch zu umständlich? Wieder einverstanden. Die einfachste Lösung ist immer, einen Befehl zu "mißbrauchen", den es bereits gibt, der ein Token hat. Wird ein Token angetroffen, springt der Interpreter in die zuständige Verb Action-Routine. So ziemlich jede dieser Routinen verzweigt, z. T. mehrmals, in das freie RAM, wo wir Vektoren auf unsere Ergänzung richten können. Nur ist leider bis zum Sprung ins RAM häufig schon allerhand passiert, das wir wieder einnorden müssen. Deshalb ist eigentlich nur eine einzige Methode wirklich empfehlenswert:

Die reinen Disk-BASIC-Befehle, die unter Level 2 zu einem ?L3 Error oder ?SN Error führen, springen das RAM sofort an. Noch ist alles im Lot, und unser Programm kann kurz und bündig bleiben. Deshalb die zahllosen BASIC-Erweiterungen, die mit NAME aufgerufen werden. Wenn dem Befehl (hier PUT) noch ein weiteres Zeichen oder Token (hier T0) folgt, kann gleich am Anfang festgestellt werden, ob alles seine Richtigkeit hat. Spätere Syntaxchecks übernimmt wieder der Interpreter. Wenn z. B. PUT T0 20000? eingegeben wurde, wird der Befehl zunächst korrekt ausgeführt. Erst dann erkennt Mr. Microsoft das Fragezeichen und versteht Bahnhof: ?SN Error. Wir haben ihn ja nur ergänzt, nicht verändert.

Mit EDTASM ist die Eingabe des Programms am bequemsten. Wer diesen Assembler nicht hat, muß ihn entweder für sehr viel Geld kaufen oder gewisse Mitglieder unseres Clubs ansprechen, die angeblich jedes Copyright mißachten und Programme weitergeben. Einem böswilligen Gerücht zufolge soll es kein Mitglied geben, das es nicht tut. Muß ich deutlicher werden?

```

00100 ;***          PUT TO adresse          ***
00110 ;***
00120 ;*** Ein neuer BASIC-Befehl zum freien Verschieben ***
00130 ;*** von BASIC inkl. Programmtext im RAM. Variable ***
00140 ;*** werden dabei gelöscht.          ***
00150
00160 ;(C) 1984 by Arnulf Sopp, Wakenitzstr. 8, D-2400 Lübeck 1
00170
00180
00190 ;Symbolvereinbarungen:
4182 00200 PUT EQU 4182H ;Adresse PUT-Verb Action-Routine
06CC 00210 BASIC EQU 06CCH ;BASIC-Warmstartadresse
1E5A 00220 ASCINT EQU 1E5AH ;ASCII-String -> DE als Integer
40FF 00230 DATPTR EQU 40FFH ;DATA-Pointer
40A4 00240 PSTPTR EQU 40A4H ;Pointer auf BASIC-Textanfang
40F9 00250 VLTPTR EQU 40F9H ;dto. auf Anfang Variablenliste
40FB 00260 ARRANF EQU 40FBH ;dto. Anf. dimens. Var.
40FD 00270 ARREND EQU 40FDH ;dto. Ende (Anfang freies RAM)
00BD 00280 TO EQU 0BDH ;TO-Token
00290
00300 ;PUT-Vektor verbiegen:
4182 00310 ORG PUT
4182 C30080 00320 JP PUTTO ;neue Verb Action-Routine
00330
00340 ;neue Verb Action-Routine:
8000 00350 ORG 8000H ;oder je nach RAM-Größe
8000 CF 00360 PUTTO RST 08H ;Syntax Error, falls
8001 BD 00370 DEFB TO ;nicht TO angetroffen
8002 CD5A1E 00380 CALL ASCINT ;Adresse in DE einlesen
8005 2AFF40 00390 LD HL,(DATPTR) ;für die Ermittlung
8008 ED4BF940 00400 LD BC,(VLTPTR) ;der Programmlänge
800C 0B 00410 DEC BC ;auf die 0 davor
800D DF 00420 RST 18H ;rauf oder runter?
800E C8 00430 RET Z ;falls dieselbe Adresse
800F E5 00440 PUSH HL ;HL und BC vertauschen
8010 C5 00450 PUSH BC
8011 E1 00460 POP HL
8012 C1 00470 POP BC
8013 D5 00480 PUSH DE ;neue Adresse für später
8014 380B 00490 JR C,UPLOAD ;Carry, falls rauf
8016 C5 00500 PUSH BC ;als Quelle für LDIR
8017 ED42 00510 SBC HL,BC ;Programmlänge
8019 E5 00520 PUSH HL ;als Zähler für LDIR
801A C1 00530 POP BC ;nach BC laden
801B 03 00540 INC BC ;Bytezähler korrig.
801C E1 00550 POP HL
801D EDB0 00560 LDIR ;BASIC umschaukeln
801F 180C 00570 JR CNGPTR ;dort weiter
8021 E5 00580 UPLOAD PUSH HL ;spätere Quelle für LDDR
8022 B7 00590 OR A ;Carry löschen
8023 ED42 00600 SBC HL,BC ;wie oben
8025 E5 00610 PUSH HL
8026 C1 00620 POP BC
8027 03 00630 INC BC ;Bytezähler korrig.
8028 19 00640 ADD HL,DE ;ans Ende des Programms
8029 EB 00650 EX DE,HL ;Ziel DE laden
802A E1 00660 POP HL ;Quelle
802B EDB8 00670 LDDR ;wie oben, von hinten
00680
00690 ;BASIC-Zeiger nachstellen:
802D E1 00700 CNGPTR POP HL ;neue Adresse
802E 22FF40 00710 LD (DATPTR),HL ;DATA-Pointer laden
8031 23 00720 INC HL ;eins höher
8032 22A440 00730 LD (PSTPTR),HL ;PST-Pointer laden
00740

