

CLUB 80

Das Computerleben...

Clubinfo

der

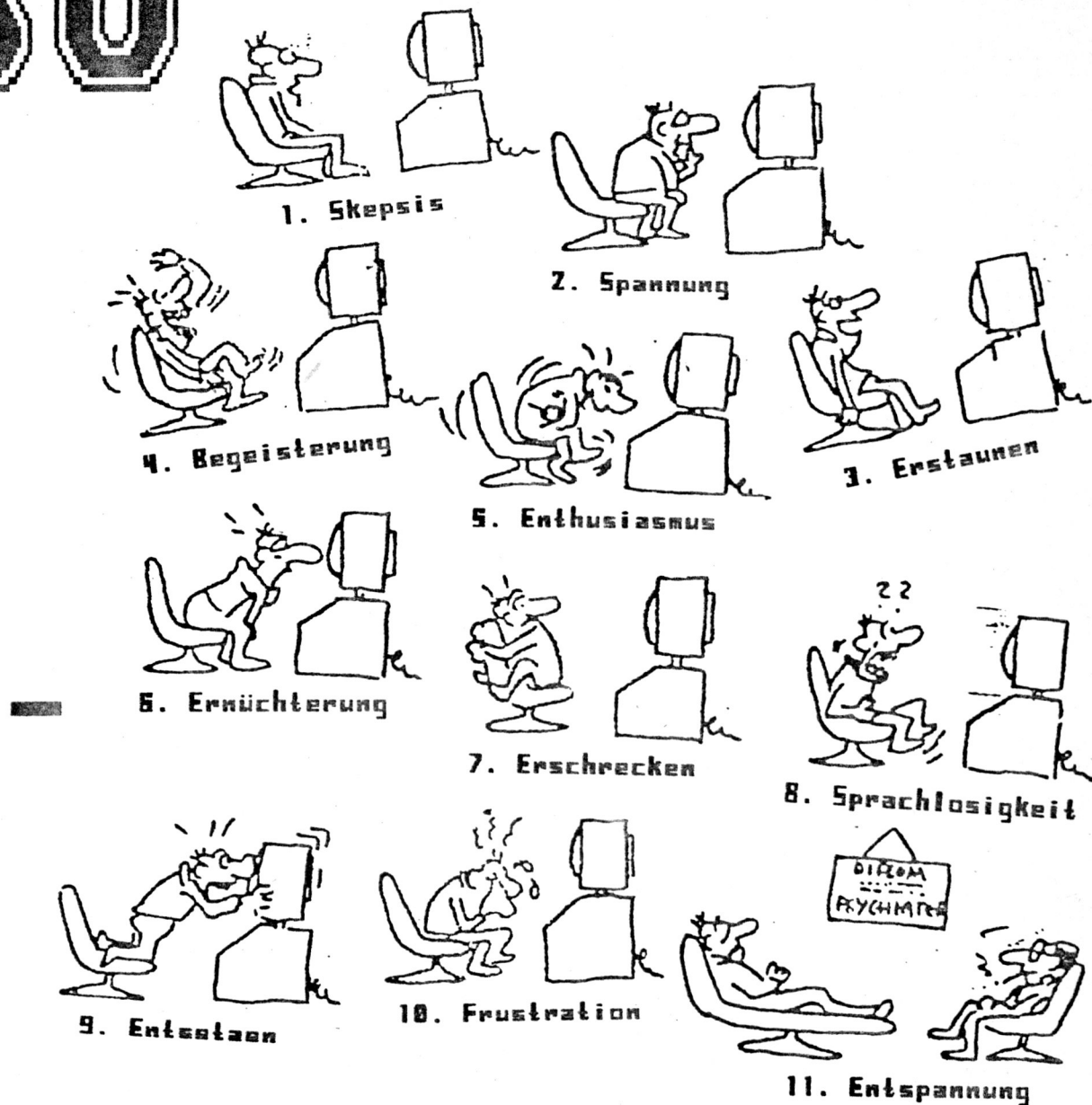
TANDY

GENIE

und KONTEK

ANWENDER

9. AUSGABE



Kontaktadresse: CLUB 80 / Günther WAGNER / Gartenstraße 4 / 8201 Neubarnau

Tel.: 08835/3351 (18 - 20 Uhr)

...ist nicht einfach !!!

Inhaltsverzeichnis

Seite:

Clubinternes

Der Vorstand informiert	01 - 02
Die Neuen	01

Software

Festlegung von ORG	03
Directory lesen	04
Editor	05 - 08
LHOFFSET mit DISABLE DOS	09 - 10
Lprint alles einfacher	11
PDRIVE-Angaben	12
VIDHEK mit HRG	13 - 17
Unbegrenztes Spielvergnügen	18
Am Bildschirm definiert	19 - 22
Adventure - Ecke	23 - 24
Basic - Dialekte	25 - 32
Zeichencodes	30
BASIC - Wörtervergleich	33 - 36

Hardware

LCB - Bus System	39 - 44
Testadapter	45 - 46
Relais - Interface	47 - 52
Billiges CP/M.	53 - 62
Extra RAM	63 - 66
Test -- Genie IIs	67 - 68

Börse

Wer hat was ???--Wer sucht was ???	71 - 72
Fragen, Fragen, Fragekasten	73 - 76

Sonstiges

Berichtigungen	37 - 38
Nachlese	69

Programmhinweise

Neue Programme	77 - 80
--------------------------	---------

Die letzten Seiten

Clubmitgliederadressen	82
Impressum	83
Schluß	84
Beilageblätter	am INFO-Ende

Liebe Mitglieder,

wenn ich diese Zeilen schreibe, habe ich zwar das Info noch nicht gesehen - aber ich habe mir vom Jens kurz erzählen lassen, was so alles erscheinen wird. Es ist demnach ein sehr gutes Info sowohl hinsichtlich Qualität als auch Umfang! das freut mich sehr - zeigt es doch davon, daß alle Mitglieder an einem guten Info interessiert sind.

Trotzdem möchte ich eine 'Nachlese' zum 8. Info bringen. Mir (und wahrscheinlich auch vielen von Euch) hat es gut gefallen; weniger interessant dürfte es für den Hartmut gewesen sein - er hat nämlich etwa 15 Seiten des Clubinfo bestritten! Es war also insgesamt relativ wenig Neues für den Hartmut dabei. Da wird ihm dieses 9. Info schon wieder besser gefallen. Ich wollte damit nur gesagt haben, daß es wichtig ist, daß sich viele Mitglieder am Clubinfo beteiligen - ich möchte damit vor allem die Mitglieder ansprechen, die sehr wenig bzw. nichts für das Info schreiben, obwohl Sie einiges auf dem Kasten hätten.

Nun zur Clubkasse: Wir haben zur Zeit ein Guthaben von ca. 660 DM; am 19. August waren folgende Beträge noch ausständig:

Jeweils 10 DM Nachzahlung von:

Troesch, May, Wies, Hummel, Bozek, Grajewski, Boecker,

Baldes, Preuss, Smerling, Retzlaff, Fröhlich

6 DM (für Info 1 und 3) von Alber

Für mich ist dies der einfachste und billigste Weg, die Mitglieder an Ihre Zahlungen zu erinnern - ich glaube (bzw. hoffe), daß sich durch diese Art und Weise niemand auf den Schlies getreten fühlt. Ferner ist ganz klar, daß sich Zahlungen von Mitgliedern mit Erscheinen dieses Info's bereits erledigt haben können!

An dieser Stelle möchte ich auch unserem Mitglied Peter Spiess danken, der für einen sehr günstigen Preis das Info umdruckt. Dadurch konnte ich seit dem 7. Info eine erfreuliche Reduzierung der Kosten feststellen.

Mit dem Info bekommt Ihr auch ein Werbeblatt für den CLUB B0. Wer Gelegenheit hat, sollte diese an günstigen Stellen (z.B. Schulen, Universitäten, Computergeschäften etc.) aushängen oder an ev. interessierte Personen weitergeben. Wer weitere Werbeblätter benötigt, kann diese bei mir anfordern (in den Größen DIN A3 - DIN A6). Weitere Werbung für den CLUB B0 wird hoffentlich in den nächsten Monaten in einschlägigen Computer-Zeitschriften erscheinen. Ich habe die entsprechenden Verlage darum gebeten.

Nun möchte ich noch unser jährliches Clubtreffen ansprechen; schön langsam muß man sich da wieder an die Planung machen. Ich denke an folgende Termine:

22./23. März
05./06. April
12./13. April

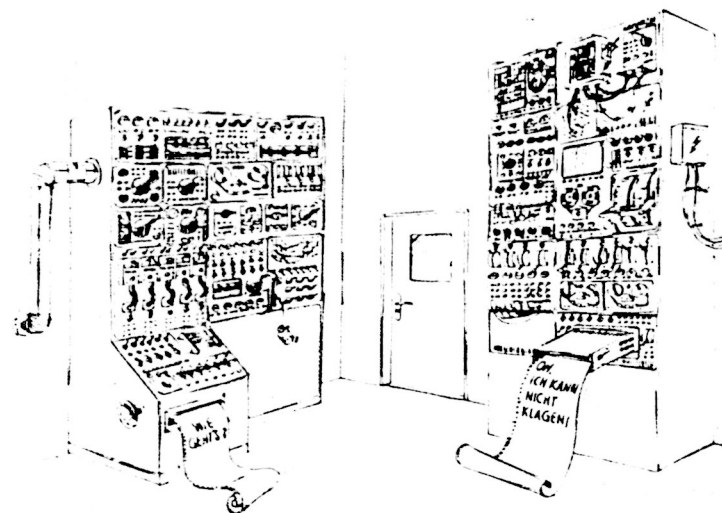
Das Treffen sollte meiner Meinung nach wieder im Großraum Frankfurt/Wiesbaden stattfinden. Bitte schreibt mir, an welchem Wochenende es bei Euch besonders gut oder schlecht geht und was Ihr sonst noch zum Thema Clubtreffen anzusprechen habt. Ansprechen möchte ich noch die Möglichkeit, regionale Clubtreffen abzuhalten. Dies muß aber von den Mitgliedern selbst ausgehen. So möchte ich ganz gerne ein kleines Treffen im November in München abhalten (1 Nachmittag oder so). Wer Interesse am Erscheinen hat, sollte sich bitte bei mir melden.

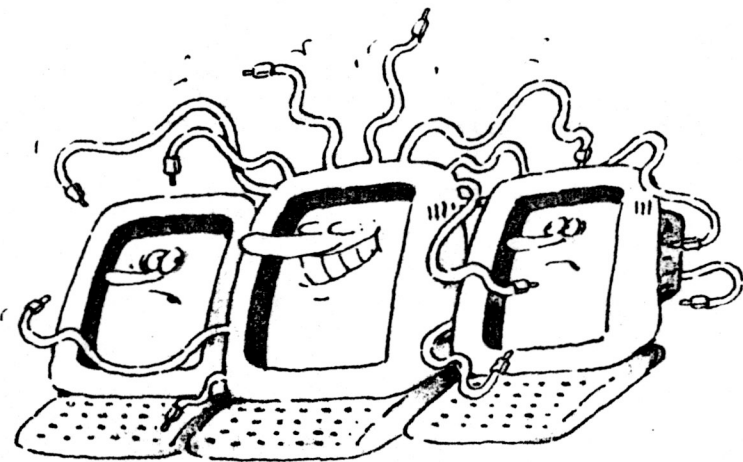
Jedes Mitglied erhält mit diesem Info einen Abdruck seiner abgespeicherten Daten. Ich vermute, daß einige dieser Daten bei vielen Mitgliedern nicht mehr stimmen. Der eine hat ein weiteres Floppy, der andere eine HRG usw. Bitte überprüft diese Ausdrücke und falls es was zu ändern gibt, so informiert mich bitte unbedingt bis Ende September. D A N K E !

Hoffentlich habe ich Euch mit diesen Zeilen jetzt nicht zu sehr gelangweilt. Viel Freude mit dem 9. Info wünscht Euch allen (und auch mir selbst)

Euer

Gunter Dagner





HALBAUTOMATISCHE FESTLEGUNG VON ORG =====

Bei der Programmierung in Assembler steht man öfters vor dem Problem, den Beginn des Programms so festzulegen, daß es genau an einer bestimmten Stelle endet.
Die Lösung dieser Probleme kann man auf einfache Weise auch dem Assembler überlassen:

```

TOPMEM EQU 0FFFFH ;oder ein anderer Wert
PRGLEN EQU 0000    ;zunächst irgend ein beliebiger Wert
ORG TOPMEM-PRGLEN
START ;des Programms
      'Programm'

      'letzte Anweisung des Programms'
RAMUSE EQU $-START-1 ;ergibt Programmlänge in HEX
END START ;Programmende
  
```

Nun kann man das Programm assemblieren (ohne Abspeichern!), wodurch der Assembler den Wert RAMUSE berechnet. Diesen braucht man dann nur in die PRGLEN-EQU-Anweisung einzusetzen und hat nun für den endgültigen Assemblerlauf das korrekte ORG.

Josef Konrad

Hallo liebe Club-Freunde,

dieses Mal sollt Ihr auch wieder etwas von Euerem Vorstand zu lesen bekommen. Ich habe gerade etwas Zeit und möchte Euch auch als 'gutes Vorbild' vorangehen:

Schreibt doch ruhig auch mal Eueren kleinen und großen Kniffe. Bestimmt ist dieser Artikel für viele kalter Kaffee - aber bestimmt auch für manchen von Interesse. Also - laßt Euch nicht lumpen und haut in die Tasten!

Directory lesen

Euer
Günter Rognier

So - nach der Vorrede aber nun zum Programm. Oft möchte ich in BASIC-Programmen das Directory in Variablen einlesen und damit arbeiten. Nach einer Vorlage aus einer alten CHIP entstand dabei folgendes Unterprogramm, das sich in jedes BASIC-Programm einbauen läßt:

```

1000 ON ERROR GOTO 1130
1010 DR$="DIR/SYS:"+RIGHT$(STR$(DR),1)
1020 OPEN"R",1,DR$
1030 FOR I=3 TO LOF(1)
1040 GET 1,I
1050 FOR II=0 TO 7
1060 FIELD 1,(II*32) AS D$,1 AS A$,4 AS D$,8 AS FS$,3 AS FE$
1070 IF NOT (CVI(A$+CHR$(0)) AND 208)=16 THEN 1100
1080 A$(J)=FS$+" "+FE$
1090 J=J+1
1100 NEXT II,I
1110 CLOSE
1120 RETURN
1130 RESUME NEXT
  
```

DR = Directory im Drive DR wird gelesen (muß im Hauptprogramm festgelegt werden)

FS\$ = Dateiname bis zum Schrägstrich (/)

FE\$ = Namenserverweiterung nach dem Schrägstrich (z.B. BAS, CMD)

Selbstverständlich muß das Feld A\$ ausreichend dimensioniert werden und genügend Speicher zur Verfügung gestellt werden.

Mögliche Anwendungen gibt es genügend für diese Routine. Man kann z.B. damit seine ganzen Disketten katalogisieren und zu jedem Programm noch eine kurze Erklärung abspeichern oder man kann sich in einem Programm alle Dateien mit der Namenserverweiterung DAT ausgeben lassen. Man kann auch das Directory alphabetisch sortieren lassen und noch vieles mehr.

E D I T O R

=====

Das hat nichts mit einem Konditor zu tun, sondern so nenne ich ein Unterprogramm, welches eine Bildschirmorientierte Eingabe ermöglicht. Mit Hilfe dieses UP kann man in einer Bildschirmmaske beliebig von Eingabe zu Eingabe springen, innerhalb einer Eingabe sich bewegen, Zeichen einfügen und löschen, etc. Erst zum Schluß werden dann die Eingaben direkt vom Bildschirm in die Variablen eingelesen. Ich benutze dieses UP schon seit ca. 2 Jahren. In diesen 2 Jahren wurde es laufend verbessert und ergänzt. Jetzt halte ich den Zeitpunkt gekommen, Euch dieses UP vorzustellen. Übrigens gibt es dieses UP jetzt auch in unserer Programmbibliothek (wer es z.B. nicht eintippen möchte). Die Zeilen 500-1150 des Listings gehören nicht zum UP selbst - sie sollten nur die Möglichkeiten des Editor demonstrieren.

PROGRAMMABLAUF:

- A) Zeile 100 sollte immer die 1. Anweisung im Programm sein. DEFINIT U definiert U für das Kopierfeld (siehe hierzu BASIC FASTER & BETTER, Seite 40 ff, 6. Clubinfo). Das Kopierfeld wird für löschen, einfügen und auffüllen mit '#' benutzt. Die DIM-Anweisung ist nur notwendig, wenn mehr als 10 Eingabefelder vorhanden sind.
- B) Im Hauptprogramm wird zunächst die Eingabemaske geladen; anschließend wird der Editor aufgerufen. Nach der Eingabe können die eingegebenen Werte (befinden sich in E\$()) weiter bearbeitet werden.
- C) Zeile 20030: G=0 bedeutet 1. Eingabezeile
- D) Zeile 20040: A erhält die Anfangsposition der 1. Eingabe
- E) Zeile 20050: B liest das Zeichen an der Bildschirmposition A
- F) Zeile 20060: Ausgabe des 'Cursor'
- G) Zeile 20070: Tastaturabfrage
- H) Zeile 20080 - Zeile 20220: Abarbeitung der Eingaben/Befehle (weiteres siehe unter Befehle); Verwaltung von A und G; in Zeile 20210 muß das GOTO-Statement für den Befehl 'SHIFT V B' an das jeweilige Programm angepaßt werden.
- I) Zeile 20230: keine gültige Eingabe; Rücksprung nach 20070
- J) Zeile 20260-20320: Einlesen der Eingaben direkt vom Bildschirm (mittels PEEK). Die Eingaben werden in E\$() abgespeichert. Wird ein '#' eingelesen, so wird das Ende der Eingabe angenommen. Das Einlesen der Eingaben dauert einige Sekunden. Das UP wird mit RETURN abgeschlossen
- K) Die Zeilen 21000 - 24050 dürften verständlich sein.
- L) Zeile 50030: Eine der beiden Zeilen, die bei Einbinden in Programmen geändert werden muß. Nach DATA stehen jeweils die Bildschirmpositionen für Anfang und Ende der jeweiligen Eingabefelder. Die Werte müssen an die einzelnen Bildschirmmasken angepaßt werden. Anhand der Anzahl der eingelesenen Werte weiß der Editor auch, wieviele Eingabefelder vorhanden sind. Die beiden Nullen am Ende sind unbedingt notwendig. Sie signalisieren, daß alle Werte eingelesen wurden.

BEFEHLE:

- Cursor um 1 nach rechts; wenn Ende von Eingabefeld erreicht ist, wird die nächste Eingabe angesprochen
- Cursor um 1 nach links
- ⬆ Cursor entweder auf das vorliegende Eingabefeld o. innerhalb des Eingabefeldes eine Zeile höher (nur bei Eingabefeldern mit einer Mindestlänge von 64 Zeichen möglich)
- ⬇ Cursor entweder auf das nachfolgende Eingabefeld o. innerhalb des Eingabefeldes eine Zeile tiefer
- SHIFT ➤ Sprung auf Ende des Eingabefeldes
- SHIFT ➤ Sprung auf Anfang des Eingabefeldes
- SHIFT ⬆ Sprung auf Anfang des 1. Eingabefeldes
- SHIFT ⬆ D Sprung auf Anfang des letzten Eingabefeldes
- SHIFT ⬆ E Einfügen innerhalb des Eingabefeldes
- SHIFT ⬆ L Löschen innerhalb des Eingabefeldes
- SHIFT ⬆ B Abbruch und Rücksprung
- ⓐ Auffüllen des Eingabefeldes mit '#' von Cursor-Position bis Ende des Eingabefeldes; Sprung auf Anfang des nächsten Eingabefeldes
- ENTER Sprung auf Anfang des nächsten Eingabefeldes
- CLEAR Einlesen der Eingabefelder in E\$(); Ende des UP

VARIABLENLISTE

- A Bildschirmposition
- B Inhalt der Bildschirmposition A
- EA() Anfang einer Eingabezeile
- EE() Ende einer Eingabezeile
- F Anzahl der Eingabefelder
- G gegenwärtiges Eingabefeld
- I FOR-NEXT-Schleife
- US Variable für Kopierfeld
- A\$ Inhalt der Tastaturabfrage
- E\$() Inhalt der Eingabezeilen

Ich hoffe, daß dieser Artikel für den Einen oder Anderen interessant war. Das UP ist ganz einfach - nur die Erklärung fällt mir schwer. Sollten noch Fragen zu diesem Artikel auftauchen, so meldet Euch bitte bei mir.