```

```

00750 ;Zeilenzeiger im Programmtext nachstellen:
8035 E5      00760      PUSH   HL           ;erster Zeilenpointer
8036 23      00770      INC     HL           ;insges. 4X erhöhen
8037 D1      00780 LOOP   POP     DE
8038 23      00790      INC     HL           ;auf 1. Programmstatement
8039 23      00800      INC     HL           ;erhöhen (die nächste 0
803A 23      00810      INC     HL           ;muß EOL sein.)
803B AF      00820      XOR    A           ;A=0 für CPTR
803C 47      00830      LD     B,A         ;dto. BC (=256!)
803D 4F      00840      LD     C,A
803E EDB1    00850      CPTR                    ;0 als EOL aufsuchen
8040 EB      00860      EX     DE,HL       ;Register umordnen
8041 73      00870      LD     (HL),E      ;Zeilenpointer korri-
8042 23      00880      INC     HL           ;gieren
8043 72      00890      LD     (HL),D
8044 EB      00900      EX     DE,HL       ;alte Registerordnung
8045 E5      00910      PUSH   HL           ;neuer Zeilenpointer
8046 7E      00920      LD     A,(HL)      ;auf EOL prüfen
8047 23      00930      INC     HL           ;eine zweite 0?
8048 B6      00940      OR     (HL)
8049 20EC    00950      JR     NZ,LOOP     ;nein, weiter
00960
00970 ;Variablenpointers laden und Ende:
804B 23      00980      INC     HL           ;auf neuen VLTPTTR erhöhen
804C 22F940  00990      LD     (VLTPTTR),HL
804F 22FB40  01000      LD     (ARRANF),HL
8052 22FD40  01010      LD     (ARREND),HL
8055 E1      01020      POP    HL           ;Stack korrigieren
8056 C3CC06  01030      JP     BASIC
01040
06CC      01050      END     BASIC
00000 TOTAL ERRORS
32283 TEXT AREA BYTES LEFT

```

```

ARRANF 40FB 00260 01000
ARREND 40FD 00270 01010
ASCINT 1E5A 00220 00380
BASIC 06CC 00210 01030 01050
CNGPTR 802D 00700 00570
DATPTR 40FF 00230 00390 00710
LOOP 8037 00780 00950
PSTPTR 40A4 00240 00730
PUT 4182 00200 00310
PUTTO 8000 00360 00320
TO 00BD 00280 00370
UPLOAD 8021 00580 00490
VLTPTTR 40F9 00250 00400 00990

```

Die Idee zu diesem Programm bekam ich als ein anderer Computer-Freak mir ein kleines BASIC-Programm zeigte, daß mit der TIME-Funktion des Betriebssystems eine graphische Uhr in der unteren rechten Ecke des Bildschirms anzeigte. Nachteil des Programmes war, daß der Computer nur die Uhr anzeigte und nichts anderes machen konnte. Also warum nicht ein Assembler-Programm schreiben, daß die Uhr im Interrupt immer anzeigt, wobei man ansonsten den Computer noch für etwas Anderes nutzen kann.

Dieses Programm wurde speziell für NEWDOS80 Version 2.0 geschrieben, da dieses im Augenblick das verbreitetste und von den Möglichkeiten das umfangreichste Betriebssystem ist. Für reine Level II - User ist dieses Programm nicht geeignet, da es DOS-spezifische Routinen benutzt. Zur Interrupt-Technik: Unter Punkt 3.8 und 3.9 steht im NEWDOS-Manual die Erklärung zur Aktivierung und Deaktivierung von Interrupt-Routinen.

Dazu einige Erklärungen: Ein Interrupt ist eine Unterbrechung, die beim TRS-80 alle 25 Millisekunden durchgeführt wird. Dafür steht dem Programmierer eine Interruptkette zur Verfügung, in die er sein eigenes Programm einfügen kann.

Dazu benötigt er praktisch 2 Programme. 1. das Startprg., das im DE-Registerpaar die Startadresse des Anwenderprg. hat und dann ein CALL 4410H ausführt. In meinem Prg. wird am Anfang zusätzlich noch HIMEM begrenzt. Dazu muß die Speicherstelle 4049H den neuen Wert bekommen. Das 2. Prg. ist das Anwenderprg., das durch den Interrupt immer wieder aufgerufen werden soll. Dabei ist folgendes zu beachten: Die ersten 4 Bytes sind für die Interrupt-Routine reserviert. In den ersten beiden Bytes speichert die Routine den Pointer, der beim Aufruf einen beliebigen Wert (hier 0C9H, 44H) haben kann. Das nächste Byte gibt die Anzahl der 25ms - Interrupts an, nach denen die Routine das Prg. aufrufen soll. Hier wird das Prg. jede Sekunde = $25 \text{ ms} * 40$ aufgerufen. Das vierte Byte dient dem DOS als Zähler, der den Inhalt bei jedem Interrupt um 1 herunterzählt und wenn dieser 0 ist, den Inhalt des 3. Bytes im 4. wieder speichert und das Prg. beim 5. Byte ausführt.

Das Anwenderprg. ab dem 5. Byte zeichnet erst einen Kasten mit einem Blockladebefehl auf den Bildschirm, dann die Doppelpunkte zwischen Stunden, Minuten und Sekunden. Danach wird der Wert der aktuellen Zeit (Speicheradresse: 4041H) in einen Puffer geladen. Jedes Byte von Stunden, etc. wird dann in 2 Werte zerlegt (CALL HEXCV), danach wird die entsprechende Stelle in der Graphik-Ziffern-Tabelle gesucht (CALL ABFRAG) und auf dem Bildschirm ausgegeben (CALL Z1 und CALL Z2). Danach bekommen die Register ihre alten Werte und das Prg. springt mit einem RET zurück.

Noch einige Anmerkungen zum Assembler: Ich habe das Prg. mit dem Prg. MACRO 80 bearbeitet. Vorteil des Assemblers ist, daß er normale ASCII-Files, die z. B. mit SCRIPSIT geschrieben sind, assembliert, sowie, daß er keine Zeilennr. benötigt, wie sie z. B. EDTASM generiert.

Wer sich nicht die Arbeit machen will, das Prg. einzutippen, sei es nun mit MACRO 80, EDTASM oder einen anderen Assembler, kann mir eine Diskette zuschicken und bekommt den Source-File und einen assemblierten und gelinkten CMD-File.

Adresse: Klaus van der Meulen, Altendorfer Str. 502, 4300 Essen
Tel.: 0201/670338


```

;-----;
; Uhr in Interrupt-Routine ;
;-----;
; Klaus van der Meulen, 21.02.84 ;
;-----;

```

```

ORG OFCOOH ; STARTADRESSE DES PRG.
;
CLOCK EQU 4041H ; ZEIT IM DOS
VIDEO EQU 3COOH ; BEGINN DES VIDEO-RAM
HIMEM EQU 4049H ; HIMEM-BEGRENZUNG
;

```

```

;-----;
; START-PROGRAMM, UM HIMEM ZU BEGRENZEN UND DAS ;
; ANWENDER-PRG. IN DIE INTERRUPT-KETTE ANZUHAENGEN ;
;-----;

```

```

START: PUSH DE ; RETTE DIE REGISTER DE UND
        PUSH HL ; HL AUF DEN STACK
        LD HL,HIMEM ; BEGRENZE HIMEM IM DOS
        LD DE,ANWPRG ; UM DAS ANWENDERPRG. VOR
        LD (HL),E ; BASIC, ETC. ZU SCHUETZEN
        INC HL ;
        LD (HL),D ;
        ;
        LD DE, ANWPRG ; HAENGE ANWENDERPRG IN
        CALL 4410H ; INTERRUPTKETTE EIN
        POP DE ;
        JP 402DH ; ZURUECK IN'S DOS
        ;

```

```

;-----;
; ANWENDERPRORGRAMM ;
;-----;

```

```

ANWPRG: DB OC9H, 44H ; DOS-POINTER, BELIEBIGER WERT
        DB 28H ; ANZAHL DER 25 MS INTERRUPTS
        DB 28H ; DOS-ZAHLER FÜR DEN NÄCHSTEN AUFRUF
        ;
        PUSH HL ; RETTE DIE REGISTER:
        PUSH BC ; HL, BC, DE UND AKKU
        PUSH DE ;
        PUSH AF ;
        ;

```

```

KASTEN: LD BC,23 ; BRINGE DEN KASTEN AUF DEN
        LD DE,VIDEO+41 ; BILDSCHIRM
        LD HL,KAST1 ;
        LDIR ;
        LD BC,23 ;
        LD DE,VIDEO+105 ;
        LD HL,KAST2 ;
        LDIR ;
        LD BC,23 ;
        LD DE,VIDEO+169 ;
        LD HL,KAST2 ;
        LDIR ;
        LD BC,23 ;
        LD DE,VIDEO+233 ;
        LD HL,KAST3 ;
        LDIR ;

```

```

DPLPKT: LD HL,VIDEO+112 ; ZEICHNE DIE TRENNENDEN
        LD (HL),140 ; DOPPELPUNKT&ZWISCHEN
        LD HL,VIDEO+120 ; STUNDEN UND MINUTEN BZW.
        LD (HL),140 ; MINUTEN UND SEKUNDEN AUF

```

```

LD      HL,VIDEO+176      ;DEN BILDSCHIRM
LD      (HL),131         ;
LD      HL,VIDEO+184     ;
LD      (HL),131         ;
LD      A,(CLOCK)        ;LADE DEN WERT VON CLOCK IN AKKU
LD      HL,PUFFER        ;RETTE ZEIT IN PUFFER
LD      (HL),A           ;
INC     HL                ;
LD      A,(CLOCK+1)      ;
LD      (HL),A           ;
INC     HL                ;
LD      A,(CLOCK+2)      ;
LD      (HL),A           ;
LD      A,(PUFFER)       ;
CALL    HEXCV            ;WANDLE WERT IN AKKU IN ZWEI WERTE
LD      A,B               ;ERSTE ZIFFER IN AKKU
CALL    ABFRAG           ;WELCHE ZIFFER AUS TAB
LD      HL,VIDEO+122     ;
CALL    Z1               ;GEBE OBEREN TEIL AN VIDEO + 122 AUS
LD      HL,VIDEO+186     ;
CALL    Z2               ;GEBE UNTEREN TEIL AN VIDEO + 186 AUS
LD      A,C               ;ZWEITEX ZIFFER IN AKKU
CALL    ABFRAG           ;
LD      HL,VIDEO+125     ;
CALL    Z1               ;
LD      HL,VIDEO+189     ;
CALL    Z2               ;
LD      A,(PUFFER+1)     ;
CALL    HEXCV            ;
LD      A,B               ;
CALL    ABFRAG           ;
LD      HL,VIDEO+114     ;
CALL    Z1               ;
LD      HL,VIDEO+178     ;
CALL    Z2               ;
LD      A,C               ;
CALL    ABFRAG           ;
LD      HL,VIDEO+117     ;
CALL    Z1               ;
LD      HL,VIDEO+181     ;
CALL    Z2               ;
LD      A,(PUFFER+2)     ;
CALL    HEXCV            ;
LD      A,B               ;
CALL    ABFRAG           ;
LD      HL,VIDEO+106     ;
CALL    Z1               ;
LD      HL,VIDEO+170     ;
CALL    Z2               ;
LD      A,C               ;
CALL    ABFRAG           ;
LD      HL,VIDEO+109     ;
CALL    Z1               ;
LD      HL,VIDEO+173     ;
CALL    Z2               ;
POP     AF                ;RESTAURIERE DIE REGISTER MIT
POP     DE                ;DEN ALTEN WERTEN
POP     BC                ;
POP     HL                ;