Günther Wagner

```

100 CLEAR 5000: DEFINT U: DIM EE(15), EA(15), E$(15): GOSUB 21000: GOSUB
23000
500 REM -----
510 REM *** HAUPTPROGRAMM
520 REM -----
550 GOSUB 1000
560 GOSUB 20000
570 CLS: STOP
1000 REM -----
1010 REM *** EINGABEMASKE AUSGEBEN
1020 REM -----
1030 CLS
1040 PRINT "Anrede:      *****"
1050 PRINT "Vorname:      *****"
1060 PRINT "Familiennamen: *****"
1070 PRINT "Str. e Nr.:      *****"
1080 PRINT "PLZ Wohnort:    *****"
1090 PRINT "Telefon:      *****"
1100 PRINT "Geb.-Datum:      *****"
1110 PRINT TAB(49) "Geschlecht:  #";
1120 PRINT "Beruf:      *****"
1130 PRINT "Bemerkung 1:  *****"
#####
1140 PRINT "Bemerkung 2:  *****"
#####
1150 RETURN
20000 REM -----
20010 REM *** EDITOR
20020 REM -----
20030 G=0
20040 A=EA(G)
20050 B=PEEK(15360+A)
20060 PRINT @ A, CHR$(143);
20070 A$=INKEY$: IF A$="" THEN 20070
20080 IF A$="a" THEN PRINT @ A, "#";: US(1)=15360+A: US(3)=15361+A: US(5)=EE(G)-A-1: US(6)=-20243: GOSUB 22000: G=G+1: GOSUB 24000: GOTO 20040
20090 IF A$=CHR$(91) THEN PRINT @ A, CHR$(B): IF A-64<EA(G) THEN G=G-1: GOSUB 24000: GOTO 20040: ELSE A=A-64: GOTO 20050
20100 IF A$>CHR$(31) AND A$<CHR$(128) THEN PRINT @ A, A$: A=A+1: IF A<EE(G) THEN 20050 ELSE G=G+1: GOSUB 24000: GOTO 20040
20110 IF A$=CHR$(9) THEN PRINT @ A, CHR$(B): A=A+1: IF A<EE(G) THEN 20050 ELSE G=G+1: GOSUB 24000: GOTO 20040
20120 IF A$=CHR$(8) THEN PRINT @ A, CHR$(B): A=A-1: IF A>=EA(G) THEN 20050 ELSE G=G-1: GOSUB 24000: GOTO 20040
20130 IF A$=CHR$(10) THEN PRINT @ A, CHR$(B): IF A+64<EE(G) THEN A=A+64: GOTO 20050: ELSE G=G+1: GOSUB 24000: GOTO 20040
20140 IF A$=CHR$(5) THEN PRINT @ A, CHR$(B): US(3)=15359+EE(G): US(1)=15358+EE(G): US(5)=EE(G)-A: US(6)=-18195: GOSUB 22000: PRINT @ A, " ";: GOTO 20050
20150 IF A$=CHR$(12) THEN US(1)=15361+A: US(3)=15360+A: US(5)=EE(G)-A: US(6)=-20243: GOSUB 22000: GOTO 20050
20160 IF A$=CHR$(27) THEN PRINT @ A, CHR$(B): G=0: GOTO 20040
20170 IF A$=CHR$(4) THEN PRINT @ A, CHR$(B): G=F: GOTO 20040
20180 IF A$=CHR$(25) THEN PRINT @ A, CHR$(B): A=EE(G)-1: GOTO 20050
20190 IF A$=CHR$(24) THEN PRINT @ A, CHR$(B): A=EA(G): GOTO 20050
20200 IF A$=CHR$(13) THEN PRINT @ A, CHR$(B): G=G+1: GOSUB 24000: GOTO 20040

```

```

20210 IF A$=CHR$(2) THEN CLS: GOTO 570
20220 IF A$=CHR$(31) THEN PRINT @ A, CHR$(B): GOTO 20260
20230 GOTO 20070
20240 '
20250 '
20260 FOR G=0 TO F: E$(G)=""
20270 FOR I=15360+EA(G) TO 15359+EE(G)
20280 IF PEEK(I)=35 THEN 20310
20290 E$(G)=E$(G)+CHR$(PEEK(I))
20300 NEXT I
20310 NEXT G
20320 RETURN
21000 REM -----
21010 REM *** KOPIERFELD VORBEREITEN
21020 REM -----
21030 US(0)=8448: US(2)=4352: US(4)=256: US(7)=201
21040 RETURN
22000 REM -----
22010 REM *** KOPIERFELD AUSF HREN
22020 REM -----
22030 J=0: DEFUSR=VARPTR(US(0)): J=USR(0)
22040 RETURN
23000 REM -----
23010 REM *** WERTE F R EDITOR EINLESEN
23020 REM -----
23030 RESTORE: F=0
23040 READ EA(F), EE(F): IF EA(F)=0 THEN F=F-1: RETURN: ELSE F=F+1: GOTO 23040
24000 REM -----
24010 REM *** UP FUER RICHTIGE BESTIMMUNG VON G
24020 REM -----
24030 IF G<0 THEN G=F
24040 IF G>F THEN G=0
24050 RETURN
50000 REM -----
50010 REM *** DATA-WERTE FUER EDITOR
50020 REM -----
50030 DATA 15,20,79,99,143,163,207,237,271,301,335,348,399,409,447,448,463,493,527,704,783,960,0,0

```

Computer in Schweden

Schweden, das Land der »freigeibigen« Computer: In der schwedischen Stadt Göteborg erhielt ein Forstarbeiter einen Gehaltsstreifen über 7 790 Kronen. Das sind ungefähr 6 000 Mark – das Zehnfache seines gewöhnlichen Gehalts. Der ehrliche Arbeiter gab den Lohnstreifen an das Personalbüro zurück. Wieder trat der Computer in Aktion: Er schrieb einen neuen Streifen über 77 900 Kronen.

DO IT YOURSELF: LMOFFSET with DISABLE DOS

Ihr kennt bestimmt alle den berühmten Anhang, den LMOFFSET erzeugt: "DO THE FOLLOWING:" etc., um ins Level II-Basic zu gelangen. Erst durch SYSTEM und "/" läßt sich das Programm starten.

Bei meinem neuen Genie IIs ließen sich nun einige der alten Programme (ohne diesen Anhang) nicht starten, weil sie (so vermute ich jedenfalls) im DOS--Bereich etwas ändern. Also mußte das DOS "ausgeschaltet" werden. Das kann man mit LMOFFSET machen, indem man das Programm verschieben läßt und dann auf die Frage "DISABLE DOS" mit "Y" antwortet. Abgesehen davon, daß dies nicht bei allen Programmen klappt, müßte das der Hobby-Programmierer auch selbst können. Was macht LMOFFSET also?

Ganz einfach: wenn man vom Verschieben absieht, hängt LMOFFSET nur ein kleines Anhängsel an das Programm an. Dieses Anhängsel sieht im Prinzip so aus:

```

ORG      9000H      ;erste freie Adresse hinter dem
                   ;Programm
ENTRY:    EQU       5300H      ;Entrypoint des Programms
                   ;
START:    DI        ;Interrupts sperren
          LD        HL,TAB      ;Adresse der Tabelle (s.u.)
          LD        DE,42E9H    ;feste Adresse im DOS
          LD        BC,0006H    ;Länge der Tabelle
          LDIR       ;6 Byte von (HL) nach (DE)
          ;
          LD        HL,TEXT     ;Adresse des Textes
          CALL       4467H      ;Text bis 0DH auf Bildschirm
          ;
LOOP:     JP        LOOP       ;Endlosschleife
          ;
TAB:      NOP
          NOP
          NOP
          JP        ENTRY
          ;
TEXT:     DEFB      'DO THE FOLLOWING: ...'
          DEFB      0DH        ;Ende des Textes, wichtig!
          ;
END       START

```

Will man diesen Anhang selbst erstellen, geht man am besten so vor:

1. mit LMOFFSET Start- und Endadresse sowie den Entrypoint feststellen (aber das Programm nicht verschieben oder so etwas)
 2. den ENTRYPOINT für ENTRY einsetzen und ORG mit der Endadresse + 1 setzen
 3. Anhang assemblieren und abspeichern
 4. im DOS eingeben: LOAD,programm/CMD und LOAD,anhang/CMD
 5. DUMP,neuprog/CMD,start,ende,entry wobei start=Startadresse des Originalprogramms, ende=Endadresse des Anhangs und entry=Entrypoint des Anhangs (alles mit LMOFFSET erfahrbare)
- (Punkt 4 und 5 lassen sich auch mit einem geeigneten Monitor erledigen)

Das Programm wartet nun nach dem Aufruf auf <Break und Reset>. Durch die Kommandofolge "SYSTEM" und "/" wird die nach 4209H gespeicherte Befehlsfolge angesprochen, die das Programm startet.

Dieser Aufwand wäre ja eigentlich unnötig, denn das macht LMOFFSET automatisch. Doch nun wird es erst interessant!!!

Was, wenn LMOFFSET streikt?! Mir passierte es: ASYLUM II lädt direkt in den Bildschirmspeicher und was weiß ich wo sonst noch hin. LMOFFSET zeigte nicht einmal die drei oben angesprochenen Adressen.

Jetzt müssen wir auf das RAM als Arbeitsplatz verzichten und uns direkt auf die Diskette begeben. Natürlich mit dem von Hartmut Obermann modifizierten Superzap.

Man sollte vermuten, daß wir einfach den assemblierten Anhang hinter das Programm hängen können. Doch so einfach ist das leider nicht.

Beim Laden von der Diskette muß das DOS ja wissen, wohin es das Programm bzw. bestimmte Teile laden soll und wo der Entrypoint ist. Der Entrypoint ist leicht zu finden: wir suchen uns den letzten Sektor des Programms (nicht mit dem ersten hinter dem Programm verwechseln!!!) und suchen dort nach den Bytes "0202". Hinter diesen Bytes steht der Entrypoint, natürlich wieder in dieser verqueren Darstellung, so daß die beiden Bytes also vertauscht werden müssen. Das ist aber einfach zu erkennen, denn sonst würde meist eine Adresse im ROM herauskommen, an der der Start zumindest unwahrscheinlich ist.

Nach diesen vier Bytes "0202xxxx" endet das Programm. Da kann man soviel ändern, wie man will, nichts passiert.

Also muß unser Anhang die 0202... überschreiben. Dafür fügen wir dann am Ende unseres Anhangs ein: "0202yyyy", wobei yyyy der Entrypoint des Anhangs ist.

Wer meinen Vorschlag nun so versucht, bekommt einen "LOAD FILE FORMAT ERROR" (Ihr seht, das ich alle Fehler schon gemacht habe). Es fehlt noch was, und zwar die Ladeadresse des Anhangs aus unserem ORG. Wenn wir den Anhang auf der Diskette betrachten, finden wir das Fehlende: "01xx yyyy".

Die 01 teilt dem DOS mit, daß eine neue Ladeadresse folgt. xx gibt die Länge des Blocks an, yyyy die Ladeadresse. Da die 01 auch im Programm vorkommen kann, muß die Länge des Blocks genau auf die nächste 01 oder 02 (zum Starten) zeigen. Was allerdings mit Blöcken ist, die länger als FF Byte sind, kann ich nicht sagen. Für solche Feinheiten haben wir bessere Experten im Club (Arnulf und Hartmut dürfen sich aufgefordert fühlen, hierzu etwas zu schreiben). Glücklicherweise ist das Problem bei mir noch nicht relevant gewesen, denn einen so langen Anhang habe ich noch nicht fabrizieren müssen. Allerdings habe ich einen kleinen Trick benutzt: im oben gezeigten Programm müßte der Text 64x16 Zeichen lang sein, damit der Bildschirm voll beschrieben wird. LMOFFSET benutzt im Original-Anhang noch eine zweite Blockübertragung, um den Bildschirm zu löschen. Doch das läßt sich einfacher erreichen. Am Anfang des Textes fügt Ihr ein:

```

DEFB      1CH
DEFB      1FH

```

was den Bildschirm löscht. Um ein Return (Enter, Newline) zu erzeugen, muß im Text

```

DEFB      0AH

```

auftauchen. Schon schrumpft der Text zusammen.

Nun könnt ihr herrlich auf der Diskette rumschrauben und LMOFFSET zum alten Eisen werfen, denn die erste Ladeadresse des Programms befindet sich natürlich am Anfang nach 01xx (es müssen nicht die ersten Byte im ersten Sektor sein!) und die Endadresse könnte man sich ausrechnen. Und wer eine neue Ladeadresse haben will: verschieben dürfte mit LDIR kein Problem sein. Doch dieses <Break und Reset> ist doch immer etwas umständlich. Was erreicht man eigentlich damit? Letztlich nichts anderes als ein System ohne DOS, aber mit dem Programm im Speicher. Wenn Ihr Euch nun das gute alte ROM-Listing anschaut und bei 0000H anfangt, seht Ihr die Befehle, die die DOS-Vektoren zurücksetzen und das Level II-Basic initialisieren. Viele Programme benötigen bestimmt nicht eine volle Initialisierung, sondern nur das Zurücksetzen der DOS-Vektoren und einige andere Kleinigkeiten. Seht Euch die Befehle an und schreibt den Anfang ab (0674H-0689H, 0075H ff.). Dann noch ein Jump zum Entrypoint (ohne <Break und Reset>). Das bringt eventuell die gleichen Ergebnisse wie die ursprüngliche Prozedur. Versprechen kann ich nichts, aber bei einigen Programmen lohnt es sich bestimmt!

Gerald Schröder

Ach, wie hüft mir das Herze lustig in der Brust! Da ist schon wieder einer ins Lager der Maschinsprache-Maniacs konvertiert. Lieber Hartmut, Deine Routine, mit der sich beliebige Codes mit LPRINT auf den Drucker ausgeben lassen, funktioniert einwandfrei. Wenn ich ihr jetzt eine Alternative entgegensetze, so geschieht das nur aus missionarischem Ehrgeiz. Die logische Hälfte des Assembler-Programmiers hast Du zwar schon richtig gut drauf, aber hier folgt ein wenig zur ökonomischen Hälfte:

Die Lösung, mit den Indexregistern zu arbeiten, liegt nahe, wenn ein Byte von einem direkt greifbaren Speicherplatz (hier (SP)) ein Stückweit entfernt ist. Aber das kostet Platz. Die indizierten Adressierungen erfordern nicht nur ein zusätzliches Switchbyte (DD bzw. FD), sondern manchmal sind auch Umwege nötig. So mußt Du z. B. den sonst unbenutzten Speicherplatz 4018h im Tastatur-DCB als Zwischenspeicher hernehmen.

Eine genauere Untersuchung des Interpreters hilft weiter: Das Registerpaar HL wird in den Treibern beim Einsprung nur als Adressenhalter benutzt; mit JF (HL) werden sie adressiert. Einmal im Treiber angelangt, ist HL frei. Also holen wir uns das zu druckende Byte folgendermaßen:

0000	211101	00001	LD	HL, 0111h	; Summand, MSB 1 zu hoch
0003	25	00002	DEC	H	; jetzt HL = 0011h
0004	39	00003	ADD	HL, SP	; Stelle im Stack errechn.
0005	7E	00004	LD	A, (HL)	; Byte aus dem Stack laden
0006	C3B405	00005	JP	05B4h	; und ausdrucken lassen

Die ersten beiden Befehle erscheinen widersinnig oder zumindest unständlich. Wozu HL mit einem Wert laden und den dann sofort im MSP (H) verändern? Der Ladebefehl LD HL, 0011h (Codes dez. 33, 17, 0) hätte eine logische 0 enthalten. Die hätte der Interpreter aber mitten in der Stringvariablen LN\$ als Zeilenende-Flag erkannt. Das wäre tödlich gewesen. Daher mußte dieser kleine Kunstgriff die 0 vermeiden.

HL wird mit dem Stackpointer addiert, so daß die Summe (die Adresse des Bytes) in HL steht. Aus (HL) wird es in den Akku geladen. Auch die Überprüfung der Druckerbereitschaft und die Ausgabe auf's Papier kann man sich schenken, denn der Interpreter weiß auch, wie das geht. Was Du in den Zeilen 400 bis 900 programmiert hast, steht dort ab Adresse 05B4h. Zusätzlich wird dabei sogar der Zeilenzähler im DCB unterhalten, was allerdings schon fast seit Gutenbergs Zeiten überflüssig ist. Mit JP 05B4h geht also der Code auf den Drucker. Diese Interpreter-Routine ist ebenfalls ein Unterprogramm, so daß früher oder später auch ein RET angebrochen wird. Deshalb kommen wir elegant aus dem Druckertreiber wieder heraus ohne ein eigenes RET.

Das kleine BASIC-Programm, mit dem der neue Treiber implementiert wird, kann nun folgendermaßen angepaßt werden:

```
10 LN$="123456789": REM 9 Bytes genügen!
30 FOR I=1 TO 9: REM Daher auch eine kürzere Schleife
40 DATA 33,17,1,37,57,126,195,180,5: REM und weniger DATA
```

Die übrigen Zeilen bzw. das übrige der gelisteten Zeilen können von Hartmuts Programm unverändert übernommen werden.

Arnulf Scpp

Sicher ging es nicht nur mir so; man bekommt eine Disk ins Haus und weiss ueber die Laufwerksangaben ueberhaupt nichts. Saemtliche PDRIVE-Parameter zu testen ist doch etwas zu langwierig. Ausserdem geben die Laufwerke dabei Geraeusche von sich, dass einem Angst und Bange wird. Ich machte mich also an die Suche und habe im BOOT/SYS, Sektor 2 die Parameter gefunden (ich hoffe, dass es auch stimmt). Die Tabelle ist in Byte 00 bis 9F gespeichert, wobei jedes Laufwerk die Bytes 0 bis F belegt. Also z.B. Laufwerk 1: 10 - 1F / Laufwerk 4: 40 - 4F usw. Die PDRIVE-Parameter sind wie folgt verteilt:

Byte x0: Spur-Nummer DIRectory

x1:	
x2:	
x3:	TC SP
x4:	SPT SEK
x5:	GPL EIB
x6:	(HD)
x7:	(HD)
x8:	DDSL SBIV
x9:	DDGA AEIV
xA:	(HD)
xB:	(HD)
xC:	TSR SWZ
xD:	TI 1. Teil wobei A=01/B=02/C=04/D=08/E=10
xE:	TI 2. Teil wobei H=01/I=02/J=03/K=04/L=08/M=10
xF:	TD wobei A=00/B=01/C=02/D=03/E=04/F=05/G=06

Die Tabelle wurde einer Systemdisk NEWDOS 80 V 2.5 fuer Modell III entnommen. Bei V 2.0 duerfte es sich wohl aehnlich verhalten. Beim Lesen einer Model I - Disk im Modell III waere schliesslich noch der Befehl WRDIRP (bzw. ueber DIRCHECK) notwendig, da das Directory schreibgeschuetzt sein sollte. (KH)

Computer in Dänemark

Dänemark, das Land der »beschränkten« Computer: Sofie Madsen, mit 104 Lenzen älteste Oma ihres Landes, zweifelte an der Schärfe ihrer Augengläser. Erhielt sie doch wahrlich eine Aufforderung, sich als Erstkläblerin einschulen zu lassen. Oma Sofie knirschte mit den Leihzähnen...

Die Hollerith-Anlage ihres Heimatortes konnte in der Alttersspalte nur bis 99 »zählen«. Und somit war die 104jährige Sofie erst fünf Jahre alt - und schulpflichtig.

VIDHEX - Hexanzeige des Bildschirms mit der HRG

Die Sonderzeichen des Genie 3 (serienmäßig), 2 und 1 (nach Hardwareänderung) mit den ASCII-Codes 00-1F sehen zwar putzig aus. Sieht man sie jedoch in einem Dump mit dem Debugger oder erscheinen sie auf andere Weise auf dem Bildschirm, so ist man gelegentlich ratlos. Sie sind nämlich in keiner Tabelle zu finden, so daß man ihre ASCII-Codes bei Bedarf nur erraten kann. Um diesem Mißstand abzuweichen, entwarf ich das Programm VIDHEX/CMD, das mit Hilfe der HRG 1b anstelle der ASCII- und Sonderzeichen kleine zweistellige Hexzahlen anzeigt. Was dabei herauskommt, ist in dem HRG-Ausdruck am Ende des Listings zu sehen. Es handelt sich um eine Anzeige des Debuggers.

Die Ladeadresse 3900 ist mit der serienmäßigen Hardware natürlich nicht zu realisieren. Das geht mit einer kleinen Zusatzplatine, die Helmut Bernhardt in c't vorstellte (RAM von 3900-3BFF). Wer sie nicht hat, braucht nur ORG auf irgendeine andere Adresse zu setzen (genau 256 Bytes müssen bis zum Himm noch mindestens frei sein).

Das Programm ist mit ZEUS/CMD erstellt. Für EDTASM müssen alle Labels auf max. 6 Zeichen gekürzt werden. Kleinbuchstaben sind nur in den Kommentaren erlaubt. DB muß in DEFB geändert werden, DW in DEFW. In einem DEFB-Statement dürfen die einzelnen Bytes auch nicht durch Komma getrennt in eine gemeinsame Zeile gepackt werden, sondern jedes Byte erhält eine eigene Zeile. Das alles ändert aber nichts an der Programmlogik, die im folgenden erläutert werden soll.

Unter allen Steuerzeichen ist (zumindest in der Direkteingabe über die Tastatur) ESC das überflüssigste (escape, Shift-Hochpfeil, ASCII 27 bzw. 1B). Es hat zudem den Vorteil, daß sich mit ESC nichts auf dem Bildschirm ändert, solange man nicht etwa mit dem Level-4-ROM arbeitet und ESC zusammen mit einer anderen Taste drückt. Deshalb wurde als Trigger für die Umwandlung des Bildschirms ESC ausgewählt. Hierzu erhielt der Tastaturreiber einen kleinen Vorspann, der vor der normalen Tastaturabfrage prüft, ob Shift mit dem Hochpfeil gedrückt wurde. Dies geschieht im Programmsegment newdrv. Wird kein oder ein anderes Zeichen festgestellt, geht es weiter an 4516, dem Beginn des normalen Tastaturreibers (G-DOS bzw. H-DOS).

Nach ESC erfolgt nun die Umwandlung. Hierzu wird zunächst der Bildschirm gerettet, denn er muß gelöscht werden, um die HRG-Anzeige nicht zu stören. Ein Puffer wird für das Kilobyte des Bildschirms gebraucht. Die HRG benutzt von jedem Byte ihres Speichers nur 6 Bits zur Anzeige. Die beiden höchstwertigen Bits bleiben unsichtbar. Was liegt näher, als dort den Bildschirm zu verstauen? Dazu wird zunächst der Videozeiger mit 3C00 geladen, dem Beginn des Screens. Der HRG-Zeiger kommt auf den Anfangswert 0000. In den verschachtelten Schleifen vidsav1 und vidsav2 wird jeweils der Akku mit dem Videobyte geladen und mit C0 undiert, um die übrigen Bits zu löschen. Dieses Viertelbyte im Akku wird nun über den Port 5 in die HRG geladen. Jetzt werden die beiden oberen Bits der Videostelle nach unten rotiert; die nächsten beiden Bits stehen an. Das ist ein bißchen fummelig und kostet auch ein paar Bytes Programmcode, aber 1 kB Bildschirmpuffer im RAM wäre ein vielfacher Verlust.

Anschließend wird der HRG-Speicher gelöscht. Weshalb das nötig ist, wird später erklärt. Seine internen Adressen gehen von 0000 bis 2FFF. Der Zeiger HL startet deshalb nach dem Puffern des Bildschirms mit 0400 (= 1 kB), wo der HRG-Zeiger jetzt gerade steht. Das MSB der HRG wird bei jedem Schleifendurchlauf auf 30, den ersten nicht mehr erlaubten Wert geprüft. Dies geschieht in der Schleife clear.

Nach dem Löschen geht es im UF hexdisp weiter. Es wird jeweils ein Bildschirmzeichen geladen und mit Blank verglichen. Um den Bildschirm übersichtlich zu halten, wird ein Blank nicht mit der Hexzahl 20 angezeigt. In diesem Fall wird das UP byte nicht angesprochen. Das würde ohne vorheriges Löschen der HRG bedeuten, daß die alten Codes dort erhalten blieben. Daher war zuvor die Löschung erforderlich.

Im UP byte wird die Videoadresse auf die HRG-Adresse umgerechnet. Hierzu braucht (zumindest für die oberste Dotzeile) nur 3C vom MSB subtrahiert zu werden. In diesem Falle wird mit 03 undiert, was auf dasselbe hinausläuft. Nacheinander werden nun die beiden Halbbytes in eine Ziffer umgerechnet und angezeigt:

Das obere Nibble wird zunächst durch 16 dividiert und damit ins untere geschoben. DE wird nun als Zeiger auf die Zeichensatztafel chrtab geladen und das UP nibble angesprochen. Je nach Ziffer wird der Zeiger dort bis zu 16mal um fünf Stellen weitergerückt, denn jede Ziffer setzt sich aus fünf Dotmustern zusammen. Die jeweilige Stelle in der HRG wird im UP HRGadr adressiert und bei output das Byte ausgegeben. Für das nächste der fünf Bytes pro Ziffer muß der HRG-Zeiger HL um 1 kB erhöht werden. Das geht am einfachsten durch viermaliges Inkrementieren des MSB.

Um beide Hexziffern auf dem engen Raum einer einzigen Video-Anzeigestelle gut unterscheidbar zu machen, stehen sie untereinander und sind um zwei Dots in der Waagerechten gegeneinander verschoben. Das geschieht beim unteren Nibble (Einerstelle der Zahl) durch zweimaliges RLCA. Da das UP Nibble nicht unterscheidet, welche der beiden Ziffern gerade ansteht, werden die beiden RLCA immer abwechselnd abgearbeitet oder übersprungen. Dazu dient der relative Sprungbefehl JR displc. Die Sprungdistanz displc wird mit dem XOR-Befehl gleich zu Beginn des UP nibble ständig zwischen 00 und 02 (durch Verwendung von Labels variabel gehalten) hin- und hergeschaltet.

Wenn beide Ziffern angezeigt sind, geht es mit der nächsten Bildschirmstelle weiter, bis alle Bildschirmzeichen als Hexzahlen auf dem HRG-Screen stehen. Die HR-Graphik bleibt stehen, bis irgendeine Taste gedrückt wird. Jetzt muß der alte Bildschirm wieder restauriert werden. Die Bildschirmzeichen, die bisher geduldig im Puffer gewartet haben, werden zurückgeladen. Das geschieht im Prinzip wie das Puffern, nur eben umgekehrt: Die beiden oberen Bits aus einer HRG-Stelle werden aus dem Akku hinaus- und in ein Bildschirmbyte hineinrotiert. Damit ist endlich alles erledigt, mit RET geht es zurück ins Betriebssystem.

Die Befehle OUT (1),A und OUT (0),A zum Ein- und Wiederausschalten der HRG stehen an frühst- bzw. spätestmöglicher Stelle. Daher kann der User bei 1,77 MHz ungefähr 2-3 Sekunden lang ein wildes Schauspiel auf dem Screen beobachten. Gute Unterhaltung!

Arnulf Sopp

```

00001 ;=====
00002 ;   Umwandlung der Bildschirmanzeige von ASCII-
00003 ;   Zeichen in Hexzahlen mit Hilfe der HRG 1b
00004 ;   (C) '85 by The HACKTORY
00005 ;=====
3900 00006
00007   ORG      3900h           ;Lade- u. Einsprungsadr.
00008
00009 ;neuer Tastaturreiber für Shift-Aufwärtspfeil
3900 3A403B 00010 newdrv LD      A,(3840h)   ;Tastatur Steuerzeichen
3903 CB5F   00011 BIT      3,A             ;Aufwärtspfeil gedrückt?
3905 2B04   00012 JR      Z,exit           ;falls nein
3907 3A803B 00013 LD      A,(3880h)   ;ja, Shift-Reihe
390A B7     00014 OR      A             ;Shift gedrückt?
390E CA1645 00015 exit   JP      Z,4516H   ;falls nicht Sh.-Hochpf.
00016
00017 ;nach Shift-Hochpfeil Bildsch. retten
390E D301   00018 OUT      (1),A         ;HRG einschalten
3910 21003C 00019 LD      HL,3c00h       ;Bildschirmadresse
3913 55     00020 LD      D,L           ;DE <- 00xx, HRG-Adresse
3914 5D     00021 LD      E,L           ;DE <- 0000
3915 D5     00022 PUSH     DE           ;HRG-Adresse retten
3916 E5     00023 PUSH     HL           ;dto. Videoadresse
3917 01C004 00024 vidsav1 LD      BC,04c0h ;4*2 Bits/Byte, Konst. C0
391A CDA839 00025 vidsav2 CALL   HRGadr    ;HRG-Stelle adressieren
391D 7E     00026 LD      A,(HL)        ;Bildschirmzeichen
391E A1     00027 AND      C           ;nur oberste 2 Bits
391F D305   00028 OUT      (5),A        ;auf HRG ausgeben
3921 CB06   00029 RLC      (HL)         ;Zeichen 2 Bits aufrücken
3923 CB06   00030 RLC      (HL)         ;
3925 10F3   00031 DJNZ     vidsav2      ;bis 1 Byte fertig
3927 23     00032 INC      HL           ;nächste Videostelle
3928 CB74   00033 BIT      6,H          ;Bildsch. überschritten?
392A 2BEB   00034 JR      Z,vidsav1     ;falls noch nicht
00035
00036 ;HRG-Speicher ab nächster freier Stelle löschen
392C CDA839 00037 clear  CALL   HRGadr    ;HRG-Stelle adressieren
392F FE30   00038 CP      30h           ;Speicher überschritten?
3931 2B05   00039 JR      Z,cleared     ;falls ja
3933 AF     00040 XCR      A           ;A <- 00
3934 D305   00041 OUT      (5),A        ;diese Stelle löschen
3936 1B4F   00042 JR      clear        ;nächste Stelle
3938 E1     00043 cleared POP     HL      ;Bildschirmanfang
3939 E5     00044 PUSH     HL          ;für später retten
00045
00046 ;Bildschirm von ASCII- in Hexanzeige ändern
393A E5     00047 hexdisp PUSH   HL      ;Videozeiger retten
393B 4E     00048 LD      C,(HL)        ;Bildschirmzeichen
393C 3E20   00049 LD      A,' '       ;Blank
393E B9     00050 CP      C           ;Blank? (nicht verändern)
393F 77     00051 LD      (HL),A      ;diese Stelle löschen
3940 C46539 00052 CALL     NZ,byte      ;kein Bl., 1 Byte umwand.
00053
00054 ;Zeichen umgewandelt oder Blank überspr.; nächst. Zeichen
3943 E1     00055 POP      HL          ;Bildschirmzeiger
3944 23     00056 INC      HL          ;nächste Stelle
3945 CB74   00057 BIT      6,H          ;Bildsch. überschritten?
3947 2BF1   00058 JR      Z,hexdisp     ;nein, nächstes Byte
00059
00060 ;Nach Anzeige auf Taste warten, dann Bildschirm restaur.
3949 CD4900 00061 CALL     0049h       ;auf Tastendruck warten
394C E1     00062 POP      HL          ;Videoadresse
394D D1     00063 POP      DE          ;HRG-Adresse
394E 0604   00064 restor1 LD      B,4    ;4*2 Bits/Byte

```

```

3950 CDA839 00065 restor2 CALL   HRGadr    ;HRG-Stelle adressieren
3953 DB04   00066 IN      A,(4)        ;HRG-Byte holen
3955 07     00067 FLCA           ;2 oberste Bits
3956 CB16   00068 FL      (HL)          ;in den Bildsch. laden
3958 07     00069 FLCA           ;
3959 CB16   00070 RL      (HL)        ;
395B 10F3   00071 DJNZ     restor2     ;bis 1 Videobyte fertig
395D 23     00072 INC      HL          ;nächste Bildschirmstelle
395E CB74   00073 BIT      6,H          ;Bildsch. überschritten?
3960 2BEC   00074 JR      Z,restor1     ;falls noch nicht
3962 D300   00075 OUT      (0),A        ;HRG ausschalten
3964 C9     00076 RET              ;Tastaturreiber verlass.
00077
00078 ;UP Hexanzeige: 1 Byte ändern
3965 7C     00079 byte   LD      A,H     ;MSB der Videoadresse
3966 E603   00080 AND      03          ;Adr. Vid. -> Adr. HRG
3968 57     00081 LD      D,A         ;neues MSB
3969 5D     00082 LD      E,L         ;HRG-MSB wie Video-MSB
396A 79     00083 LD      A,C         ;Videozzeichen
396B E6F0   00084 AND      0f0h        ;oberes Nibble
396D 0F     00085 RRCA           ;ins untere schieben
396E 0F     00086 RRCA           ;
396F 0F     00087 RRCA           ;
3970 0F     00088 RRCA           ;
3971 21AB39 00089 LD      HL,chrTab-5   ;vor Tab. f. Hexzeich.
3974 E5     00090 PUSH     HL          ;brauchen wir noch
3975 CD7C39 00091 CALL     nibble      ;oberes Nibble anzeigen
3978 79     00092 LD      A,C         ;alter Code
3979 E60F   00093 AND      0fh          ;unteres Nibble
397B E1     00094 POP      HL          ;Tabellenzeiger
00095
00096 ;einzelnes Halbbyte in die HRG laden
397C 47     00097 nibble  LD      B,A     ;als Zähler i. d. Tabelle
397D 3A9639 00098 LD      A,(displc)   ;Sprungdistanz
3980 EE02   00099 XOR      output-displc-1 ;umschalten
3982 329639 00100 LD      (displc),A   ;neu laden
3985 04     00101 INC      B           ;wegen DE = Tabelle -5
3986 23     00102 seekchr INC     HL      ;Zeiger nachstellen
3987 23     00103 INC     HL          ;über 5 Stellen, weil
3988 23     00104 INC     HL          ;5 Codes pro Zeichen
3989 23     00105 INC     HL          ;
398A 23     00106 INC     HL          ;
398B 10F9   00107 DJNZ     seekchr     ;bis Code gefunden
398D 0605   00108 LD      B,5           ;5 Dotzeilen pro Zeichen
398F C5     00109 nibloop PUSH   BC     ;Zähler retten
3990 CDA839 00110 CALL     HRGadr    ;HRG-Stelle adressieren
3993 1B     00111 DEC      DE          ;HRG-Zeiger korrigieren
3994 7E     00112 LD      A,(HL)        ;Dotzeile laden
3995 1B00   00113 JR      $+2          ;variable Sprungdistanz
3996         00114 displc EQU     $-1   ;hier Distanzbyte
3997 07     00115 RLCA           ;lower Nibble verschieben
3998 07     00116 RLCA           ;um 2 Dots
3999 4F     00117 output  LD      C,A     ;Dotzeile retten
399A DB04   00118 IN      A,(4)        ;HRG-Byte mit Videocode
399C B1     00119 OR      C           ;mit Dotzeile verknüpfen
399D D305   00120 OUT      (5),A        ;Dotzeile in HRG laden
399F 14     00121 INC      D           ;im MSB um 1 kB erhöhen
39A0 14     00122 INC      D           ;für nächste Dotzeile
39A1 14     00123 INC      D           ;
39A2 14     00124 INC      D           ;
39A3 23     00125 INC      HL          ;nächster Code für Ziffer
39A4 C1     00126 POP      BC          ;Zähler restaurieren
39A5 10EB   00127 DJNZ     nibloop     ;bis Nibble angezeigt
39A7 C9     00128 RET              ;zurück
00129
00130 ;UP, um die HRG-Adresse auszugeben

```



```

39A5 7B      00131 HRGadr LD      A,E      ;LBB der HRG-Adresse
39A9 D302    00132      OUT      (2),A      ;auf Port ausgeben
39AB 7A      00133      LD      A,D      ;MSB
39AC D303    00134      OUT      (3),A      ;dto.
39AE 13      00135      INC      DE      ;nächste HRG-Stelle
39AF C9      00136      RET
39AF C9      00137
00138 ;Codetabelle für HRG-Zeichensatz:
39B0 02      00139 chrtab DB      2,5,5,5,2 ;0
39B5 04      00140      DB      4,6,5,4,4 ;1
39BA 03      00141      DB      3,4,2,1,7 ;2
39BF 07      00142      DB      7,4,2,4,7 ;3
39C4 04      00143      DB      4,5,7,4,4 ;4
39C9 07      00144      DB      7,1,3,4,3 ;5
39CE 06      00145      DB      6,1,7,5,2 ;6
39D3 07      00146      DB      7,4,2,1,1 ;7
39D8 02      00147      DB      2,5,2,5,2 ;8
39DD 02      00148      DB      2,5,7,2,1 ;9
39E2 02      00149      DB      2,5,7,5,5 ;A
39E7 03      00150      DB      3,5,3,5,3 ;B
39EC 06      00151      DB      6,1,1,1,6 ;C
39F1 03      00152      DB      3,5,5,5,3 ;D
39F6 07      00153      DB      7,1,3,1,7 ;E
39FB 07      00154      DB      7,1,3,1,1 ;F
00155
00156 ;neue Tastatur-Treiberadresse im DCB
4016      00157      ORG      4016h      ;und zwar hier
4016 0039    00158      DW      newdrv     ;Adresse
00159
3900      00160      END      newdrv      ;dort Einsprung

```

00000 Fehler

```

HRGadr 39AB  byte  3965  chrtab 39B0  clear  392C  cleared 393B
displc 3996  exit  390B  hexdisp 393A  newdrv 3900  nibble 397C
nibloop 39BF  output 3999  restor1 394E  restor2 3950  seekchr 39B6
vidsav1 3917  vidsav2 391A

```

Unbegrenztes Spielvergnügen

Sicherlich hat sich schon die ein oder andere Spieler - Natur geärgert, wenn eine schwierige Stelle im Ablauf der Bildchen zu einem großen Verschleiß an "Men" oder "Ships" geführt hat. In zwei Fällen läßt sich Abhilfe schaffen mit Debug oder einem Monitor:

DONKEY-KONG lädt bei mir von 7E2C bis B87B, Einsprung 9000.

Eine Änderung an der Stelle AFD6 : 3D = Dec A
in 00 = Nop

...und bei einem crash erfolgt kein Abzug mehr, es geht mit dem gleichen Guthaben an "Männchen" und Punkten weiter.

JUNGLE-BOY von 5200 bis BFFF, Einsprung 5300

Adresse 77D0 : 3D

in 00 ... erbringt den gleichen Erfolg wie oben.

Nun wünsche ich allen viel Spaß beim Üben der kniffligen Passagen und ich hoffe die Hiscore-Freaks sind nicht allzu böse auf mich!

Dieter Kasper

Freitag, 01011.
August 010001 (lt. L.)*

Als der Computer ersetzt wurde

*(laut Leibniz)

Am Bildschirm definiert

TRS-80 Modell I und III, Star Gemini-10x - Zeicheneditor

Ein Bildschirmeditor für die definierbaren Zeichen des Matrixdruckers Star Gemini-10x ist dieses Programm. Es läuft auf den Rechnern Modell I und III von Tandy und Kompatible.

Der Matrixdrucker Gemini-10x von Star verfügt über einen vom Benutzer definierbaren Zeichensatz („Download Character

Set“). Selbstdefinierte Zeichen können mit Hilfe dieses Zeichensatzes genauso benutzt werden, wie es mit dem normalen Zeichensatz möglich ist. Lediglich kursive Schrift (Italics) und die Umschaltung zwischen nationalen Sonderzeichen sind damit nicht verfügbar. Die Definition dieser Zeichen ist jedoch eine umständliche Angelegenheit. Für jedes Zeichen müssen 9 x 7 Punkte als gesetzt oder nicht gesetzt bestimmt werden. Zusätzlich ist noch festzulegen, ob das Zeichen zum Erreichen einer Unterlänge beim Druck verschoben werden muß.

Das hier vorgestellte Programm erleichtert Benutzern eines TRS-80 mit Diskettenlaufwerk (Modell I oder III) die Definition von Zeichensätzen. Es er-

vorgenommen und bestimmt, ob das Zeichen eine Unterlänge hat oder nicht. Beim Abspeichern eines Zeichens wird auch überprüft, ob Punkte eines Zeichens direkt nebeneinander liegen und damit den Anforderungen für Download Characters widersprechen.

Zur Bedienung

Nach einer Initialisierungsphase von etwa 20 Sekunden erscheint auf dem Bildschirm das 9 x 9-Anzeigefeld für die Zeichen. Unterhalb des Anzeigefeldes wird der Code (ASCII-dezimal) des Zeichens angezeigt und das Zeichen, so wie es im Zeichensatz des Rechners vorhanden ist. Weiterhin erscheint auf dem Bildschirm das Hauptmenü mit den verfügbaren Befehlen und eine zehnspaltige Eingabezeile mit blinkendem Cursor. Zum Editieren in dieser Eingabezeile stehen die Tasten (←) und (Shift)(→) mit den bekannten Funktionen zur Verfügung. Leerzeichen vor oder innerhalb von Befehlen sind nicht zulässig. Alle Befehlseingaben müssen mit (ENTER) abgeschlossen werden.