```

16

```

RET                                ;SPRINGE AUS INTERRUPT-PRG. ZURUECK
;
;-----;
; UNTERPROGRAMME DURCH "CALL" AUFGERUFEN ;
;-----;
Z1:  LD      A,(DE)                ;GEBE DEN OBEREN TEIL DER ZIFFER
     LD      (HL),A                ;AUF DEM BILDSCHIRM AUS
     INC     HL                    ;
     INC     DE                    ;
     LD      A,(DE)                ;
     LD      (HL),A                ;
     RET                                ;
;
Z2:  INC     DE                    ;GEBE DEN UNTEREN TEIL DER ZIFFER
     LD      A,(DE)                ;AUF DEM BILDSCHIRM AUS
     LD      (HL),A                ;
     INC     HL                    ;
     INC     DE                    ;
     LD      A,(DE)                ;
     LD      (HL),A                ;
     RET                                ;
;
;-----;
; ABFRAGE, WELCHE ZIFFER AUS DER TABELLE AUF DEN BILDSCHIRM KOMMT.;
; DER WERT ERRECHNET SICH : DE:= ADRESSE VON TAB + (AKKU * 4) ;
;-----;
ABFRAG: ADD     A,A                ;AKKU := AKKU + AKKU
       ADD     A,A                ;AKKU := AKKU + AKKU
; ERGEBNIS: AKKU BEKOMMT DEN 4-FACHEN WERT
       LD      DE,0000H           ;LADE DE MIT 0
       LD      E,A                ;LADE DEN WERT VON AKKU IN E
       LD      HL,TAB             ;LADE DIE ADRESSE DES TAB.-ANFANG IN HL
       ADD     HL,DE              ;ADDIERE DEN WERT VON DE ZU HL
       EX     DE,HL              ;WECHSELE DIE WERTE VON HL UND DE
;
       RET                        ;KEHRE ZURUECK
;
;-----;
; KONVERTIERE DEN WERT IM AKKU IN ZWEI WERTE. ;
; Z.B. AKKU = OFH, DANN B = 1 UND C = 5 ODER ;
; AKKU IN HEX = B * 10 + C IN DEZIMAL ;
;-----;
HEXCV: LD      DE,010AH           ;LADE DE MIT DEM WERT 010AH
       LD      HL,0000H           ;HL := 0
VERGL: CP      L                  ;AKKU = L * 0AH ?
       JP      M,WEITER           ;
       ADD     HL,DE              ;
       JP      VERGL             ;
WEITER: DEC     H                  ;
       LD      B,H                ;
       SUB     L                  ;
       ADD     A,0AH              ;
       LD      C,A                ;
       RET                        ;
;
;-----;
; PUFFER FUER DIE ZEIT ;
;-----;
PUFFER: DB     'SMH'              ;
;
;-----;
; GRAPHIK-TABELLE FUER DIE ZIFFERN ;
;-----;

```

```

;-----;
TAB:  DB      151, 171, 141, 142 ; ZIFFER 0
      DB      175, 128, 143, 143 ; ZIFFER 1
      DB      179, 187, 141, 140 ; ZIFFER 2
      DB      179, 181, 140, 143 ; ZIFFER 3
      DB      149, 176, 131, 143 ; ZIFFER 4
      DB      183, 179, 140, 143 ; ZIFFER 5
      DB      181, 176, 141, 142 ; ZIFFER 6
      DB      131, 155, 138, 128 ; ZIFFER 7
      DB      183, 187, 141, 142 ; ZIFFER 8
      DB      183, 187, 128, 143 ; ZIFFER 9
;
;-----;
; GRAPHIK-KASTEN
;-----;
KAST1: DB      156, 140, 140, 140 ; OBERER TEIL DES KASTENS
      DB      140, 140, 140, 140 ;
      DB      140, 140, 140, 140 ;
      DB      140, 140, 140, 140 ;
      DB      140, 140, 140, 140 ;
      DB      140, 140, 172      ;
KAST2: DB      149, 128, 128, 128 ; MITTERER TEIL
      DB      128, 128, 128, 128 ;
      DB      128, 128, 128, 128 ;
      DB      128, 128, 128, 128 ;
      DB      128, 128, 128, 128 ;
      DB      128, 128, 170      ;
KAST3: DB      131, 131, 131, 131 ; UNTERER TEIL
      DB      131, 131, 131, 131 ;
      DB      131, 131, 131, 131 ;
      DB      131, 131, 131, 131 ;
      DB      131, 131, 131, 131 ;
      DB      131, 131, 131      ;
;
END      START

```

Nützliche PEEK's und POKE's fürs GENIE
=====

Mergen von BASIC-Programmen:

1. Programm A laden
2. PRINT PEEK(16633)
3. Ergebnis > 1? Wenn ja, dann weiter bei 4.
Wenn nein, dann weiter bei 5.
4. POKE 16548, PEEK(16633)-2: POKE 16549, PEEK(16634)
Weiter bei 6.
5. POKE 16548, PEEK(16633)+254: POKE 16549, PEEK(16634)-1
6. Programm B laden. (Die Zeilennummern in Programm B müssen größer sein,
als die in Programm A verwandten)
7. POKE 16548,233: POKE 16549,66
8. Fertig, die Programme sind gemergt.

Sperrern der BREAK-Taste:

Sperrern: POKE 16396,175

Aktivieren: POKE 16396,201

Mit "SHIFT" "BREAK" abbrechen: POKE 16396,165

Mit "SHIFT" "BREAK" "↓" abbrechen: POKE 16396,123: POKE 16397,61

Sperrern des LIST-Befehls:

POKE 16863,195: POKE 16864,163: POKE 16865,30

Nach diesen Poke's reagiert der LIST-Befehl wie ein RUN. Voraussetzung ist allerdings, daß eine Zeile 0 im Programm existiert!

POKE 16863,195: POKE 16864,114: POKE 16865,6

Damit ist der LIST-Befehl vollständig blockiert. (Nach Eingabe von LIST springt der Computer in die Memory-Size zurück.)

MEM-SIZE, bzw. READY? in einem BASIC-Programm setzten:

Die Adressen 40B1H und 40B2H (16561D, 16562D) enthalten in der Form LSB/MSB die Adresse des letzten Speicherplatzes, der für Basic, bzw. für Basic-Variablen zur Verfügung steht. Der entsprechende Dezimalwert (in Basic ermittelbar durch: PRINT PEEK(16561)+256*PEEK(16562)) ist gleich dem um 2 verminderten Eingabewert bei READY?, bzw. MEM-SIZE.

Will man nun einen RAM-Bereich für ein Maschinenprogramm innerhalb eines Basic-Programmes schützen, so geht man folgendermaßen vor:

A ist die erste Adresse des Maschinenprogrammes; natürlich in Dezimal; setze A=A-2 und direkt am Programmanfang

POKE 16561,A AND 255: POKE 16562, INT(A/256): CLEAR 50

Abschalten der Tastatur während größerer Berechnungen:

Abschalten: POKE 16406,82

Wieder einschalten: POKE 16406,227

Achtung: Vor einem INPUT oder INKEY\$-Befehl ist die Tastatur vorher wieder einzuschalten. Ebenso muß die Tastatur vor dem Programmende wieder eingeschaltet werden.