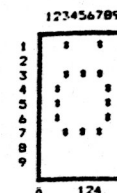
Der wichtigste Befehl des Hauptmenüs ist der Befehl zum Editieren eines Zeichens (E). Die Form „E“ ermöglicht das Editieren des gerade im Anzeigefeld dargestellten Zeichens, „E“ bzw. „E“ das Editieren des Zeichens mit dem nächstniedrigeren beziehungsweise nächsthöheren Code. Die Form „En“ ermöglicht das Editieren des Zeichens mit dem Code n (32-126), „Ex“ das Editieren des Zeichens x. Das hintere Anführungszeichen kann dabei auch weggelassen werden.

Nach der Eingabe des Befehls wird das zu editierende Zeichen im Anzeigefeld dargestellt. Ein blinkender Cursor erscheint in der linken oberen Ecke des Anzeigefeldes. Ein neues Menü erläutert die Eingabemöglichkeiten. Mit den Pfeiltasten kann der Cursor dabei durch das Anzeigefeld bewegt werden.

Mit der Taste (•) (mit oder ohne (Shift)) wird der Punkt an der Cursorposition gesetzt, dargestellt durch ein „•“. Mit der Leertaste wird der Punkt an der Cursorposition gelöscht. Beide Tasten bewirken keine Cursorbewegung. Mit der Taste (0) werden alle Punkte des Anzeigefeldes gesetzt, mit der Taste (CLEAR) alle Punkte gelöscht. Die Taste (Z) ermöglicht die Übernahme eines anderen Zeichens in das Anzeigefeld. Nach dem Druck auf die Taste erscheint unterhalb des Anzeigefeldes eine Eingabezeile mit blinkendem Cursor. Hier kann entweder der Code des gewünschten Zeichens (32-126) oder das Zeichen selbst in der Form „x“ angegeben werden.

Die Eingabe ist mit (ENTER) abzuschließen. Das angegebene Zeichen erscheint dann im Anzeigefeld. Zu diesem Zeitpunkt bereits gesetzte Punkte werden dabei gelöscht. Die Taste (Q) ermöglicht das Abbrechen des Editiervorgangs. Die für das editierte Zeichen bereits abgespeicherte Darstellung erscheint wieder im Anzeigefenster. Nach der Eingabe von (ENTER) wird das Zeichen im Anzeigefeld auf Korrektheit hinsichtlich der Anforderungen für Download Characters überprüft. Stellen, an denen zwei Punkte direkt nebeneinandergesetzt sind, werden in der Form Zeile, Spalte gemeldet. Sind sowohl in einer der beiden oberen als auch in einer der beiden unteren Zeilen Punkte gesetzt, so wird die Meldung „Unterlänge ??“ ausgegeben. Das Zeichen würde später ohne die beiden oberen Zeilen an den Drucker übergeben.

Sind Unkorrektheiten entdeckt worden, so erscheint die Frage „Trotzdem abspeichern?“ Wird als Antwort die Taste (ENTER) gedrückt, so wird das Zeichen im Anzeigefeld abgespeichert. Wird eine andere Taste gedrückt, so bleibt das bisher gespeicherte Zeichen erhalten. Ist das Zeichen korrekt, so wird es ohne weitere Fra-



So sieht der Editor am Bildschirm aus

gen abgespeichert.

Der Befehl „S“ des Hauptmenüs dient zum Abspeichern des Zeichensatzes auf Diskette. Der Befehl hat die Form S-Dateiname. Der Dateiname darf maximal acht Zeichen lang sein und keinen Zusatz (Extension) enthalten. Die Extension /DLC (für Download Character) wird vom Programm angehängt.

Der Befehl „L“ dient entsprechend zum Einlesen eines Zeichensatzes von der Diskette. Er hat ebenfalls die Form L-Dateiname. Für den Dateinamen gelten die gleichen Bedingungen wie beim S-Befehl. Es ist nicht möglich, einzelne Zeichen auf Diskette zu speichern oder von der Diskette zu lesen.

Fehler, die bei der Ein- und Ausgabe von Diskette auftreten, werden abgefangen und führen nicht zum Programmabbruch. Die am häufigsten auftretenden Fehler werden dabei mit deutscher Fehlerbeschreibung gemeldet. Bei anderen wird nur der Fehlercode angegeben. Die Beschreibung des Fehlers muß dann im BASIC-Handbuch nachgesehen werden. Die Fehlermeldung muß durch Drücken von (ENTER) quittiert werden, dann kehrt das Programm ins Hauptmenü zurück.

Der Befehl „X“ bewirkt den Transfer eines Zeichensatzes von der Diskette auf den Drucker, ohne daß der Zeichensatz im Rechner gespeichert wird. Der Befehl hat die Form X-Dateiname. Für den Dateinamen gilt das selbe wie beim L- und S-Befehl.

Der Befehl „K“ sorgt dafür, daß im Drucker der normale Zeichensatz in den freidefinierbaren Zeichensatz kopiert wird. Dies ist die Voraussetzung dafür, mit dem Z-Befehl einzelne Zeichen neu zu definieren.

- E - Zeichen editieren
- S - Zeichensatz sichern
- L - Zeichensatz laden
- X - Zeichensatz übertragen
- D - Zeichensatz an Drucker übergeben
- Z - Ein Zeichen an Drucker übergeben
- K - ROM-Zeichensatz als DLC
- P - DLC-Zeichensatz drucken
- A - DLC-Zeichensatz ein
- 7E.

Der Befehl „Z“ sorgt dafür, daß ein selbstdefinierter Zeichensatz an den Drucker übergeben wird. In der Form „Z“ wird das im Anzeigefeld dargestellte Zeichen übergeben. Mit „Zn“ wird das Zeichen mit dem Code n (32-126) übergeben, mit „Zx“ das Zeichen x.

Der Befehl „D“ bewirkt die Übergabe des gesamten selbstdefinierten Zeichensatzes an den Drucker. Der Befehl „P“ sorgt dafür, daß der auf dem Drucker vorhandene selbstdefinierte Zeichensatz ausgegeben wird. Dies geschieht immer zweizeilig: die obere Zeile zeigt die Zeichen des normalen Zeichensatzes, die untere die des selbstdefinierten Zeichensatzes.

Der Befehl „A“ sorgt dafür, daß auf dem Drucker der selbstdefinierte Zeichensatz eingeschaltet wird. Alle weiteren Ausgaben des Druckers erfolgen mit diesem Zeichensatz. Die Verwendung des Befehls „P“ macht diese Umschaltung allerdings wieder rückgängig. Alle Befehle, die auf den Drucker zugreifen, prüfen zunächst die Einsatzbereitschaft des Druckers. Ist er nicht in der Lage, Zeichen zu empfangen, erscheint die Meldung „Drucker nicht bereit!“ auf dem Bildschirm. Sie verschwindet wieder, sobald der Drucker einsatzbereit ist. Ein Druck auf die (ENTER)-Taste sorgt für den Abbruch des Befehls, für den Fall, daß der Drucker nicht einsatzbereit gemacht werden kann oder soll.

Die folgenden Befehle des Hauptmenüs werden aus Platzgründen nicht auf dem Bildschirm angezeigt: Die Befehle (und) bewirken, daß das Zeichen mit dem nächstniedrigeren beziehungsweise nächsthöheren Code ins Anzeigefeld gebracht wird. Wird der Code eines Zeichens (32-126) oder das Zeichen selbst in der Form „x“ eingegeben, so wird dieses Zeichen angezeigt. Der Befehl „B“ sorgt für die Rückkehr ins BASIC. Die Speicherung des selbstdefinierten Zeichensatzes auf der Diskette erfolgt in derselben Darstellung, die für die Übergabe an den Drucker verwendet wird. So werden pro Zeichen nur 10 Byte auf der Diskette gebraucht, für den gesamten Zeichensatz also 950. Für die Umrechnung von der Darstellung auf dem Bildschirm und im Speicher in die kodierte Darstellung und zurück werden Routinen in Maschinensprache benutzt. Diese Routinen werden über den USB-Befehl angesprochen. Während der Initialisierungsphase werden die Routinen mit POKE in die Strings U0\$ und U1\$ gebracht. Bei der Eingabe des Programms ist darauf zu achten, daß U0\$ 46 Zeichen und U1\$ 53 Zeichen lang ist. Welche Zeichen verwendet werden, ist unwichtig.

Ein Hinweis zum Listing: Programmzeilen, deren Nummern nicht mit 0 enden, sind Kommentarzeilen. Sie können ohne Folgen für den Programmablauf weggelassen werden. Auch das Listing wurde übrigens mit einem selbstdefinierten Zeichensatz gedruckt. Nur so ist es möglich, den „Klammeraffen“ @ und die Umlaute gleichzeitig in einem Listing zu haben. Gerd Kluge

21

(22)

```

6049 REM----- File öffnen, Pointer auf Buffer -----
6050 OPEN "I",1,NM$:CLOSE:OPEN "R",1,NM$:FIELD 1,1 AS P$
6060 Y=256:P=PEEK(VARPTR(P$)+1)+256:PEEK(VARPTR(P$)+2)
6070 CC=CD:FOR CD=32 TO 126:PRINTZP+704,CHR$(CD),CD;
6079 REM----- Kodiertes Zeichen aus Buffer lesen -----
6080 FOR I=0 TO 9:IF Y=256 THEN GET I:Y=0
6090 P(I)=PEEK(P+Y):Y=Y+1:NEXT I
6099 REM----- Zeichen an Drucker übergeben (X) -----
6100 IF XF THEN GOSUB 4040:GOTO 6140
6109 REM----- oder dekodieren (L) -----
6110 FOR ZZ=0 TO 8:PRINTSPD+(ZZ*64),BL$;:NEXT ZZ
6120 Z=USRO(VARPTR(P(0)))
6130 FOR ZZ=0 TO 8:Z=(CD-32,ZZ)=BZ$(ZZ):NEXT ZZ
6140 NEXT CD:CLOSE:XF=0:CC=CC:GOTO 360
6999 REM===== P - DLC-Zeichensatz drucken =====
7000 GOSUB 14000:IF FE THEN 370
7010 LPRINT CHR$(27);"B";CHR$(27);"E"
7020 I=31:GOSUB 7040:I=79:GOSUB 7040
7030 LPRINT CHR$(27);"B";GOTO 370
7038 REM----- Eine Zeile drucken -----
7039 REM----- Standard-Zeichen drucken -----
7040 LPRINT CHR$(27);CHR$(36);:D=0:GOSUB 15000
7050 FOR Y=1 TO 1+47:LPRINT CHR$(Y);:NEXT Y
7059 REM----- DLC-Zeichen drucken -----
7060 LPRINT:LPRINT CHR$(27);CHR$(36);CHR$(1);
7070 FOR Y=1 TO 1+47:LPRINT CHR$(Y);:NEXT Y
7080 LPRINT:LPRINT:RETURN
7999 REM===== A - DLC-Zeichensatz aktivieren =====
8000 GOSUB 14000:IF FE THEN 370
8010 LPRINT CHR$(27);CHR$(36);CHR$(1);:GOTO 370
9998 REM

9999 REM----- UP: Zeichen darstellen -----
10000 FOR ZZ=0 TO 9
10010 PRINTZP+(ZZ*64),Z$(CD-32,ZZ);
10020 NEXT ZZ:RETURN
10999 REM----- UP: Menue löschen -----
11000 FOR ZZ=0 TO 14
11010 PRINTZP+ZZ*64,BL$;BL$;BL$;
11020 NEXT ZZ:RETURN
11999 REM----- UP: Eingabezeichen holen -----
12000 PRINTZP+Z+S,CHR$(191);:IS=INKEY$:
IF IS<>"
THEN RETURN
ELSE FOR I=1 TO 50:NEXT Y:PRINTZP+Z+S,CHR$(CU)::
IS=INKEY$:
IF IS<>" THEN RETURN
ELSE FOR Y=1 TO 50:NEXT Y:GOTO 12000

12999 REM----- UP: Zeile einlesen -----
13000 PRINTZP,"?";:Z=0:S=1:IN$=""
13010 CU=PEEK(15360+CP+S):GOSUB 12000:PRINTZP+Z+S,CHR$(CU);
13020 IF IS=CHR$(8)
THEN IF S=1
THEN 13010
ELSE S=S+1:PRINTZP+S," ";
IN$=LEFT$(IN$,LEN(IN$)-1):
GOTO 13010
13030 IF IS=CHR$(24) THEN 13000
13040 IF IS=CHR$(13) THEN RETURN
13050 IF IS<" " OR IS=CHR$(126) THEN 13010
13060 IF S=11 THEN 13010
13070 IN$=IN$+IS:PRINTZP+S,I;:S=S+1:GOTO 13010
13999 REM----- UP: Drucker bereit " -----
14000 FE=0:IF PEEK(14312)=63 THEN RETURN
14010 PRINTZP+768," Drucker nicht bereit ":
14020 IF INKEY$=CHR$(13) THEN FE=-1: RETURN
14030 IF PEEK(14312)<63 THEN 14020
14040 PRINTZP+769,BL$;BL$;BL$:RETURN
14999 REM----- UP: Byte auf Druckerport geben -----
15000 IF PEEK(14312)<63 THEN 15000
15010 IF PEEK(4)=6 THEN POKE 14312,D ELSE OUT 248,D
15020 RETURN
15999 REM----- Bytes für USR-Routinen -----
16000 DATA 205,127,10,235,33,205,60,26,183,40,3,33
16010 DATA 77,61,6,9,197,19,19,1,1,7,26,161
16020 DATA 40,2,54,42,197,175,71,14,64,9,193,203
16030 DATA 1,16,239,1,65,254,9,193,16,226,201,205
16040 DATA 127,10,235,33,205,60,26,183,40,3,33,77
16050 DATA 61,6,9,197,19,19,1,1,7,175,245,126
16060 DATA 254,42,32,3,241,177,245,197,175,71,14,64
16070 DATA 9,193,203,1,16,237,241,18,1,65,254,9
16080 DATA 193,16,220,201
16999 REM----- ON ERROR Routine -----
17000 IF ERR<45 THEN ON ERROR GOTO 0:END
17010 CLOSE
17020 IF ERR=106 THEN IN$="File existiert nicht ":GOTO 17070
17030 IF ERR=122 THEN IN$="Diskette voll ":GOTO 17070
17040 IF ERR=128 THEN IN$="Filename nicht zulässig ":GOTO 17070
17050 IF ERR=134 THEN IN$="Diskette Schreibgeschützt ":GOTO 17070
17060 IN$="Fehler Nummer ":"+STR$(ERR/2)+1)
17070 PRINTZP+768,IN$;
17080 IF INKEY$="" THEN 17080 ELSE RESUME 370

```


Die Hauptsache von "Die Suche nach dem Schlüssel von Nightshade" ist folgende:

1) Der Abenteuerer muß versuchen die Größe und Stärke seiner Armee zu vergrößern.

2) Der Abenteuerer muß 8 Stücke eines Schlüssels, die zerstreut im Lande Nightshade herumliegen, sich erkämpfen.

Der Bildschirm

Der Bildschirm ist in 5 Sektionen unterteilt. Ganz oben wird die Aufgabe des Spielers angezeigt. Ganz links wird die Größe und Stärke der Armee angezeigt. In der Mitte des Bildschirms wird eine Teilsicht vom Land, und der Standpunkt des Spielers, angezeigt.

Die unterste Sektion wird für Kärtchen gebrauch. Die Sektion Karte zeigt dem Spieler Wahlmöglichkeiten an.

Die Karte

Das Land ist im 30 mal 30 Einheiten unterteilt.

Jede Einheit ist eine mögliche Stellung, die du im Besitz nehmen kannst.

Der Bildschirm zeigt einen kleinen Teil der Karte mit dir in der Mitte.

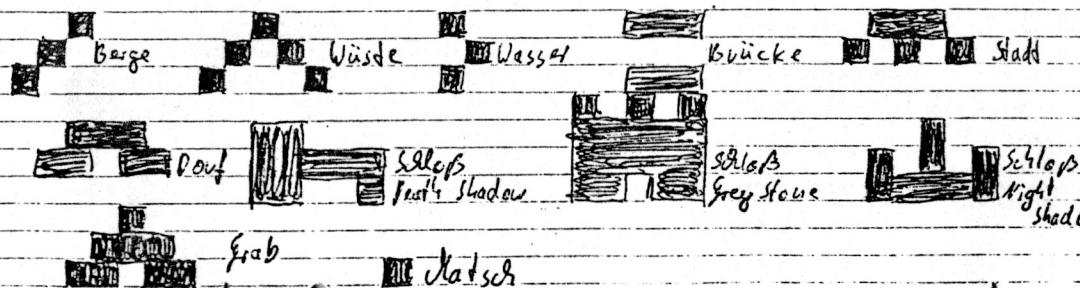
Die Karte stellt das Land mit vier Einheiten in jede Richtung dar.

Wenn du dich bewegst, stehst du selbst still, und nur die Karte bewegt sich unter dir.

Kleine Grafik-Symbole auf der Karte, symbolisieren die Landschaft.

Landstiche, Flüsse, Seen, Berge und Wälder sind in jedem Spiel an der gleichen Stelle. Menschenbezogene Objekte wie: Brücken, Städte, Dörfer, Schlösser und Gräber variieren von Spiel zu Spiel.

Tabelle 1: Symbole



Auf dem Marktplatz bekommst du alles was du für dich, und deine Armee, brauchst.

Marktplätze findest du überall dort wo Menschen sind.

Dinge kosten in kleinen Dörfern mehr als in Städten. Wenn du etwas kaufst, kaufst du für die ganze Armee.

Waffen und Rüstungen

Waffen und Rüstungen machen eine Armee stark und mächtig.

Je mehr Waffen eine Armee besitzt, je mehr kauft sie.

Lebensmittel, Wasser, Arme

Lebensw., Wasser und Arme sind im Spiel unentbehrlich, du brauchst sie um deine Armee am Leben zu halten.

Wenn die Arme winst, geht deine Erschöpfung auf 0% zurück.

Erschöpfung

Wenn du dich bewegst oder kämpfst, wirst du erschöpft. Dieser Faktor wird ganz links in % angezeigt. Ist die % Rate = 100, beginnt die Armee langsam dahin zu nieden. Du stirbst immer zuletzt.

Begegnungen.

Wenn du unterwegs jemanden begegnest, töte sie denn Sie tragen Schätze, Lebensmittel oder Schlüsselstücke.

Schlösser und Gräber

In Schlössern und Gräbern findest du Schätze und Schlüsselstücke.

Schlöß Nightshade

Schlöß Nightshade kann nur mit dem Schlüssel geöffnet werden.

Eine Armee von Feind drachen bewacht dieses Schloß.

Nur wenn du sie tötest, und den König des Landes befreist hast, ist das Spiel beendet.

So wenn viel Spass beim Abenteueren willst, suchst du dich selbst.

Basic-Dialekte im Vergleich - Teil 2

In Ausgabe 18 begannen wir einen Vergleich der Basic-Versionen der beliebtesten Heimcomputer. Auch bei den Befehlen für die hochauflösende Grafik gibt es etliche Unterschiede.

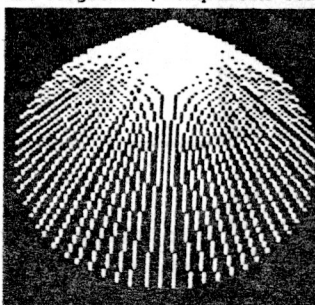
Entscheidendes Kriterium für den Kauf eines Heimcomputers war bis vor kurzem noch die Speicher-Kapazität. Der ZX80 bot nur 1 KByte RAM, der VC 20 immerhin das fünffache. Diese Zeiten sind vorbei. Denn ein Heimcomputer bietet heute mindestens 16 KByte RAM. Mit ein Punkt bei der Entscheidung für oder gegen einen neuen Heimcomputer sind heute die grafischen Fähigkeiten. Nach höherer Bildschirmauflösung, mehr Farben und besserer Grafik-Software bei möglichst niedrigeren Preisen ist gefragt. 64.000 oder gar 256.000 Grafikpunkte auf einem Bildschirm sind schon keine Seltenheit mehr und ein Ende dieser Entwicklung ist nicht abzusehen. Der Mitsubishi DC-186 zum Beispiel bringt es auf über eine Million einzeln ansteuerbarer Pixel. Für dieses Gerät müssen etwa 17.000 Mark bezahlt werden, aber auch preiswertere Geräte leisten schon Erstaunliches. Schon für weniger als 1000 Mark gibt es Computer zu kaufen, deren Grafikfähigkeiten wie Auflösung, Anzahl der Farben und Software-Unterstützung einiges bieten. Bei der Erstellung eigener Grafik-Programme lassen sich fast alle Heimcomputer mit Basic in der Grundversion in zwei Gruppen einteilen: die »Poke«-Gruppe und die Grafik-Basic-Gruppe. Zur ersten Gruppe, die Grafiken nur mit entsprechenden Tlokits oder über komplizierte POKE-Routinen erzeugen, zählen zum Beispiel alle Commodore-Computer und der TI 99/4A.

Auch bei Grafikbefehlen: Gleicher Name, andere Funktion

Die meisten Heimcomputer erlauben den Einsatz von Basic-Befehlen wie »Line«, »Set« und »Circle«, um Grafiken zu erzeugen. Bei diesen Befehlen gibt es aber bei jedem

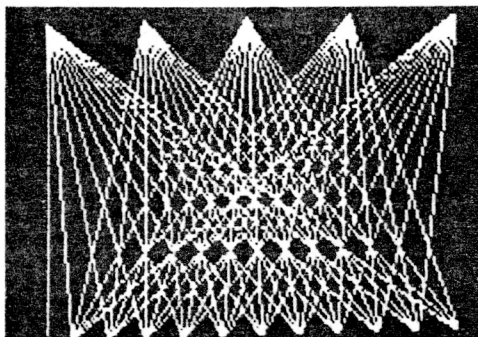
Heimcomputer eine andere Syntax oder gar ein anderes Befehlswort. Der »Line«-Befehl dient im allgemeinen dazu, eine Linie zwischen zwei Punkten zu zeichnen. Beim Spectrum aber wird der »Line«-Befehl nur mit den Befehlen »Save« und »Input« eingesetzt und hat dort ganz andere Funktionen. Um bei diesem, wie auch bei anderen Computern (siehe Übersicht auf Seite 129), eine Linie zu ziehen, ist der Befehl »Draw« einzusetzen. Für Computer, die diesen Befehl nicht eingebaut haben, kann das Programm aus Listing 1 eingesetzt werden. Die allgemeine Syntax des »Line«-Befehls ist: Line (X,Y)-(V,W),pl,p2. Mit den Variablen X und Y wird ein Punkt des Grafikbildschirms definiert, von dem aus eine Linie zum Punkt V,W in der Farbe pl gezogen werden soll. Die Variable p2 kann bei den Computern Dragon 32 oder Dragon 64 »B« oder »BF« sein. Durch den Einsatz von »B« interpretiert das System die Punkte X,Y und V,W als gegenüberliegende Ecken eines Rechtecks, das mit der Option »BF« ausgemalt wird. Bei anderen Computern als denen von Dragon sollten diese Parameter fortgelassen werden, da sie eine Fehlermeldung hervorrufen können. Der »Pset (X,Y,F)«-Befehl setzt einen Punkt mit den Koordinaten X,Y in der Farbe F. In anderen Basic-Dialekten lautet das entsprechende Befehlswort »Plot«. Etwas kompli-

zierter ist da schon der Einsatz des Befehls »Circle«, der einen Kreis erzeugt. Die allgemeine Syntax hierzu lautet: Circle (X,Y),R,F,H,B,A,E. Wieder geben die Variablen X und Y einen Punkt auf dem Bildschirm an. Dies ist der Mittelpunkt des Kreises, der mit dem Radius R in der Farbe F gezeichnet wird. Hier sind die Möglichkeiten der meisten Heimcomputer erschöpft. Die Syntax der Dragon-Computer erlaubt noch drei weitere Parameter. Die Variable HB steht für den Quotienten Höhe/Breite, mit dem man die Exzentrizität für das Zeichnen einer Ellipse definiert. Mit A und E wie Anfang und Ende wird vereinbart, ob nur Kreis- oder Ellipsenbögen gezeichnet werden sollen. Diese Eingaben sind nicht erforderlich, bieten aber eine einfache Möglichkeit, komplizierte Teil-



Diese Effekte entstehen mit dem Programm aus Listing 7, obwohl nur Linien gezeichnet werden

Auch bei diesem Beispiel sind nur »Plot«-Befehle eingesetzt worden



zeile zu zeichnen. Die Werte für Anfang und Ende beziehen sich auf den 350-Grad-Vollkreis und geben den zu zeichnenden Ausschnitt an. Auch beim Spectrum kann man einen Teilkreis erzeugen, indem an die »Draw«-Anweisung ein dritter Parameter angehängt wird. Heimcomputer, die den Befehl »Circle« nicht haben, dafür aber »Plot« oder »Set« beziehungsweise »Pset«, können den Befehl durch ein kleines Programm simulieren (Listing 2). Für grafische Darstellungen muß bei den meisten Heimcomputern der Bildschirmmodus umgeschaltet

```
1 ' ziehe Linie von
2 ' der Koordinate
3 ' A,A zu der
4 ' Koordinate A,B
10 FOR P = A TO B
20 SET A,P
30 NEXT P
```

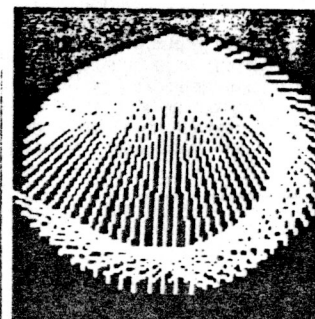
Listing 1. Mit dieser Befehlsfolge wird der »Plot to« oder »Draw«-Befehl simuliert

werden. Die Befehle dazu heißen »Pmode«, »Mode«, »Fgr«, »Graphics« oder »Hgr«. Dabei sind, je nach Computertyp, unterschiedliche Parameter einzusetzen.

Auch mit einfachen Befehlen sind Grafiken zu erzeugen

Alle Beispielprogramme sind auf den Dragon-Computern 32 und 64 ohne Änderungen lauffähig. Für andere Typen müssen die jeweils angegebenen Befehle ersetzt werden.

Wer noch keine Erfahrungen mit der Programmierung von Grafiken hat, sollte die Beispiele aus Listing 4,5 und 6 ausprobieren. Mit dem Programm aus Listing 4 wird ein Punkt kurzzeitig ein- und ausgeschaltet, indem die Befehle »Set« und »Reset« eingesetzt werden. Für andere Computer werden diese Befehle durch »Plot« beziehungsweise »Unplot« ersetzt. Das Programm aus Listing 5 setzt den »Line«-Befehl ein, um eine Linie und ein Rechteck zu zeichnen. In anderen Basic-Dialekten ist dieser Befehl durch »Draw« oder »Plot to« zu ersetzen. Für Computer, die nur die Befehle »Set« und »Reset« haben, kann das Programm aus Listing 3 eingesetzt werden. Das Programm aus Listing 6 schließlich stellt Kreise mit unterschiedlichen



Das Programm aus Listing 7 durchläuft einige Schleifen. So entstehen immer neue Bilder.

Listing 2. Auch Computer ohne den »Circle«-Befehl können Kreise zeichnen

```
1 ' Mit dieser Befehlsfolge
2 ' können Computer, die
3 ' den »Circle«-Befehl
4 ' nicht kennen, einen
5 ' Kreis darstellen.
10 ' R = Radius
20 ' CHOR = Cursor horizontal
30 ' CVER = Cursor vertikal
40 ' PI = Kreiszahl (3.145)
50 FOR A = 0 TO 2*PI STEP PI/R
60 X = R * COS A + CHOR
70 Y = R * SIN A + CVER
80 SET X,Y : 'oder PLOT X,Y
90 NEXT A
91 ' Der Schnittpunkt von
92 ' CHOR und CVER ist der
93 ' Mittelpunkt des Kreises
```

```
10 ' Befehlsfolge um ein
20 ' Rechteck zu zeichnen
40 ' K = Kantenlänge
50 ' A = Eckpunkt oben links
60 ' B = " oben rechts
70 ' C = " unten links
80 ' D = " unten rechts
90 ' jeweils die Werte der
100 ' X- Achse sind anzugeben
110 ' BY- und CY- Werte müssen
111 ' angegeben werden.
120 FOR Z = A TO A + K
130 SET Z,A : NEXT Z
140 FOR Z = BY TO BY + K
150 SET B,Z : NEXT Z
170 FOR Z = D TO D-K STEP -1
180 SET Z,D : NEXT Z
190 FOR Z = CY TO CY - 10 STEP -1
200 SET C,Z : NEXT Z
```

Listing 3. Dieses Programm stellt ein Rechteck dar

Höhe/Breite-Verhältnissen dar. Wer durch diese kleinen Beispiele noch nicht genügend motiviert ist, sich mehr mit Computergrafik zu beschäftigen, sollte das Programm aus Listing 7 eingeben. Kein Bild und keine Erklärung kann die Effekte,

unterschiedliche Namen, die Funktion bleibt die Gleiche. Schwieriger gestaltet sich schon die Umsetzung eigener Ideen in entsprechende Programme. Das ist aber gar nicht so kompliziert, wie wohl mancher denken wird.

Der Bildschirm, oder besser der Grafikbildschirm, ist in Zeilen und Spalten aufgeteilt. Zeilen liegen horizontal und Spalten vertikal auf dem Schirm. Jeder Schnittpunkt dieser beiden Linien wird als Punkt oder

Der Bildschirm als Koordinatensystem

Pixel bezeichnet und kann beeinflusst werden. Bei den meisten Heimcomputern ist die obere linke Ecke mit den Koordinaten 0,0 bezeichnet. Von ihr aus werden die Spalten und Zeilen fortlaufend gezählt, abhängig von der höchsten Auflösung. Beim Dragon 64 zum Beispiel lautet

```
10 'XXDEMO/LINEXX
15 'XXXXXXXXXXXXXXXXXXXXXXXXXXXX
20 PH=4 'X AuflösungsgradX
25 SC=1 'X FarbgruppeX
30 HF=0 'X HintergrundfarbeX
35 X=128 'X-Koordinate von PIX
40 Y=96 'X-Y-Koordinate von PIX
45 U=200 'X-Koordinate von P2X
50 M=125 'X-Y-Koordinate von P2X
55 M=800 'X MarteschleifeX
60 'XXXXXXXXXXXXXXXXXXXXXXXXXXXX
65 PHODEM, I:SCREEN1, SC:PCLSHF
70 FOR I=0 TO M:NEXT I
75 LINE(X,Y)-(U,M),PSET
80 FOR I=0 TO M:NEXT I
85 LINE(X,Y)-(U,M),PSET
90 FOR I=0 TO M:NEXT I
95 LINE(X,Y)-(U,M),PSET,B
100 FOR I=0 TO M:NEXT I
105 LINE(X,Y)-(U,M),PSET,B
110 FOR I=0 TO M:NEXT I
115 LINE(X,Y)-(U,M),PSET,BF
120 FOR I=0 TO M:NEXT I
125 LINE(X,Y)-(U,M),PSET,BF
130 GOTO 70
```

die größte Zeilennummer 192 und die höchste Spaltennummer 256. Dadurch wird die unterste rechte Ecke definiert. Halbiert man diese beiden Zahlen, so erhält man die Werte 128 und 96, wodurch die Bildschirmmitte bezeichnet sein sollte. Eine Überprüfung dieser Berechnung mit dem »Set«-Befehl wird dieses Ergebnis bestätigen. Mit dem Punkt P=(128,96) hat man einen Koordinatenpunkt bestimmt, wobei der erste Wert die X-Achse und der zweite Wert die Y-Achse angibt. Bei allen Befehlen, die diese Koordinatenschreibweise einsetzen, muß man darauf achten, daß jeder Wert im zulässigen Bereich liegt. Für die X-Achse liegt dieser in unserem Bei-

```
10 'XXDEMO/PSET,PRESETXX
15 'XXXXXXXXXXXXXXXXXXXXXXXXXXXX
20 PH=4 'X AuflösungsgradX
25 SC=1 'X FarbgruppeX
30 HF=0 'X HintergrundfarbeX
35 X=128 'X-KoordinateX
40 Y=96 'X-Y-KoordinateX
45 F=5 'X ZeichenfarbeX
50 M=200 'X MarteschleifeX
55 'XXXXXXXXXXXXXXXXXXXXXXXXXXXX
60 PHODEM, I:SCREEN1, SC:PCLSHF
65 FOR I=0 TO M:NEXT I
70 PSET(X,Y,F)
75 FOR I=0 TO M:NEXT I
80 PRESET(X,Y)
85 GOTO 65
```

Listing 4.
Blinkender Punkt

Das Wenn und das Aber sind das Erbteil der Dummen.

Italienisches Sprichwort

Unsere Prinzipien dauern gerade so lange, bis sie mit unseren Leidenschaften oder Eitelkeiten in Konflikt kommen, und ziehen dann jedesmal den kürzeren.

Theodor Fontane,
deutscher Schriftsteller (1819-1898)

Listing 5. Durch »Pset, B« wird ein Rechteck dargestellt. Für andere Computer als den Dragon 32/64 muß das Programm aus Listing 3 eingesetzt werden.

spiel zwischen 0 und 256, für die Y-Achse sind Werte von 0 bis 192 zulässig. Nicht jeder Basic-Interpreter läßt bei Grafikbefehlen den Einsatz von Gleitkommazahlen zu. Deshalb sollte man bei allen Berechnungen nur Integer-Variablen anwenden.

Ohne Mathematik geht es auch bei Grafik nicht

Solange Linien parallel zum Bildrand gezeichnet werden, erfordert die Bestimmung der Koordinaten keine große Aufgabe. Für ein Rechteck sind vier Linien zu definieren, je zwei parallel zu den horizontalen

und zwei zu den vertikalen Bildschirmrändern. Davon abweichende Lagen eines Rechtecks, wie zum Beispiel ein auf der Spitze stehendes Quadrat, müssen über Formeln berechnet werden. In diesen Fällen reicht es aus, die Eckpunkte zu bestimmen. Jeweils zwei Eckpunkte liegen mit dem Mittelpunkt des Quadrats auf einer Geraden. Das heißt aber auch, daß die X-Koordinaten der Punkte A,C und M gleich sind, wie auch die Y-Koordinaten der Punkte B,D und M. Ausgehend von einer Auflösung von 256 x 192 Punkten haben wir also folgende Koordinaten festgelegt: M = (128,96), A = (128,y(A)), B = (x(B),96), C = (128,y(C)) und D = (x(D), 96). Da das Qua-

Spruch der Woche

Es kann passieren, was will: es gibt immer einen, der es kommen sah.

Ich erkenne die Männer an den Früchten, die sie bevorzugen. Apfelsinenesser werden gute Familienväter. Birnenesser schätzen Ruhe und Gemütlichkeit. Himbeeren verraten den Romantiker, Nüsseknacker sind sehr kompliziert. Vorsicht vor Männern, die grüne Früchte lieben. Am temperamentvollsten sind Grapefrucht-Freunde.

Martine Carol,
französische Filmschauspielerin
(1922-1967)

drat in der Mitte des Bildschirms gezeichnet wird, sind die Abstände der Eckpunkte zum Mittelpunkt konstant, nämlich genau halb so lang, wie die Länge der Diagonale (d) des Quadrats. Legen wir fest, daß die Kantenlänge aus 60 Pixeln besteht, errechnen sich die Abstände der Eckpunkte vom Mittelpunkt aus mit der Formel: $60 \cdot \sqrt{2}/2$. Aufgerundet ergibt das 43 Pixel, die nun, immer vom Mittelpunkt aus betrachtet, die Koordinaten der Eckpunkte ergeben: M = (128,96), A = (128,53), B = (85,96), C = (128,139) und D = (171,96).

Daß der Bildschirmmittelpunkt für Berechnungen von Grafiken zentrale Bedeutung hat, zeigt auch das nächste Beispiel. Auf dem Bildschirm soll ein gleichseitiges Dreieck gezeichnet werden. Die Dreiecksbasisseite soll dabei parallel zum unteren Bildschirmrand verlaufen. Wieder legen wir eine Seitenlänge von 60 Pixel fest. Da der Bildschirmmittelpunkt auch genau in der Mitte der Höhe (h) des Dreiecks

Die Teilung der Hand in Finger hat keine Schwächung herbeigeführt, vielmehr ihre Leistungsfähigkeit gesteigert. Ebenso erhöht der Staatsmann, der anderen einen Anteil an der Führung gibt, durch die gemeinsame Leistung den Erfolg seiner Tätigkeit.

Plutarch,
griechischer Philosoph und Historiker (50-125 n. Chr.)

liegt, berechnen sich die X-Koordinaten relativ einfach aus: $x(B) = x(M) - 30 = 98$ und $x(C) = x(M) + 30 = 158$. Der Mittelpunkt (M) hat die Koordinaten M = (128,96). Für die Punkte A, B und C ergibt sich also

folgendes: A = (128, y(A)), B = (98, y(B)) und C = (158, y(C)). Anschließend wird die Dreieckshöhe (h) berechnet: $h = 60 \cdot \sqrt{3}/2$. Das ergibt 52 und die Strecke (h/2) ist somit 26 Pixel lang. Um nun die richtigen Y-Werte zu finden, müssen diese 26 Pixel für den Punkt A von der Mittelpunktordinate abgezogen und für die Punkte B und C addiert werden. Die Koordinaten lauten also: A = (128,70), B = (98,122) und C = (158,122).

Attraktive Grafiken durch Bewegung

Erweitert man die bisher besprochenen Programme durch eine oder mehrere For-Next-Schleifen, kann man Bewegungen simulieren. Die einfachste Bewegung ist wohl die eines Punktes, der über den Bildschirm läuft (Listing 8). Dabei wird ein Pixel ein- und ausgeschaltet, wobei der Wert der X-Koordinate immer um den Wert »eins« er-

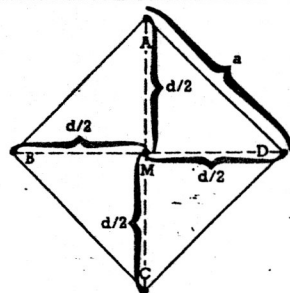
höht wird. Im Programm aus Listing 9 wird der Befehl »Line« eingesetzt, um immer größere Quadrate zu zeichnen. Dadurch entsteht der Eindruck, als ob sich ein langer Tunnel öffnen würde. Dieser Effekt wird

Befehl	Computer	Apple II	Atari	Colour Genie	Dragon 32/64	Laser/ V2200	Sinclair ZX81	Sinclair Spectrum	TRS-80 Level II	Sharp MZ 700
Zieht eine Linie von (x,y) nach (vw)	HPlot x,y to vw	Draw to vw: x,y vorher mit Plot defini- eren	Plot x,y to vw	Linie (x,y) — (vw), P set	siehe Li- stung 1	in Listing 1 Plot ein- setzen	Draw vw: x,y vorher mit Plot defi- nieren	siehe Li- stung 1	siehe Li- stung 1	
Setzt eine Linie (x,y) nach (vw)	Auf Hinter- grund- farbe um- schalten. HPlot x,y to vw	Auf Hinter- grundfarbe umschalten und Draw to vw eingeben	Nplot x,y to vw	Linie (x,y) — (vw), Preset	in Listing 1 Reset einsetzen	in Listing 1 Unplot einsetzen	Over 1: Draw, vw: x,y vorher mit Plot defi- nieren	in Listing 1 Reset einsetzen	in Listing 1 Reset einsetzen	
	Plot x,y	Plot x,y	Plot x,y	Pset (x,y)	Set x,y	Plot x,y	Plot x,y	Set x,y	Set x,y	
	Plot x,y in Hinter- grund- farbe	Plot x,y in Hintergrund- farbe	Nplot x,y	Preset (x,y)	Reset x,y	Unplot x,y	Plot; Over 1: x,y	Reset x,y	Reset x,y	
	siehe Li- stung 2	siehe Listing 2	Circle (x,y), r	Circle (x,y), r	siehe Li- stung 2	siehe Li- stung 2	Circle (x,y), r	siehe Li- stung 2	Circle (x,y), r nur mit Plot- ter wirk- sam	
	Call FB H0 oder H0r, H0r	Graphics (Mode)	FCLS	PCLS	CLS	CLS	CLS	CLS	CLS	
	In Listing 3 Set durch Plot er- setzen	In Listing 3 Set durch Plot ersetzen	Plot A to B to C to D to A	Linie (A,B)- (C,D), Pset,B	siehe Li- stung 3	In Listing 3 Set durch Plot er- setzen	Draw A,B: Draw B,C: Draw C,D: Draw D,A	siehe Li- stung 3	siehe Li- stung 3	
	GR mit 40x48 Punkten, 16 Farben	Graphics 3 mit 40x48 Punkten, 4 Farben, Gra- phics 4 mit 80x48 Punk- ten, 2 Farben	FGR mit 160x96 Punkten, 4 Farben (alte ROM's)	Pmode 0 mit 128x96 Punk- ten, 2 Far- ben; Pmode 1 mit 128x96 Punkten, 4 Farben; ROM's	Mode 1 mit 128x64 Punkten, 8 Farben	64x44 Punkte, 2 Farben	256x192, 8 Farben	128x48, 2 Farben	80x50, 8 Farben	
	HGR mit 280x160 Punkten, 16 Farben	Graphics 5 mit 80x48 Punkten, 4 Farben, Gra- phics 6 mit 160x96 Punk- ten, 2 Farben	Graphics 5 mit 80x48 Punkten, 4 Farben, Gra- phics 6 mit 160x96 Punk- ten, 2 Farben	Pmode 2 mit 192 x 128 Punkten, 2 Farben; Pmode 3 mit 192x128 Punkten, 4 Farben; Pmode 4 mit 256x192 Punkten, 2 Farben						
	HGR2 mit 280x192 Punkten, 16 Farben	Graphics 7 mit 160x96 Punkten, 4 Farben, Gra- phics 8 mit 320x192 Punkten, 2 Farben								