J o y s t i c k i n t e r f a c e f ü r s G E N I E

=====

Hier beschreibe ich den Einbau eines Joysticks in das GENIE-Gehäuse, mit dem man alle Spiele spielen kann, die die Pfeiltasten und die Spacetaste benutzen.

Als Material benötigt man den "Atari"-Joystick und eine Joystickbuchse. Die Buchse kann man im allgemeinen bei einem Radio- und Fernsehändler als Ersatzteil bestellen.

Öffnen des GENIES und Ausbau der Platinen:

1. Netzstecker ziehen und alle Verbindungskabel hinten am GENIE entfernen.
2. Entfernen der schwarzen Tastenkappen an der Hinterseite (vorsichtig abziehen).
3. Nun das GENIE mit der Tastatur nach unten vor sich hin legen und mit einem Kreuzschlitzschraubendreher die acht Halteschrauben lösen. Achtung: Eine Schraube ist etwas kürzer. Merken, wo sie hingehört.
4. Gerät wieder normal vor sich hinstellen. Aber aufpassen, das Oberteil ist jetzt gelöst!
5. Oberteil links und rechts anfassen und vorsichtig etwas anheben. Durch eine leichte Rechtsdrehung Oberteil vorne an den Tasten des Kassettenrecorders vorbei hochheben. Weiter vorne hochheben, mit leichtem Druck nach hinten, sodaß die hinteren Taster nicht beschädigt werden. Oberteil völlig abheben.
6. Jetzt das Flachbandkabel, das von der Tastatur zur CPU-Platine geht vorsichtig nach hinten abziehen. Darauf achten, daß die blanken Kabelenden nicht verbogen werden. Im allgemeinen dieses Kabel nicht unnötig oft bewegen, da die Kontakte zur CPU-Platine leicht brechen.
7. Sämtliche Schrauben der Tastaturplatine lösen und Tastatur zur Seite legen.
8. Die beiden Flachbandkabel, die von der Interface- zur CPU-Platine führen, lösen. Das unter 6. beschriebene beachten.
9. Die Spannungsversorgungsstecker (Farben: gelb, grün, rot, schwarz) von beiden Platinen lösen. Achtung: Aufpassen, wo die jeweiligen Stecker hingehören.
10. Wenn ein Lautsprecher eingebaut ist, auch diese Verbindung lösen. (geschieht durch Hochziehen der schwarzen Stecker ~~xx~~ nahe des Modulators). Antennenkabel vom Modulator lösen.
11. Die drei Halteschrauben der CPU- und Interfaceplatine lösen, und Platinen zur Seite legen.

Einbau der Joystickbuchse:

1. Geeigneten Platz für die Buchse wählen. (Ich habe dafür einen Platz unterhalb der Tastatur gewählt, um den Joystick bequem von außen anzuschließen (siehe auch Photo!)).
2. Die genauen Maße der Buchse nehmen und auf der entsprechenden Gehäusestelle anreißen.

3. In der Mitte des Anrisses ein Loch bohren und mit Rund- und Schlüsselfeilen den Ausschnitt auffeilen. Achtung: Lieber einmal zu viel, als einmal zu wenig die Buchse anpassen!
4. Wenn die Buchse sauber im Ausschnitt sitzt, wird die Buchse mit Pattex oder Zweikomponentenkleber festgeklebt.

Verdrahtung:

Jetzt stehen zwei Möglichkeiten zum Anschluß zu Verfügung.

1. Möglichkeit: Man lötet die Kabel an die im Schaltbild gekennzeichneten Stellen auf der Tastaturplatine fest, wobei die Zahlen 1, 2, 3, 4, 5 und 19 der Belegung des Tastatursteckers entsprechen (Draufsicht: 1, 2, 3 ... 18, 19, 20). Die Verdrahtung geht aus der umseitigen Zeichnung hervor.
2. Möglichkeit: Auf der Tastaturplatine findet man in der rechten oberen Ecke 12 Bohrungen. Hier lötet man nun eine 'halbe' 24-Pin-IC-Fassung ein. Nun werden die verzinnten Kabelenden in die auf dem untenstehenden Schaltbild angegebenen Kontakte der IC-Fassung gesteckt (siehe auch Photo2). Vorteil: Das Kabel kann jederzeit herausgenommen werden!

Jetzt wird das GENIE wieder zusammengebaut. Dabei wird in der umgekehrten Reihenfolge vorgegangen.

Zur Funktionsüberprüfung wird folgendes Programm eingetippt:

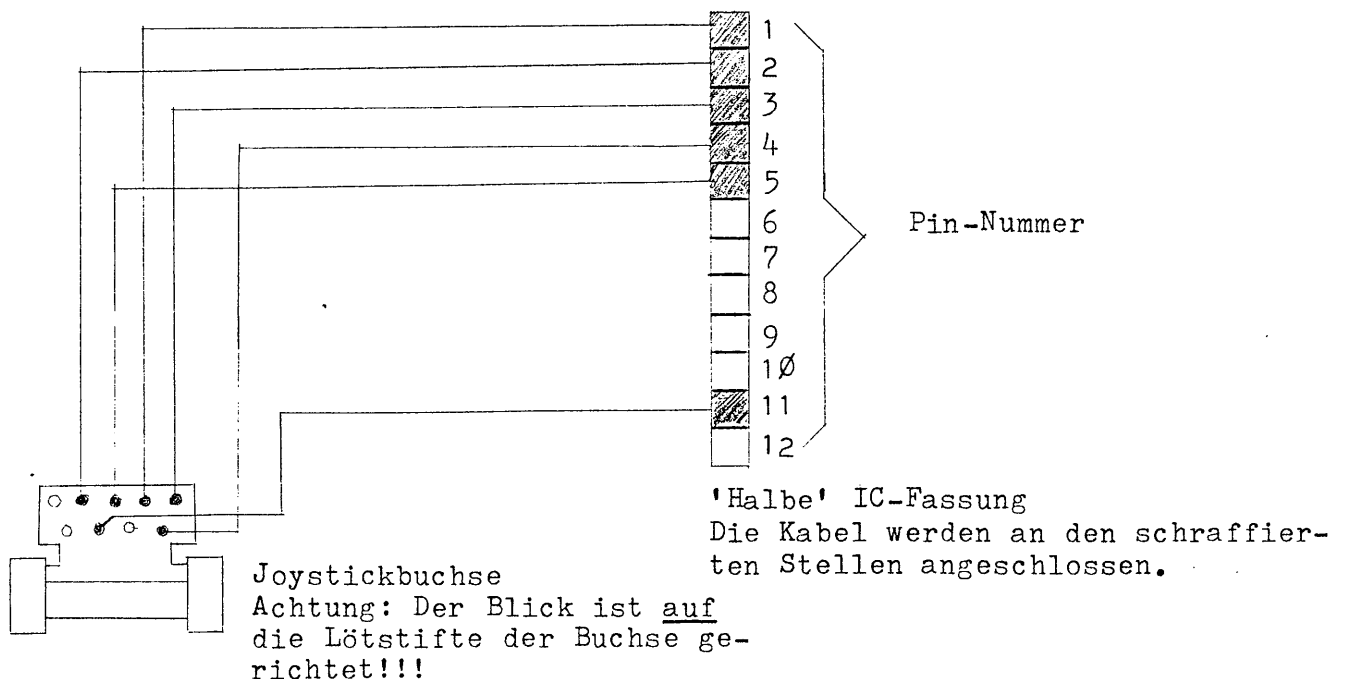
```

10 CLS
20 AS=INKEY$
30 IF AS=CHR$(91) PRINT"OBEN"
40 IF AS=CHR$(10) PRINT"UNTEN"
50 IF AS=CHR$(8) PRINT"LINKS"
60 IF AS=CHR$(9) PRINT"RECHTS"
70 IF AS=CHR$(32) PRINT"FEUER"
80 GOTO 20

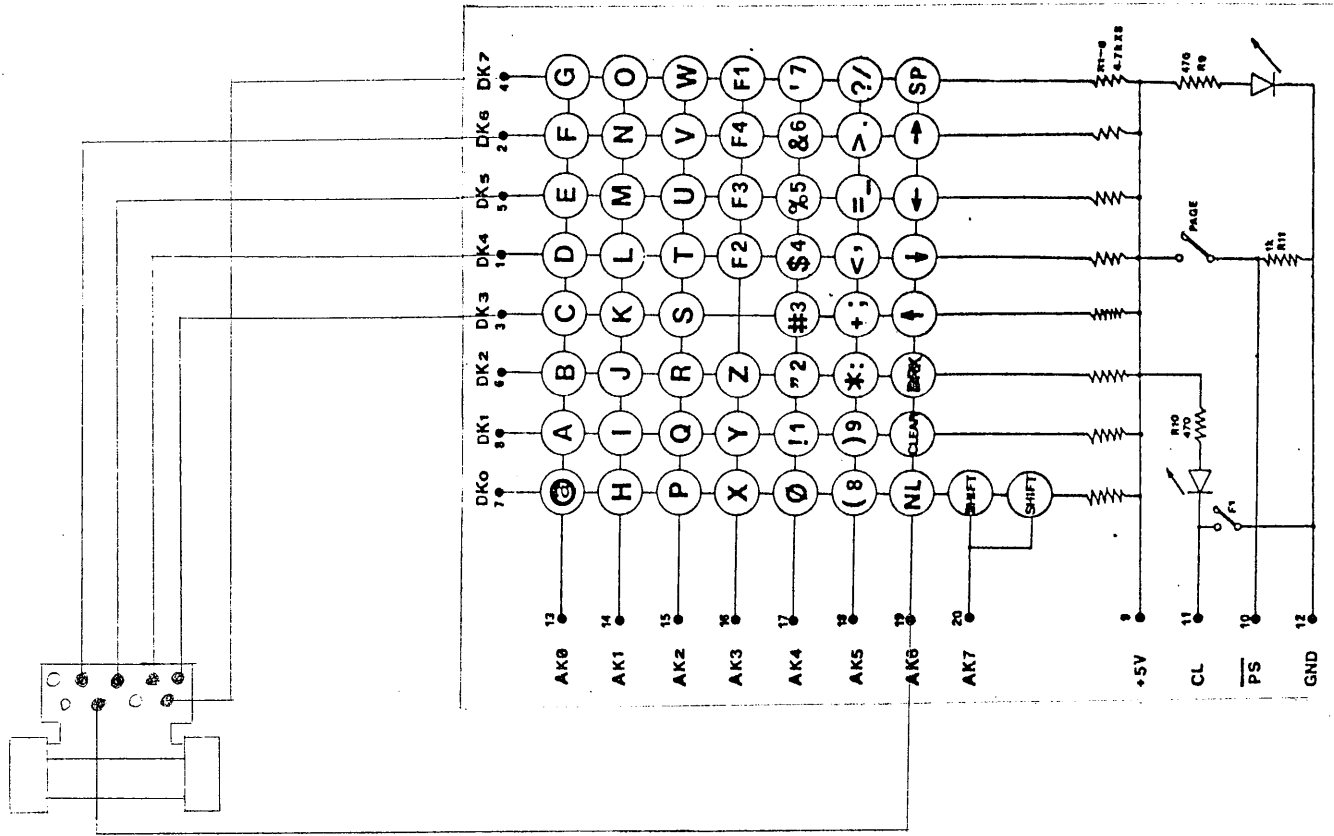
```

Je nachdem in welcher Richtung der Joystick nun bewegt wird, muß auf dem Bildschirm die entsprechende Richtung angezeigt werden.

Anschlußplan für Möglichkeit 2:



Anschlußplan für Möglichkeit 1:



Joystickbuchse

Achtung: Der Blick ist auf die Lötstifte der Buchse gerichtet!!!