Übersicht der wichtigsten Grafikbefehle ▲

▼ So berechnen sich die Koordinaten für ein auf der Spitze stehendes Quadrat.

aber durch die fortlaufende Bewe-
gung erzielt. Als stehendes Bild
sieht man tatsächlich nur Quadrate.
In dieser Ausgabe haben wir die
unterschiedlichen Grafikbefehle
der gängigen Heimcomputer ge-
genübergestellt. Die Computer VC
20 und Commodore 64 fehlen in die-
ser Übersicht, da sie nur mit Toolkits
wie zum Beispiel »Simon's Basic«
Grafik erzeugen. Der nächste und
letzte Teil der »Basic-Dialekte im
Vergleich« wird sich mit den System-
befehlen beschäftigen.
(F.L.Moersdorf/rs)



$$\text{Formel: } d = a \cdot \sqrt{2}$$

$$\text{bei } a = 60 \text{ Pixel} \\ \Rightarrow d/2 = 43 \text{ Pixel}$$

$$\Rightarrow A = (128,53) \\ B = (85,96) \\ C = (128,239) \\ D = (171,96)$$

Zeichencodes in Computern

hex	dez.	ASCII	hex	dez.	ASCII	hex	dez.	ASCII	hex	dez.	ASCII
00	0	NUL	20	32	Space	40	64	@	60	96	'
01	1	SOH	21	33	!	41	65	A	61	97	a
02	2	STX	22	34	"	42	66	B	62	98	b
03	3	ETX	23	35	#	43	67	C	63	99	c
04	4	EOT	24	36	\$	44	68	D	64	100	d
05	5	ENQ	25	37	%	45	69	E	65	101	e
06	6	ACK	26	38	&	46	70	F	66	102	f
07	7	BEL	27	39	'	47	71	G	67	103	g
08	8	BS	28	40	(48	72	H	68	104	h
09	9	HT	29	41)	49	73	I	69	105	i
0A	10	LF	2A	42	*	4A	74	J	6A	106	j
0B	11	VT	2B	43	+	4B	75	K	6B	107	k
0C	12	FF	2C	44	,	4C	76	L	6C	108	l
0D	13	CR	2D	45	-	4D	77	M	6D	109	m
0E	14	SO	2E	46	.	4E	78	N	6E	110	n
0F	15	SI	2F	47	/	4F	79	O	6F	111	o
10	16	DLE	30	48	0	50	80	P	70	112	p
11	17	DC1	31	49	1	51	81	Q	71	113	q
12	18	DC2	32	50	2	52	82	R	72	114	r
13	19	DC3	33	51	3	53	83	S	73	115	s
14	20	DC4	34	52	4	54	84	T	74	116	t
15	21	NAK	35	53	5	55	85	U	75	117	u
16	22	SYN	36	54	6	56	86	V	76	118	v
17	23	ETB	37	55	7	57	87	W	77	119	w
18	24	CAN	38	56	8	58	88	X	78	120	x
19	25	EM	39	57	9	59	89	Y	79	121	y
1A	26	SUB	3A	58	:	5A	90	Z	7A	122	z
1B	27	ESC	3B	59	;	5B	91	[7B	123	{
1C	28	FS	3C	60	<	5C	92	\	7C	124	
1D	29	GS	3D	61	=	5D	93]	7D	125	}
1E	30	RS	3E	62	>	5E	94	^	7E	126	~
1F	31	US	3F	63	?	5F	95	_	7F	127	DEL

Bedeutung der wichtigsten Steuerzeichen

NUL = ohne Wirkung, SOH = start of header, STX = start of text, ETX = end of text, EOT = end of transmission, ENQ = enquiry (Aufforderung der Gegenstation zum Senden), ACK = acknowledge (Bestätigung, Rückmeldung), DLE = data link escape (Umschalten auf eine andere Steuerzeichengruppe), DC = Device Control (DC1 = XON, DC3 = XOFF), NAK = negative acknowledge, SYN = Synchronisationszeichen, ETB = end of transmission block, VT = vertical tabulating (Cursor nach oben), HT = horizontal tab. (Cursor nach rechts), BS = Back Space (Cursor nach links), LF = Line feed (Cursor nach unten), BEL = bell (akustisches Zeichen, Klingel), CR = Carriage Return, FS = field separator, GS = group separator, US = unit separator, Space = Leerzeichen, DEL = Delete/Rub out.

Deutsche Sonderzeichen

Zeichen	hex	dez.	hex	dez.
§	15	21	40	64
Ä	8E	142	5B	91
Ö	99	153	5C	92
Ü	9A	154	5D	93
ä	84	132	7B	123
ö	94	148	7C	124
ü	81	129	7D	125
ß	E1	225	7E	126

Heute: Bildschirmgestaltung (1)

Wort:	Text ausgeben	Variable ausgeben	Ausgabe: enger Abstand	Ausgabe: weiter Abstand
Bedeutung:	Ausgabe des Textes zwischen Anführungszeichen	Ausgabe des Werts der Variablen X	Gibt Texte/Zahlen unmittelbar nebeneinander aus (bei Zahlen: 1 Leerst. Zwischenr.)	Gibt Texte/Zahlen mit größerem Abstand aus. (6-10 Leerst., je nach System)
System:				
ACORN	PRINT "Hallo!" (in allen Systemen gleich)	PRINT X (in allen Systemen gleich)	PRINT X;Y;Z (in allen Systemen gleich)	PRINT X,Y,Z (in allen Systemen gleich)
ALPHA-TRONIC	PRINT kann bei fast allen Rechnern durch ? ersetzt werden.			
APPLE				
ATARI				
COLOUR GENIE				
COMMODORE 8096/C-64/VC20				
SIMONS BASIC				
EX-BASIC				
DRAGON				
EPSON HX 20				
IBM PC				
LASER 310				
LASER 3000				
MSX-BASIC				
ORIC				
SCHNEIDER CPC				
SHARP MZ-Serie				
SHARP PC-1500				
SIRIUS				
SPECTRAVIDEO				
TANDY C.C.				
TANDY TRS 80				
TI-99/4A				
-EXT-BASIC				
ZX-81				
ZX-SPECTRUM				

Alle BASIC-Wörter im Vergleich

Ab dieser Ausgabe finden Sie in jedem Heft BASIC-Befehle, -Anweisungen und -Funktionen, wie sie auf den wichtigsten Heim- und Personalcomputern aussehen. Am Ende der Serie erhalten Sie dazu ein alphabetisches Verzeichnis aller vorgestellten BASIC-Wörter. Die Tabellenform erleichtert Ihnen das Übersetzen von Programmen aus einem BASIC-Dialekt in einen anderen.

Bitte beachten Sie: Wenn ein Ausdruck in Kursivschrift erscheint, dann ist eine Taste gemeint, die Sie drücken müssen!

Heute: Bildschirmgestaltung (1)

Wort:	Bildschirm löschen	Piepton erzeugen	Auf Spalte positionieren	Auf Spalte u. Zeile position.
Bedeutung:	Löscht alle Zeichen auf dem Bildschirm	Erzeugt einen kurzen Piepton (=akustisches Signal, abgekürzt BEL)	Setzt die Schreibmarke (Cursor) auf Spalte S und gibt dort den Text aus	Setzt die Schreibmarke (Cursor) auf Zeile Z und Spalte S und gibt dort den Text aus
System:				
ACORN	CLS	PRINT CHR\$(7)	PRINT TAB(S); "Hallo" oder PRINT TAB(S) "Hallo!"	PRINT TAB(S,Z); "Hallo!"
ALPHA-TRONIC		PRINT CHR\$(7)	Diese Anweisung gilt für alle Systeme außer für den SHARP PC-1500, wo es heißt: CURSOR(S); PRINT "Hallo!"	PRINT\$(Z,S); "Hallo!"
APPLE	HOME	BEEP		VTAB Z:HTAB S:PRINT "Hallo!"
ATARI	PRINT "ESCAPE,CLEAR"	PRINT "ESCAPE,CONTROL+2"		POSITION(S,Z);PRINT "Hallo!"
COLOUR GENIE	CLS	PRINT CHR\$(7)		PRINT @ X, "Hallo!" (X=(Z-1)*40+S)
COMMODORE 8096/C-64/VC20	PRINT "CLF"			POKE 214,Z:POKE 211,S: SYS 58460:PRINT "Hallo!"
SIMONS BASIC	ebenso			PRINT AT(S,Z) "Hallo!"
EX-BASIC	ebenso			PRINT @ X, "Hallo!" (X=(Z-1)*40+S)
DRAGON	CLS	BEEP		PRINT @ X, "Hallo!" (X=32*(Z-1)+S)
EPSON HX-20	CLS	SOUND 10,10		LOCATE S,Z:PRINT "Hallo!"
IBM PC	CLS	BEEP		LOCATE Z,S:PRINT "Hallo!"
LASER 310	CLS	SOUND 1,1		PRINT @ X, "Hallo!" (X=32*(Z-1)+S)
LASER 3000	HOME	SOUND 1,1		HTAB(S):VTAB(Z):PRINT "Hallo!"
MSX-BASIC	CLS	BEEP		LOCATE S,Z:PRINT "Hallo!"
ORIC	CLS	PING		PRINT @ S,Z; "Hallo!"
SCHNEIDER CPC	CLS	PRINT CHR\$(7)		LOCATE S,Z:PRINT "Hallo!"
SHARP MZ-Serie	CLS	MUSIC"+G4"		CURSOR S,Z:PRINT "Hallo!"
SHARP PC-1500	CLS	BEEP ON:BEEP 1,1,1000		Nicht sinnvoll bei Einzelzeilen-Display
SIRIUS	PRINT CHR\$(12)	PRINT CHR\$(7)		
SPECTRAVIDEO	CLS	BEEP		LOCATE(S,Z):PRINT "Hallo!"
TANDY C.C.	CLS	SOUND 100,3		PRINT @ X, "Hallo!" (X=32*(Z-1)+S)
TANDY TRS 80	CLS	PRINT CHR\$(7)		PRINT @ (Z,S); "Hallo!"
TI-99/4A	CALL CLEAR	CALL SOUND(200,440,2)		
-EXT-BASIC	ebenso	ebenso		DISPLAY AT(Z,S); "Hallo!"
ZX-81				
ZX-SPECTRUM	CLS	BEEP		PRINT AT Z,S; "Hallo!"

Heute: Bildschirmgestaltung (2)

Wort	Cursor hoch (↑)	Cursor tief (↓)	Cursor links (←)	Cursor rechts (→)
Bedeutung:	Setzt die Schreibmarke (CURSOR) eine Zeile nach oben	Setzt die Schreibmarke (CURSOR) eine Zeile nach unten	Setzt die Schreibmarke (CURSOR) eine Zeile nach links	Setzt die Schreibmarke (CURSOR) eine Zeile nach rechts
System:				
ACORN	PRINT CHR\$(11)	PRINT CHR\$(10)	PRINT CHR\$(8)	PRINT CHR\$(9)
ALPHATRONIC	PRINT CHR\$(30)	PRINT CHR\$(31)	PRINT CHR\$(29)	PRINT CHR\$(28)
APPLE	CALL -998	CALL -922	CALL -1008	PRINT CHR\$(21)
ATARI	PRINT CHR\$(28)	PRINT CHR\$(29)	PRINT CHR\$(30)	PRINT CHR\$(31)
COLOUR GENIE	PRINT CHR\$(27)	PRINT CHR\$(26)	PRINT CHR\$(24)	PRINT CHR\$(25)
COMMODORE 8096/C-64/VC20	PRINT CHR\$(145)	PRINT CHR\$(17)	PRINT CHR\$(157)	PRINT CHR\$(29)
SIMONS BASIC				
EX-BASIC				
DRAGON	PRINT CHR\$(94)	PRINT CHR\$(10)	PRINT CHR\$(8)	PRINT CHR\$(9)
EPSON MX-20	PRINT CHR\$(30)	PRINT CHR\$(31)	PRINT CHR\$(29)	PRINT CHR\$(28)
IBM PC	PRINT CHR\$(30)	PRINT CHR\$(31)	PRINT CHR\$(29)	PRINT CHR\$(28)
LASER 310	PRINT CHR\$(27)	PRINT CHR\$(10)	PRINT CHR\$(8)	PRINT CHR\$(9)
LASER 3000	PRINT CHR\$(30)	PRINT CHR\$(31)	PRINT CHR\$(29)	PRINT CHR\$(28)
MSX-BASIC	PRINT CHR\$(30)	PRINT CHR\$(31)	PRINT CHR\$(29)	PRINT CHR\$(28)
ORIC	PRINT CHR\$(11)	PRINT CHR\$(10)	PRINT CHR\$(8)	PRINT CHR\$(9)
SCHNEIDER CPC	PRINT CHR\$(11)	PRINT CHR\$(10)	PRINT CHR\$(8)	PRINT CHR\$(9)
SHARP MZ-Serie	PRINT CHR\$(18)	PRINT CHR\$(17)	PRINT CHR\$(20)	PRINT CHR\$(19)
SHARP PC-1500				
SIRIUS				
SPECTRAVIDEO	PRINT CHR\$(30)	PRINT CHR\$(31)	PRINT CHR\$(29)	PRINT CHR\$(28)
TANDY C.C.	PRINT CHR\$(94)	PRINT CHR\$(10)	PRINT CHR\$(8)	PRINT CHR\$(9)
TANDY TRS 80	PRINT CHR\$(27)	PRINT CHR\$(26)	PRINT CHR\$(24)	PRINT CHR\$(25)
TI-99/4A				
EXT-BASIC				
ZX-81	PRINT CHR\$(112)	PRINT CHR\$(113)	PRINT CHR\$(114)	PRINT CHR\$(115)
ZX-SPECTRUM	PRINT CHR\$(11)	PRINT CHR\$(10)	PRINT CHR\$(8)	PRINT CHR\$(9)

Heute: Bildschirmgestaltung (2)

Wort	INVERS	NORMAL	Blinken	Formatierte Ausgabe
Bedeutung:	Stellt Zeichen Invers (dunkel auf hellem Untergrund) dar	Schaltet in den Normalzustand zurück (helle Zeichen auf dunklem Hintergrund)	Läßt die Schrift in kurzen Abständen aufleuchten	Druckt Zahlen und Texte formatiert, also in Tabellenform. SS enthält Formatangaben
System:				
ACORN	mit COLOUR (abhängig vom Mode)	MODE 0	COLOUR-Anweisung abhängig vom MODE	@%-... (String format) PRINT A,B,...
ALPHATRONIC	PRINT CHR\$(28)	PRINT CHR\$(18)		PRINT USING SS:A,B,...
APPLE	INVERSE	NORMAL	FLASH	Hilfsprogramm erforderlich*
ATARI	<input checked="" type="checkbox"/> -Taste drücken	<input checked="" type="checkbox"/> -Taste drücken		Hilfsprogramm erforderlich*
COLOUR GENIE				PRINT USING SS:A,B,...
COMMODORE 8096/C-64/VC20	PRINT CHR\$(18)	PRINT CHR\$(146)		Hilfsprogramm erforderlich*
SIMONS BASIC	INV a,b,c,d		FLASH A,B (A=Farbe, B=Geschw.)	USE SS, STR\$(A):PRINT (A muß erst in String umgew. werden)
EX-BASIC				PRINT USING SS:A,B,...
DRAGON				PRINT USING \$:A,B,...
EPSON MX-20				PRINT USING SS:A,B,...
IBM PC	COLOR 0,7	COLOR 2,0	COLOR X (x>16)	PRINT USING SS:A,B,...
LASER 310				PRINT USING SS:A,B,...
LASER 3000	INVERSE	NORMAL	FLASH	PRINT USING SS:A,B,...
MSX-BASIC				PRINT USING SS:A,B,...
ORIC	PRINT CHR\$(27);"@*"; CHR\$(27);"w"	PRINT CHR\$(27);"G"; CHR\$(27);"P"	PRINT CHR\$(27);"L"	Hilfsprogramm erforderlich*
SCHNEIDER CPC	PRINT CHR\$(24)		mit EVERY	PRINT USING SS:A,B,...
SHARP MZ-Serie	mit COLOR			PRINT USING SS:A,B,...
SHARP PC-1500	mit POINT, G-CURSOR, G-PRINT			USING SS: PRINT A,B,...
SIRIUS				PRINT USING SS:A,B,...
SPECTRAVIDEO	PRINT CHR\$(27);"p"	PRINT CHR\$(27);"q"		PRINT USING SS:A,B,...
TANDY C.C.				PRINT USING SS:A,B,...
TANDY TRS 80	PRINT CHR\$(16)	PRINT CHR\$(17)		PRINT USING SS:A,B,...
TI-99/4A				
EXT-BASIC				PRINT USING SS: A,B,...
ZX-81	G-Taste drücken	G-Taste drücken	FLASH 1	Hilfsprogramm erforderlich*
ZX-SPECTRUM	INVERS 1	INVERS 0		

Berichtigungen zum letzten INFO

Zum Info selbst möchte ich noch ein paar Punkte bemerken.
Die Übersicht der Basic-Dialekte auf Seite 18 enthält ein paar Fehler. Das Colour Genie und der TRS80 kennt die Befehle SWAP (A,B) und SPACE\$(B) nicht. Im Diskbasic wird SPACE\$ zwar nicht als Fehler interpretiert, jedoch nicht richtig abgearbeitet.

Peter Spieß

* Ergänzung und Änderung *

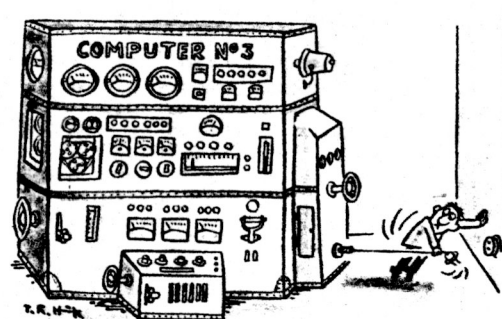
Synchronisierte Taktumschaltung fuer Videogenie

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

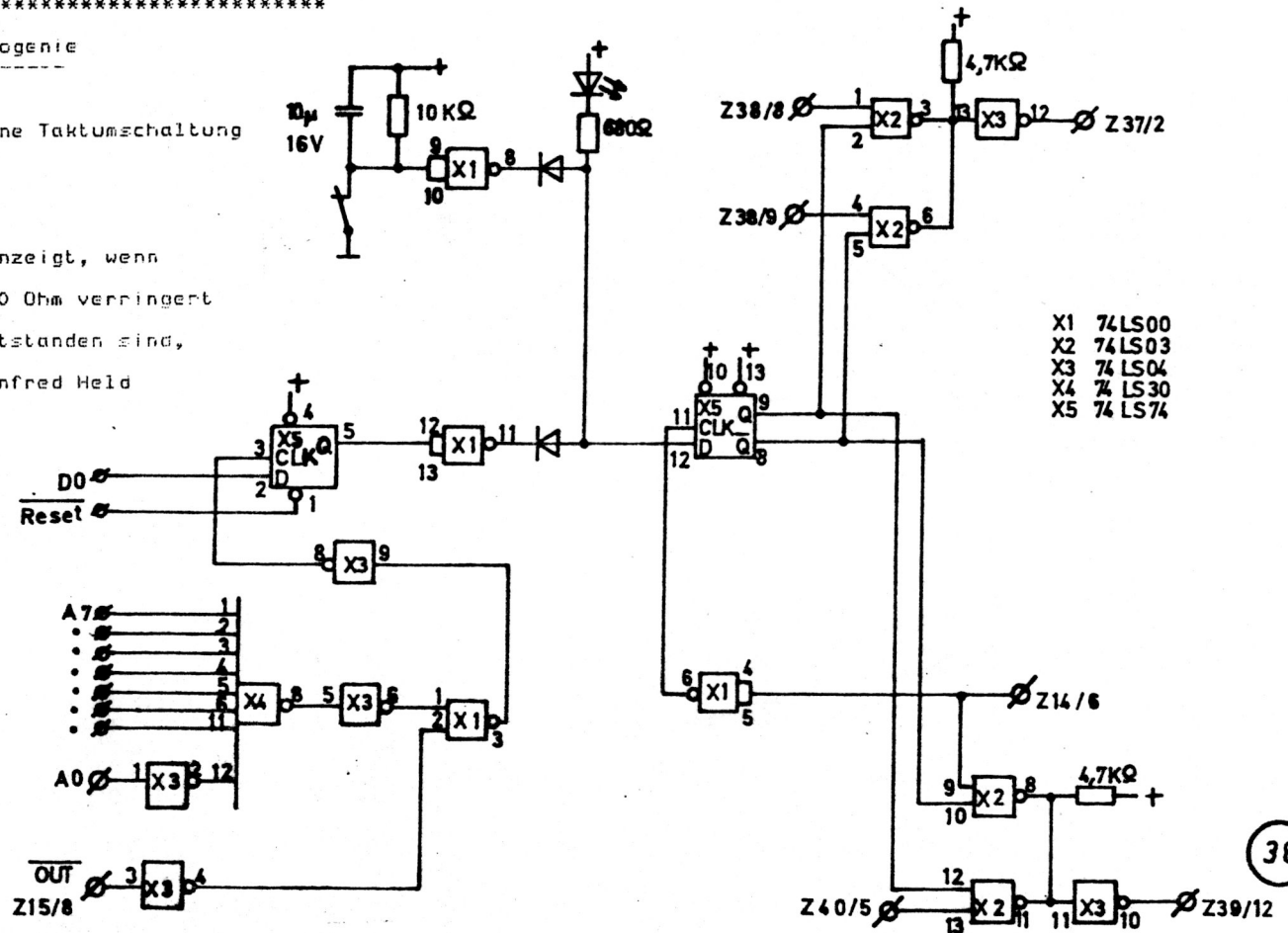
- Am Ic X3 ist Pin 10 und Pin 11 vertauscht.
- Am Ic X5 ist Pin 8 und Pin 9 vertauscht.