Photo1

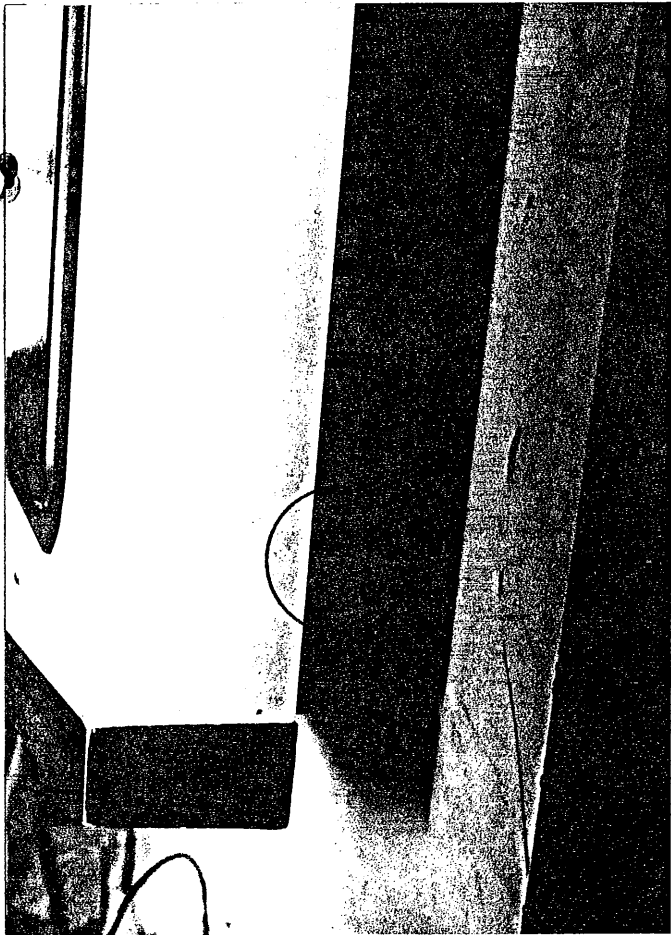
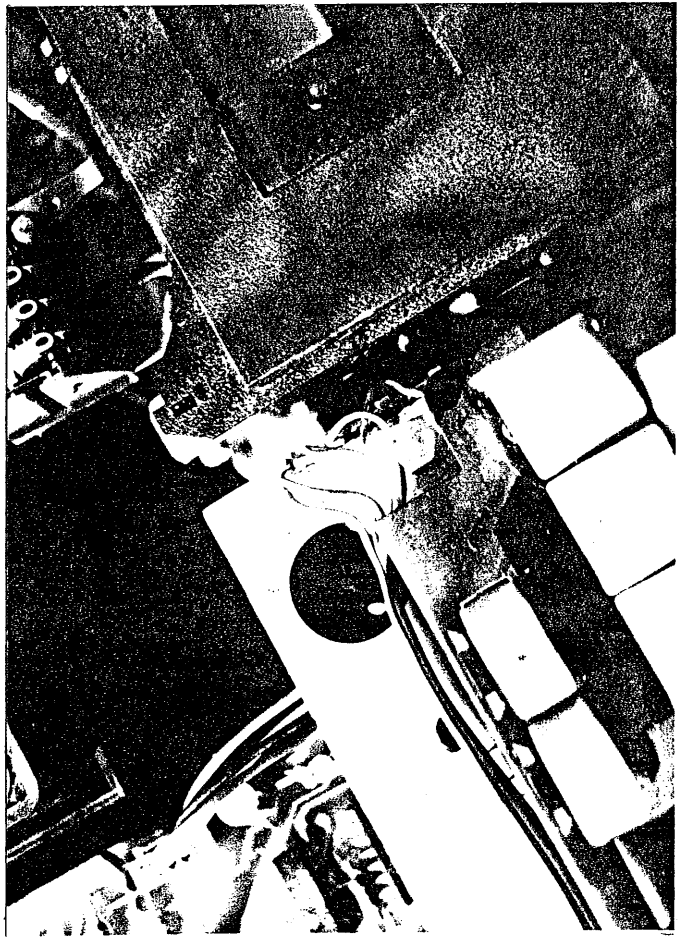