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



„Nichts auf der Welt ist wirklich vollkommen!“



Im letzten Clubinfo fragt Walter Piller, was man als Einsteiger mit dem ECB-BUS anfangen kann. Das regt mich jetzt zu einem Grundsatzartikel für alle Clubmitglieder an. Damit soll auch der Computerbesitzer der sich für Hardware nicht (oder noch nicht) interessiert wissen, wovon wir doch schon seit einigen Club-Infos schreiben.

Es gibt in unzähligen Fachzeitschriften Bauvorschläge für die diversesten Hardware-Erweiterungen. Ich will da nur die gebräuchlichsten aufzählen:

High-resolution-grafik; Speichererweiterungen; EPROM-Bank; Hardware Uhr mit Batteriepufferung; Parallelschnittstelle; serielle Schnittstelle (V 24); Eprommer; Basicode-Hardware; Supertape; Joystick; Relais-Steuerkarte;.....

Wer nun irgendeine dieser feinen Dinger kauft oder selberbaut, wird auf 2 Probleme stoßen:

Entweder ist die Erweiterung nicht speziell für den TRS/VG gebaut und ist damit nicht oder nur schwierig verwendbar. Oder es steckt schon eine Erweiterung am Expansion-port. Dann muß man je nach Anwendung ständig die diversen Erweiterungen an oder abstecken. Das ruiniert in kurzer Zeit die ohnehin nicht sehr massiven Platinenstecker auf der Hauptplatine.

Die Lösung sieht also folgendermaßen aus:

Es führt ein Kabel vom Expansion-port zu einer Platine mit 10 Steckplätzen. Dieses Kabel bleibt ständig stecken. 1 Karte (die sogenannte Basiskarte) entkoppelt alle Signale zum und vom Computer.

Die restlichen 9 Steckplätze stehen für Erweiterungen aller Art zur Verfügung. Diese können ständig stecken bleiben, da sie im passiven Zustand keinerlei Einfluß auf den Computer haben. Damit erklärt sich die Frage nach dem Anwenderkreis fast von selbst:

Jeder der sich mit Hardware-Erweiterungen beschäftigt und die ewige Herumsteckerei satt hat, wird dieses Bauprojekt gut gebrauchen können.

Nochmals: Die ECB-Bus Platine samt Basiskarte kann allein noch überhaupt nichts. Sie bildet lediglich die Basis für kommende Hardware-Basteleien aller Art.

Was alles von der Hardware-Gruppe entwickelt oder adaptiert wird, hängt von der Interessenrichtung jedes Mitglieds und auch vom Leserecho des gesamten Club80 ab.

Bei mir sind zur Zeit Schaltpläne für folgende Einrichtungen verfügbar (keine Platinen):

Parallelschnittstelle 3 mal 8 bits (mit 8255)

Serielle Schnittstelle (mit 8251)

Hardware-Uhr mit Akku-Pufferung

HRG - Grafik mit Texas Ic's (TMS....) + 8 mal 4116

Hardware - Arbeitskreis ECB - BUS System

Warum gerade ECB-BUS ?

Wenn man ein solches Steckkartensystem entwickelt, stellt sich gleich zu Anfang die Frage nach der Norm. Es soll ja nicht so sein, daß jedes Mitglied der Hardware-Gruppe seine Karten frei verdrahtet, sondern nach Möglichkeit jeder von der Entwicklung des Anderen profitieren kann. Das zwingt zu einer einheitlichen Belegung der Steckerstifte. Und wenn man schon soweit ist, kann man sich gleich am Markt umsehen, ob es schon eine derartige Norm gibt. Und siehe da es gibt sie: Die ECB-Norm. Das ist ein System zur Vereinheitlichung aller Erweiterungen für beliebige Computer, soweit sie mit einer Z80 CPU bestückt sind. Ursprünglich wurde dieses System für CP/M - Rechner der verschiedensten Hersteller entwickelt. Damit konnten Erweiterungen verschiedener Hersteller mit fast jedem CP/M Rechner verbunden werden. Dieses Konzept paßt uns ausgezeichnet denn es gibt am Markt schon einiges an Bausätzen oder gar fertigen Bausteinen zu kaufen.

Es wird auch laufend von den Zeitschriften 'elektor' und c't etwas für das ECB-System veröffentlicht, was für uns nach minimalen Modifikationen einsetzbar ist. Einheitlich ist dabei auch die Größe jeder Steckkarte. Nämlich genau eine sogenannte Europakarte, von der das System auch seinen Namen hat. (Euro-card-bus)

Die Maße sind: 160 mal 100 mm. Das ist sehr handlich; sodaß die gesamte ECB-Hardware samt Netzteil in einem relativ kleinen Metallgehäuse untergebracht werden kann.

So, ich hoffe daß nach Lektüre dieses Beitrages jeder Clubfreund weiß, was wir im Schilde führen und wir auf keinen Fall ein Club im Club sein wollen. Sollte noch die eine oder andere Frage offen geblieben sein, so bitte ich Euch, ohne Hemmungen zu schreiben oder mich anzurufen.

Walter

W. Zwickel

ECB -Bus Netzteil

Wie im Club Info Nr.8 von Hartmut berichtet, habe ich mir Gedanken um die Stromversorgung für unseren ECB -Bus gemacht. Die Aufgabenstellung war, wir brauchen ein Netzteil mit den folgenden Spannungen/Strömen: +12V/1A, -12V/0.5A, +5V/3,5A bis 5A und -5V/0.5A.

Ein Netzteil mit diesen Daten und sogenannter Längsregelung ist im Prinzip einfach aufzubauen, das Problem ist aber der Preis. Alleine der Trafo kostet min. 50,-DM, der Längsregler für 5A ca. 30,-DM, eine Kostenrechnung für alle Bauelemente einschl. Platine, Kühlkörper usw. ergab einen Preis von ca. 130,-DM.

Die billigere Alternative dazu ist ein Kaufnetzteil. Die Firma:

Erich-Willi MEYER

6343 Frohnhausen (tel 02771/35071)

Pöstfach 74c8

bietet ein Schaltnetzteil! mit folgenden Daten: +5V/3,5A, -5V/0,5A, +12V/2A, -12V/0,5A. Der Preis, komplett aufgebaut incl. Gehäuse 99,80DM.

Ich habe mir dieses Netzteil inzwischen gekauft und getestet. M.E. ist dieses Netzteil vom Preis und Leistung her für unsere Zwecke ideal geeignet. Wer bei 5V mehr als 3,5A benötigt kann auch die Version mit 5V/5A (119,80DM) oder die Version 5V/7,5A (149,50 DM) kaufen.

Ulrich Böckling

Roboter festgenommen

LOS ANGELES, 19. August (AFP). Wegen »öffentlicher Ruhestörung« hat die Polizei von Los Angeles im luxuriösen Villenviertel Beverly Hills einen Roboter »festgenommen«, der knarrend und klirrend durch die Straßen stapfte. Zwar konnten die Beamten dank des Walkie-talkie-Systems des ferngelenkten Roboters mit dessen Besitzer Verbindung aufnehmen, doch weigerte sich dieser, Name und Adresse zu nennen. Daraufhin wurde der Automat ohne weiteres Aufheben vom Abschleppdienst weggeschafft.

Zwischenbericht:

In diesem Info wird an mehreren Stellen über das ECB-Bus- Projekt berichtet. Ich möchte hier kurz über die eigentliche Adapterplatine ein paar Worte verlieren.

Nachdem Walter Zwickel die Schaltung für den ECB-Adapter entworfen und in Fädeltechnik aufgebaut hatte, war es an der Zeit, sich über die sog. Entflechtung der Platine Gedanken zu machen. Mit diesem Problem beschäftigte sich vor allem Manfred Held. Leider mußte er schon nach kurzer Zeit feststellen, daß eine vollständige Entflechtung nicht möglich war. Es würden immer ein paar, in Fädeltechnik nachzulöten, Verbindungen übrigbleiben.

Wie sehr der Manfred damit recht hatte, erwies sich, als Walter Zwickel die Möglichkeit erhielt an einem Großcomputer mit CAD - Programm eine Leiterplattenentflechtung zu machen. Selbst dieser Computer mit dem entsprechenden Spezialprogramm war nicht in der Lage alle Leiterbahnen vollständig zu entwirren.

Als Ergebnis erhielt der Walter zwei Platinenzeichnungen für eine doppelseitige Platine, bei der die beiden Anschlußleisten (Computer->Adapterplatine und Adapterplatine->96poliger ECB-Stecker) in Fädeltechnik verdrahtet werden müssen. Die Bestückungsseite der Platine enthält nur relativ wenige Verbindungen (siehe Layout).

Grundsätzlich gibt es nun zwei Möglichkeiten von dem Layout Gebrauch zu machen:
1. Man läßt sich von einer Firma eine zweiseitige Platine mit allen Bohrungen und Durchkontaktierungen herstellen.
2. Eines der Mitglieder der Hardwaregruppe (wahrscheinlich Manfred Held) stellt eine einseitige Platine her und die fehlenden Leitungen werden in Fädeltechnik nachgelötet.

Gegen die erste Möglichkeit spricht vor allem der relativ hohe Preis pro Platine. Die zweite Möglichkeit ist zwar billig, erfordert jedoch ein bisschen Erfahrung im Umgang mit dem Lötkolben. Da aber sowieso gelötet werden muß (siehe oben), fällt dieses Argument nicht ins Gewicht. Eine entgeltliche Entscheidung ist jedoch noch nicht gefallen.

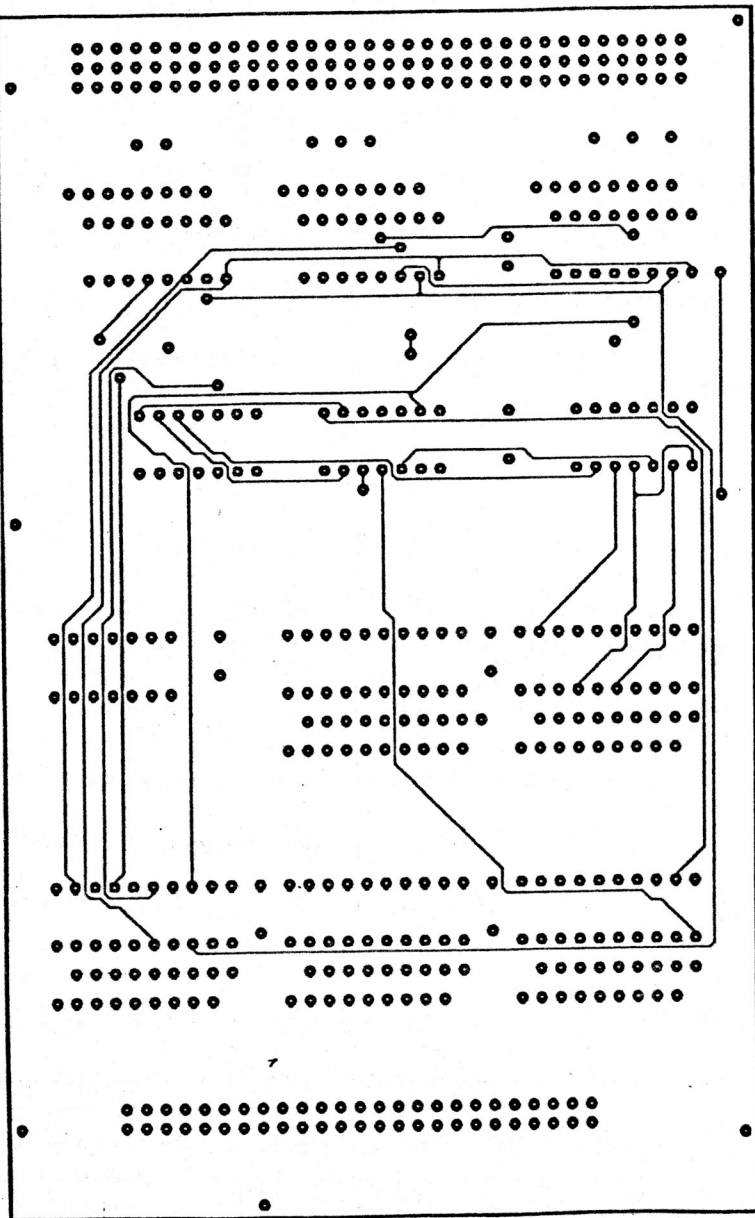
Zur Zeit wird durch Ulrich Böckling von der mit dem Computer erstellten Layout-Zeichnung eine Fotopositivfolie erstellt. Sobald diese Arbeit erledigt ist, wird eine Entscheidung (einseitige/doppelseitige Platine) getroffen und ich glaube, daß wir schon vor Erscheinen des nächsten Clubinfos den Mitgliedern der Hardwaregruppe die Platine zur Verfügung stellen können.

Sollte irgend jemand noch Vorschläge oder Einwände im Zusammenhang mit der Platinenerstellung haben, so bitte ich diese möglichst schnell (am besten telefonisch bei mir) vorzubringen.

Das war's in Sachen Hardware, tschüß

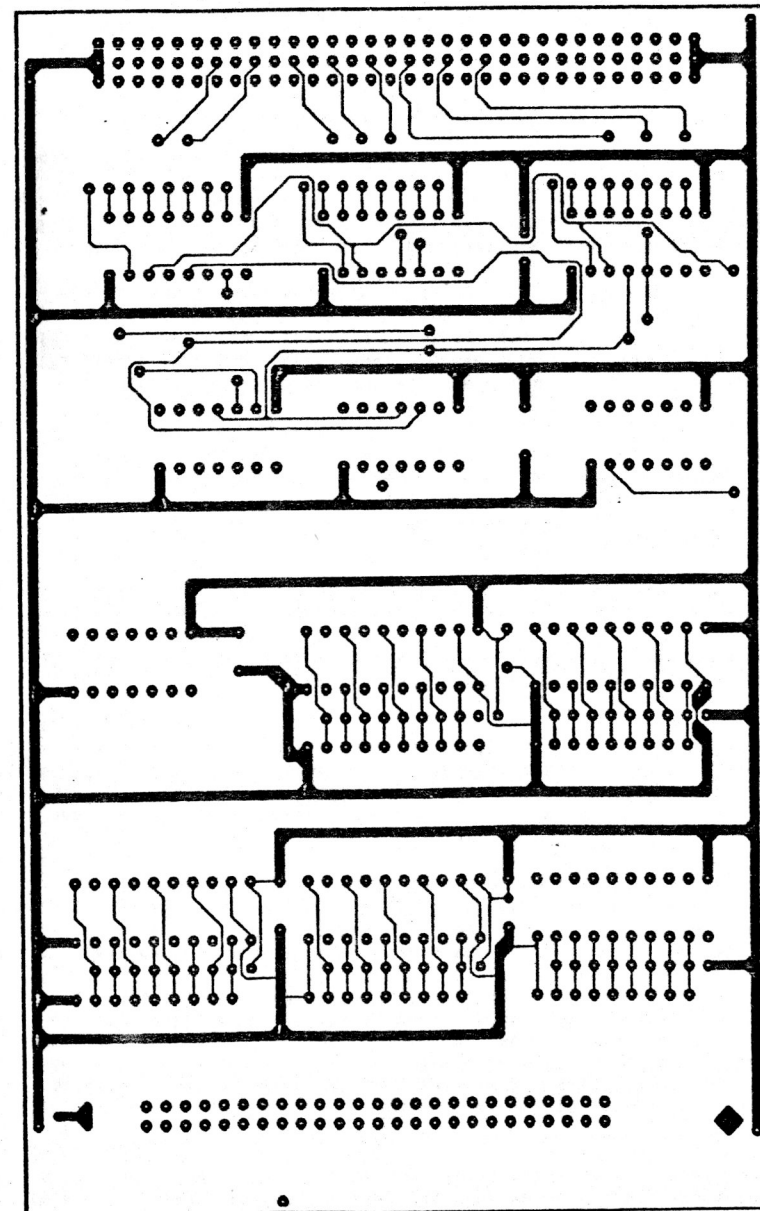
Hartmut Obermann

43



Bestückungsseite

44



Lötseite

Hans-Joachim Regge

Test-adapter

Die Fehlersuche wird oft zur Tortur, wenn die Europakarte gut bestückt ist und zudem in einem Gehäuse sitzt. Die beiden hier beschriebenen Adapterkarten schaffen da Abhilfe.

Die Idee der Verwendung von Adapterkarten ist nun keineswegs neu. Leider sind solche Karten im allgemeinen recht teuer, zumal man für unterschiedliche Bussysteme auch verschiedene Testadapter benötigt. Aus diesem Grund sind die beiden vorliegenden Adapterkarten entstanden, die beide mit Steckverbindern nach DIN 41612 ausgerüstet sind. Der wesentliche Unterschied liegt in der Anzahl der Steckerkontakte: die eine Karte ist für Systeme mit 64poligen, die andere für solche mit 96poligen Steckern vorgesehen. Sie sind nicht nur für den ECB- oder MC-65-Bus, sondern auch für eine Vielzahl anderer Bussysteme geeignet.

Kleine Kapazitäten

Die Busleitungen eines Mikroprozessorsystems werden mit Taktfrequenzen bis zu 8 MHz und mehr betrieben. Wird nun durch Verlängerung der Busleitungen die Kapazität derselben (gegen eine andere Leitung oder gegen Masse) zu groß, so kann es zum Übersprechen oder – der zweite Fall – zur Verformung der Signale kommen. Beides kann die Funktion des Gerätes beeinträchtigen und muß deshalb auf jeden Fall vermieden werden.