Photo2



ADRESENLISTE

NAME	VORNAME	M-NR.	TELEFON	STRASSE	ORT	HARDWARE
ALTHAUS	THOMAS	840441	0551/75913	WESERSTR. 35	3400 GOETTINGEN	CG,CR
BACH	SIGGI	830611	???	LEHMDENERSTR. 54	2845 DAMME 2	GENIE I, TYPENRAD
BIEWALD	MARKUS	830418	0421/471829	GESCHWISTER-SCHOLL STR 105	2800 BREMEN 41	CG, CR
ASCHEK	MANFRED	840120	0222/6400483	INZERSDORFERSTR. 111/8/9	A-1100 WIEN	TRS 1,CR
ALBUSCH	HANS-PETER	830303	0471/58206	JACOBISTR. 32	2850 BREMERHAVEN	CG, CR, LP ?
FISCHBECK	UWE	840125	04421/34282	FRIEDERIKEN- 17	2940 WILHELMSHAVEN	CG,CR
FOLKERTS	RALF	840110	04223/1282	NUTZHORNSTR. 9	2875 BOOKHOLZBERG	GENIE I, 1LW, LP NEC8023
FREY	WOLFGANG	830816	040/6958854	PILLAUERSTR. 135	2000 HAMBURG 70	GENIE I, LW
GRAJENSKI	WERNER	830507	02134/54573	ZEDERNWEG 29	4220 DINSLAKEN	GENIE I
GRUNDMANN	WALDEMAR	830815	0441/36218	BEVERBAEKSTR. 46	2900 OLDENBURG	TRS80 I,CR, LW
JERMANN	MARKUS	840127	05141/31133	LUENEBURGER HEERSTR. 47	3100 CELLE	GENIE 1,CR,HIRES
KARNATZ	MICHAEL	830419	04421/53936	SCHWERINER RING 23	2940 WILHELMSHAVEN	CG,CR,LPGEMINI 10X, 1LW
KLEIN	GERHARD	840234	040/513159	CARL-COHN-STR. 73	2000 HAMBURG 60	TRS80 M1, 1LW, LP MX80FT
KROEHER	PAUL	831023	---	GRAF-ENNO-STR. 7	2970 EMDEN	GENIE I, 1LW
KRZYZANOWSKI	PROF.DR. JER	840233	---	NUR ÜBER BETREUER	ERREICHBAR !	GENIE I, LP ?
KUMMEROW	PETER	840132	0431/30647/3	STEENBECKERWEG 8/35	2300 KIEL	GENIE I, 2LW(40,80TKS), LP10S44
KUMMEROW	JENS	840336	---	HAUPTSTR. 4	5412 HUNSDORF	GENIE I, LP DP510, 2LW80DSDD
KUTTER	WOLFGANG	830505	08370/1268	ILLERSTR. 18	8961 WIGGENSBACH	CG,CR,LP STAR510
LINNEWEBER	MANFRED	831224	0471/25453	AUF DER BRIGG 15	2850 BREMERHAVEN	TRS80 III, LP MX80FT
MAY	HOLGER	830508	02935/1668	MARIENSTR. 9	5768 SUNDERN 2	GENIE I, 1LW NEWDOS, FERNSCHRE
MEIER	HANS-CHRISTI	840126	04421/64577	RAABESTR. 42	2940 WILHELMSHAVEN	CG, LP GEMINI 10X
MASREITER		840339	08431/		8858 NEUBURG/DD.	KOMTEK, LW, LP
MEYER	FRANCISCO	840337	---	W. BESSONSTR. 5	7750 KONSTANZ 16	CG, LP OLIVETTI PRAXIS
MICHELSDORFER	WOLFGANG	840129	08431/7846	MARIENBADERSTR. 21	8858 NEUBURG/DD.	TRS80 1, 2LW(40/80SP), CR, LP ITOH 8510
RING	RUDOLF	840104	0208/53359	DUISBURGERSTR. 445/304	4330 MUELHEIM/R.	CG, CR
RUETTIGERS	MARTIN	830922	---	EIFELSTR. 85 A	5190 STOLBERG-VICHT	GENIE I, LP STAR
SCHMIDT	KLAUS	830301	0471/24998	BLESSMANNSTR. 1 B	2850 BREMERHAVEN	APPLE
SCHMIDT	HORST	830302	0471/414611	KOERNERSTR. 7	2850 BREMERHAVEN	GENIE II, CR
SCHMITZ	PAUL-JUERGEN	840235	0202/401192	HÄHNERBERGERSTR 111	5600 WUPPERTAL 12	GENIE 1, CR, LP BROTHER CE60, 2 LW
SCHNEIDER	HANS-DIETER	830621	---	POSTFACH 1346	2943 ESENS	ABC80, CR, LP MX80FT
SOPP	ARNULF	840131	0451/791926	WAKENITZSTR. 8	2400 LUEBECK 1	GENIE 1, 2 LW, LP GEMINI 10X
STARK	OTHMAR	840340	082236/81180	SCHILLERSTR. 112	A2340 MOEDLING	GENIE I, 3LW, LP MX80FT
Spieß	Peter	*80417	08434/454	Trugenhofenerstr. 27	8859 Rennertshofen 1	GENIE II, 2 LW, LP NEC 8023
THALMEIER	GREGOR	840128	08091/9085	POSTFACH 1140	8011 KIRCHSEEON	TRS80 1, 3LW(DD, DS), LP MX80, M
THOENNISSEN	HEINRICH	830306	0421/647762	GRAMBKEMOORER LANDSTR. 6	2800 BREMEN 77	TRS80 I, 2LW, CR, LP MX80FT
THOM	HARALD	840112	0203/337178	NECKARSTR. 9	4100 DUISBURG 1	CG, CR
V. SCHEID	UWE	830509	0471/85418	STROEDACKER 45 C	2850 BREMERHAVEN	TRS80 ?, 1LW
VAN DER MEULEN	KLAUS	840338	0201/678338	ALTENDORFERSTR 502	4300 ESSEN 11	TRS1, 2LW, LP TYPENRAD
VAN DER TOUW	WILLEM G.	840130	004117805421	TOBELRAINSTR. 2	CH-8820 WAEDENSWIL	GENIE 3, LP ITOH F10-40
VOLLMER	TORSTEN	830614	---	RHEINSTR. 42	2850 BREMERHAVEN	CG, CR

FLOHMARKT

24

VERKAUFE - VERKAUFE - VERKAUFE

Schreibmaschine: Olivetti Praxis 35 mit Interface,

Anschluß direkt an Centronics-Schnittstelle,
Textverarbeitung für TRS-80 und Video Genie
(SCRIPSIT und SUPERSCRIPSIT) vorhanden.
(Dieser Text wurde mit SCRIPSIT geschrieben.)

Schriftprobe: abcdefghijklmnopqrstuvwxyzäöü
ABCDEFGHIJKLMNPOQRSTUVWXYZÄÖÜ
1234567890
;°£\$µ()_ *²³ Å"=%&§/:`´B^+O,.-?!'|å

VB: 900 DM (Schreibmaschine mit Koffer, Interface, 2 Farbbänder)

Klaus van der Meulen,
Altendorfer Str. 502
4300 Essen 11
Tel.: 0201/670338

(siehe auch Bericht von Christoph Wachendorf von Ausgabe 12
1. Jahrgang)

VERKAUFE - VERKAUFE - VERKAUFE

==> Bei der Betreuungsadresse gibt es die Möglichkeit, alte Farbbandkassetten mit neuem Farbband auffüllen zu lassen. Es wird garantiert neues Material verwendet (kein wiederauffrischen der alten Farbbänder). Die Standardfarben sind schwarz, blau und braun (Lieferzeit ca. 14 Tage). Auf Wunsch gibt es noch die Farben grün, rot und silbergrau (Lieferzeit ca. 4 Wochen).

Preise:

	ITOH/NEC	EPSON	EPSON
	8510/8023	MX 80	MX 100
Schwarz	: 8,50	14,--	21,--
Braun,blau	: 12,--	20,--	30,--
Sonderfarben	: 18,--	30,--	45,--

Alle Preise incl. Mwst. + Porto und Verpackung. Bei Bestellung bitte alte Kassette mitschicken. Andere Druckertypen auf Anfrage.

Auf Grund der Anzeige im letzten Info kann ich nun folgende Adressen von Firmen, die RS-232 Schnittstellen verkaufen bekanntgeben:

TROMMESCHLAGER, Kölnstr. 4, 5205 St. Augustin 2	195,-DM
RB Elektronik-Vertrieb, Bouraueler Str. 13, 5208 Eitorf	199,-DM
Schmidtke Elektronik, Sandkaulstr. 84, 5100 Aachen	255,-DM
Geissler Elektronik, Im Holdental 12b, 7632 Friesenheim	250,-DM

Vor einer Bestellung empfehle ich aber, zunächst mehr Informationen von den Firmen einzuholen.