Bei den beiden hier vorgestellten Karten sind mit Absicht die Leiterbahnen nur 0,33 mm breit gewählt worden, um die besagten Kapazitäten so gering wie möglich zu halten. Die Bilder 1 und 2 zeigen das Layout der 96poligen Adapterkarte. Bei der 64poligen Karte konnte zwei-

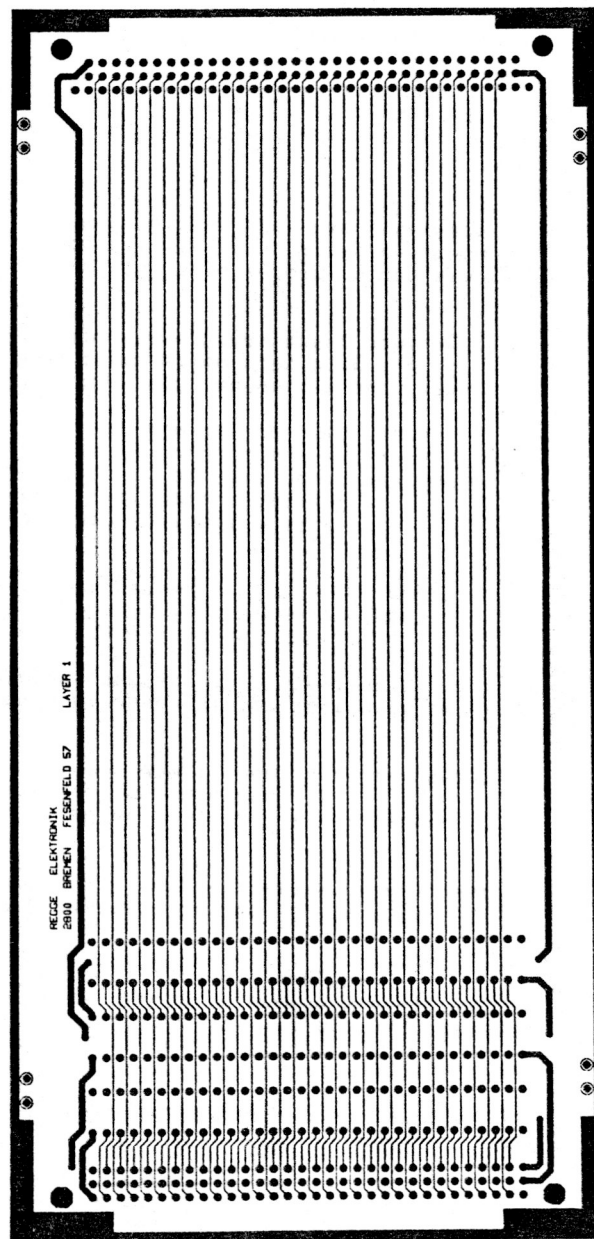


Bild 1. Die Bauteilseite der 96poligen Adapterkarte

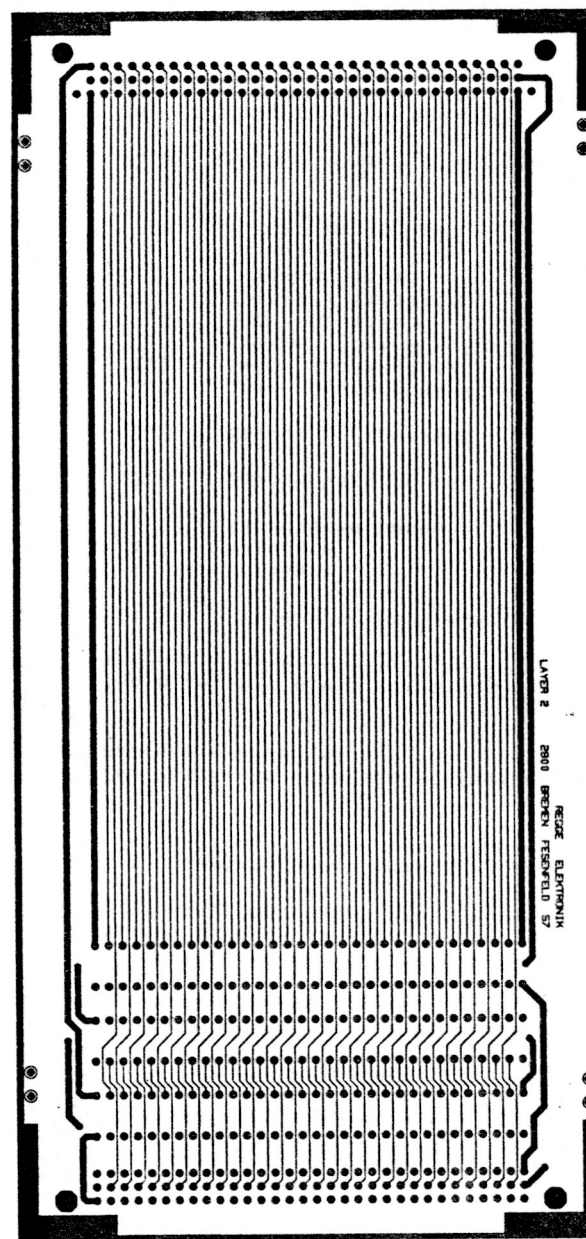


Bild 2. Die Lötseite des 96poligen Testadapters

schen zwei Signalleitungen jeweils eine dünne Masseleitung untergebracht werden, um eine zusätzliche Leiterbahnenkopplung zu erreichen. Versuche ergaben, daß sowohl der mc-CP M-Computer als auch der MC-65 problemlos arbeiten, wenn Systemkarten mit diesen Adaptern betrieben wurden.

Eine große Hilfe: Schalter

Zur Untersuchung von Karten, auf denen Kurzschlüsse vermutet werden, ist es erforderlich, einzelne oder auch alle Leitungen vom System zu trennen. Hierzu sind auf den Adapterkarten zwei (64polig) bzw. drei (96polig) Reihen DIL-Schalter vorgesehen, mit denen alle Leitungen aufgetrennt werden können. Besonders bewährt haben sich da die Spezialschalter von Siemens, die über einen offenen Drahtbügel verfügen. Dieser Bügel eignet sich vorzüglich zum Anklebmen von Prüfspitzen und dergleichen. Die Schalter sollten so herum eingesetzt werden, daß der Drahtbügel zum System hin zeigt, da man dann im geöffneten Zustand die Systemsignale messen kann. Wer keine Schalter benötigt, kann stattdessen feste Drahtbrücken einlöten, die ebenfalls als Testpunkte gut geeignet sind.

Noch ein Hinweis zum Auftrennen von Leitungen, die Betriebsspannungen führen: Werden Datenanschlüsse von ICs, deren Betriebsspannung „abgeklemmt“ wurde, von anderen „versorgten“ ICs angetrieben, so können ICs beschädigt werden. Deshalb: Wenn zu Testzwecken die Spannungsversorgung abgeklemmt wird, auch die Datenleitungen öffnen! ICs mit mehreren Versorgungsspannung nehmen es oft übel, wenn nur eine derselben abgeklemmt wird.

Damit die 64polige Adapterkarte für verschiedene Bussysteme verwendet werden kann, wurden die Masseleitungen zwar zusammengefaßt, aber noch nicht mit einem Anschluß des Steckers verbunden. Diese Verbindung ist nach Bedarf, am besten auf der Systembus-Seite, herzustellen. Wird bei bestimmten Bussystemen auch über eine dünne Mittelleitung eine Stromversorgung geführt, so muß diese gegebenenfalls verstärkt werden, da die Belastbarkeit im günstigsten Fall 1,5 A beträgt.

Platinen und Bausätze gibt es bei Regge Elektronik, Fessenfeld 57, 2800 Bremen.

* * R E L A I S - I N T E R F A C E * *

Nachdem im letzten INFO von Walter Zwickel eine ECB-Bus Erweiterung fuer unsere Rechner vorgestellt wurde, moechte ich an dieser Stelle ueber ein Projekt sprechen, an dem ich seit Februar 85 arbeite.

Es handelt sich hierbei um ein Relais-Interface ueber welches ich nun kurz berichten moechte.

Ganz am Anfang stand der Gedanke mit meinem Computer 'Etwas' steuern zu wollen. Nun dieses 'Etwas' sollte ein sogenannter Lehn-Roboter sein oder aber auch 'Etwas anderes'. Nun war es gerade dieses 'Etwas andere' worueber ich mir den Kopf zerbrochen habe.

Jeder der sich etwas in der Hardware auskennt wird sofort einwenden, dass es keines grossen Aufwandes bedarf um ein paar Relais mit dem Computer steuern zu koennen.

Doch wieviel Relais muessen es sein, damit ich einen Roboter und jemand anderer 'Etwas anderes' steuern kann ??
Und wie soll die Software aussehen, damit Jedermann das steuern kann, was er gerne moechte ??

Es gab viele Probleme in jenem strengen Winter fuer mich um ein moeglichst universelles Relais-Interface zu erstellen.

Die entsprechende Hardware war schnell ausgedacht und gefaellt. Nach einigen kleinen Basic-Programmen und den entsprechenden Aenderungen in der Hardware war der Hardware - Prototyp fertiggestellt.

Das Erstellen einer entsprechenden Software sollte sich allerdings als sehr viel schwieriger herausstellen. Anfaengliche Basic-Programme habe ich schnell ueber den Haufen geworfen, eine Steuerung der Karte von COBOL aus war auch nicht besonders sinnvoll. (Lediglich fuer die zahlreichen Masken)

Blieb also nur noch ASSEMBLER, zwar sehr aufwendig aber dafuer sehr schnell und Geschwindigkeit ist schon noetig um z.B 16 Relais innerhalb weniger Sekunden viele Schaltungen ausfuehren zu lassen.

Somit war das Problem der Wahl der Sprache geloesst - nun graute es mir vor der Ausfuehrung. Nach zwei Monaten hatte ich schon einige Programme beisammen. (Testprogramme fuer das Interface, Programme zum 'SCHALTEN' der Relais, Programme zum 'TASTEN' der Relais).

Mir war klar, dass einiges nicht besonders elegant von der Programmierung war, aber die Funktion die mir vorschwebte war da. - An dieser Stelle musste naturgemaess ein Rueckschlag erfolgen: ein kompletter Systemabsturz mit der Folge, dass alle Sourcen hin waren. Mir blieb nur noch ein etwas aeelterer Programmausdruck.

Nach diesem ersten Schreck habe ich mir nochmals ueber das gesamte Projekt Gedanken gemacht - das Ergebnis ist eine vollkommen neue Version (Version 2.1) der Software die wohl kaum noch Wuensche offen laesst.

Nach dem Motto: 'Software ist nur komplett mit Handbuch' gibt es zu dem Programm ein ca. 25 seitiges Manual.

Das Demo-Programm zum Relais-Interface kann ueber den Hartmut Obermann aus der Programm-Bibliothek bezogen werden, oder aber direkt bei mir, dann komplett mit Handbuch (bitte Diskette und 2.60 DM fuer Kopien beilegen).

Das Demo-Programm befindet sich auf einer Diskette ohne System, 40 Tracks, ss/sd. Zusaetzlich werden drei Schalt-Dateien fuer die 24-Std. Steuerung bzw. Kurzzeitsteuerung mitgeliefert. Diese Dateien koennen vom Programm eingelesen (A), Editiert (C) und ausgefuehrt (7) werden. Die Angaben in Klammern geben das entsprechende Programm an.

Noch einige Saetze zur Hardware:

Die Hardware ist zur Zeit 1. als Prototyp in Faedertechnik und 2. auf Platine aufgebaut. Die Platinenversion 1.0 muss nochmals ueberarbeitet werden, da die Leiterbahnen zu den Relais-Schaltkontakten zu duenn ausgefallen sind und sich bei Stromstaerken um 0.5 bis 1 Ampere in Rauch aufoesen.

Der Gesamt-Hardware Umfang wird sich auf vier Europa-Karten (mit Netzteil) bzw. drei Karten (ohne Netzteil) erstrecken.

Zum Abschluss noch einige (gekuerzte) Seiten aus dem Anwender-Handbuch:

I n h a l t s u e b e r s i c h t

1.0	Allgemeine Anmerkungen	Seite: 3
1.1	Software	Seite: 4
1.2	Hardware	Seite: 5
1.3	Technische Daten	Seite: 6
2.0	Demo - Version	Seite: 6
3.0	Steuer - Programme	Seite: 7
3.1	Info (Hard/Software)	Seite: 8
3.2	Test (Lauflicht)	Seite: 9
3.3	Status Port A / Port B	Seite: 10
3.4	DOS - Kommando	Seite: 11
3.5	Tasten / Schalten	Seite: 12
3.6	Uhrzeit setzen	Seite: 13
3.7	Schaltuhr	Seite: 14
3.8	Rueckkehr zum DOS	Seite: 15
4.0	Verwaltungs - Programme	Seite: 15
4.1	Schaltzeiten speichern	Seite: 16
4.2	Schaltzeiten laden	Seite: 16
4.3	Schaltzeiten editieren	Seite: 17
5.0	Hardware - Komponenten	Seite: 18
5.1	Aufbau/Funktion der Karten	Seite: 15
5.2	Anschluss - Plaene	Seite: 20

1.0 Allgemeine Anmerkungen

Wer hat sich nicht schon einmal ueberlebt mehr mit seinem Computer anzufangen, als die 12. Version von 'Invader' zu spielen ?!

Es gibt zahlreiche Moeglichkeiten den eigenen Computer fuer Steuerungs- und Ueberwachungsaufgaben einzusetzen, allein es fehlt das noetige Interface und die entsprechende Software um diesen Einsatz zu ermöglichen.

Die einfachste Steuerung ist wohl die zeitliche Ueberwachung von elektrischen Verbrauchern, die zu bestimmten Zeitpunkten ein- bzw. ausgeschaltet werden muessen.

Als Beispiele seien nur die Ueberwachung einer Zentralheizung (Pumpe, Brenner, Magnetventile fuer Gas und Wasser, Kaminklappe usw.) oder die Steuerung einer Modelleisenbahn genannt.

Es gibt aber auch Anwendungsfaelle, bei denen neben einer Schaltuhrfunktion auch die manuelle Steuerung von verschiedenen Verbrauchern gefordert wird. Ein einfacher Lehrrobotter mit vier Achsen wuerde acht Kaele zum Steuern benoetigen.

Die Arme eines Robotters muessten bei manueller Steuerung auf Tastendruck reagieren, zudem sollte eine Moeglichkeit der Speicherung von zeitlichen Bewegungsablaeuften und ihre Ausfuehrung gegeben sein. (sog. Kurzzeitsteuerungen)

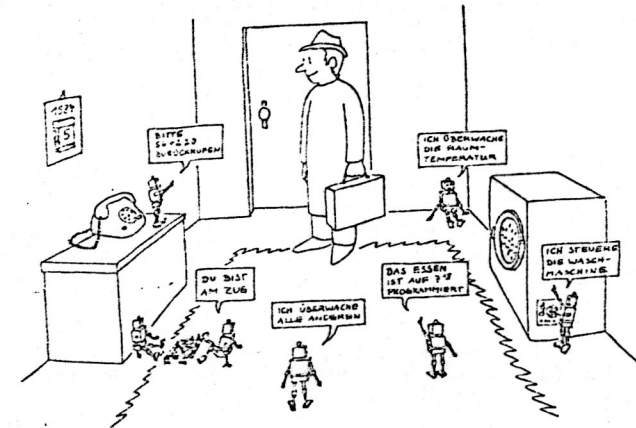
Neben der reinen 24-Stunden-Steuerung kann mit einer Kurzzeitsteuerung (also Start eines Schaltprogramms von einem frei wahlbaren Zeitpunkt aus) eine Modelleisenbahn mit mehreren Zuegen sehr komfortabel gesteuert werden.

z.B.	Prog.
10 Sek. nach Start	Zug 1 faehrt an
12 Sek. nach Start	Zug 2 faehrt an
15 Sek. nach Start	Zug 3 faehrt an
16 Sek. nach Start	Zug 1 stoppt
16 Sek. nach Start	Zug 4 faehrt an

USW.

Es gibt sicherlich noch viele andere Faelle, bei denen man mit einer Relais-Schaltsteuerung Verbraucher ueberwachen kann.

In der vorliegenden Form der Relais-Steuerung per Computer wurde besonders auf die Benutzerfreundlichkeit der Software geachtet. Sie ist so ausgelegt, dass man fast alle Anwendungen mit ihr abdecken kann, sei es eine rein manuelle Schaltsteuerung per Taster bzw. per Schalter, oder aber eine vollautomatische 24-Stunden Ueberwachung.



1.1 Software

Die Software ist vollstaendig in Assembler geschrieben um die geforderte Verarbeitungsgeschwindigkeiten zu erreichen (mehrere Schaltungen innerhalb einer Sekunde).

Der jeweilige Schaltzustand der Relais-Aussaenge ist jederzeit auf dem Bildschirm sichtbar, so dass bei Fern-Ueberwachung immer der Zustand der Verbraucher (bestromt/entstomt) am Bildschirm erkannt werden kann.

Wie bei vollautomatischen Steuerungen erforderlich gibt es auch hier das sogenannte 'NOT AUS', eine Funktion, mit der zu jedem Zeitpunkt alle Verbraucher ausgeschaltet werden koennen.

Zahlreiche Testfunktionen und die Moeglichkeit der direkten Ausfuehrung von DDS-Kommandos runden die Steuerungs-Programme ab.

Die Gruppe der Verwaltungs-Programme bietet eine gute Hilfe beim Erfassen oder Editieren der Schaltzeiten fuer die einzelnen Relais. Es steht ein Bildschirmeditor zum problemlosen Veraendern oder Neuerfassen von Zeiten zur Verfuegung. Auf fuef Bildschirmseiten stehen fuer 15 zu schaltende Relais insgesamt 270 Schaltzeiten zur Verfuegung.

Aufgerechnet auf ein einzelnes Relais kommen somit z.B. fuer die 24 Std. Schaltuhr 18 Schaltungen auf 24 Stunden (also neun mal Ein-Aus).

Benoetigt man weniger Relais-Aussaenge, da fuer aber mehr Schaltzeiten, so kann man mehrere Relais parallel schalten um somit die Anzahl der Schaltungen fuer einen Relais-Kanal zu erhoehen.

Die Schaltkontakte der Relais koennen mit 220 Volt und 1 Ampere belastet werden. Als Schalter stehen pro Relais 1 x um zur Verfuegung.

Der mechanische Aufbau der Relais-Karten sieht eine Verteilung des Steuer- und Lastteils auf drei Europa-Karten vor.

Auf der Steuer(Logik)-Karte befinden sich der Adress-Dekoder, Leitungstreiber, LED-Treiber, LED's und eine 280 PIO sowie einige weitere Komponenten.

Mit Hilfe der LED's wird zusätzlich zur Schaltzustands-Anzeige auf dem Bildschirm der Status der Relais optisch direkt auf der Karte angezeigt.

Die Verbindung Steuer(Logik) Karte und Computer geschieht ueber ein 24 adriges Flachkabel mit Anpress-Stecker. Das freie Ende des Kabels wird direkt mit dem Computer oder einer vorhandenen BUS-Platine verdrahtet.

Die Software arbeitet bereits mit der Logik-Karte. Fuer eigene Anwendungen kann man z.B. auf die Relais-Karten verzichten und die Steuerleitungen (TTL-Pegel) direkt verwenden. In diesem Fall erfolgt die Stromversorgung vom Computer aus. (Jumper auf der Logik-Karte fuer interne/externe Stromversorgung)

Die Relaiskarten sind wie folgt aufgebaut:

Karte 1 enthaelt neben acht Relais ein Netzteil, das nicht nur die Relais dieser Karte (Nr. 1 bis 8) versorgt, sondern auch die Relais der zweiten Karte (Relais Nr. 9 bis 16), sowie die Logik-Karte.

Die zweite Relais-Karte enthaelt neben den Relais ein freies Wrap-Feld fuer eigene Entwicklungen.

Die Relais-karten werden ueber 16-adrige Flachkabel mit Anpress-Steckern mit der Logikkarte verbunden.

Die von der PIO gelieferten Schaltsignale werden den Relais ueber Treiber-Transistoren zugefuehrt. Die Treiber-Stufen befinden sich auf den Relais-Karten. Die Schaltkontakte der Relais sind auf Klemm/Schraubboecke an der Stirnseite der Platine gefuehrt, so dass dort direkt Kabel bis 1mm Querschnitt angeschlossen werden koennen.

Die Relais-Karten bilden zusammen mit der Logik-Karte einen kompakten Sandwich-Aufbau bei Montage ueber Abstandsrollen. An der einen Stirnseite sind wie erwahnt die Anschlussboecke der Relais-Karten, sowie die LED's der Logik-Karte gefuehrt, die andere Stirnseite steht fuer die Anschluss-Kabel zur Verfuegung.

die Karten-Kombination kann daher bequem wahlweise in ein eigenes Gehauese oder aber in ein 19 Zoll Einschubrack installiert werden.

Logik-Karte:

- 2 80 PIO, Bus Treiber, Adressdekoder, LED-Treiber, 16 LED's, Codierbruecke fuer externe/interne Stromversorgung
- Anschluss Logik-Karte (-). Computer ueber 24 adriges Flachkabel mit Anpress-Stecker
- Signale der Pio (Port A und B) auf jeweils eine 16 pol. IC-Fassung gefuehrt

Relais-Karte 1

- 16 Relais mit Kontakt 1 x um, 1 Ampere, Treiber-Stufen
- Anschlussboecke fuer Kabel bis 1 mm Querschnitt
- Netzteil mit 5 Volt und 1.5 Ampere (fuer Logik-Karte, Relais-Karte 1 und Relais-Karte 2)

Relais-Karte 2

- wie Relais-Karte 1, Jedoch statt Netzteil freies Wrap-Feld

2.0 Demoversion

Die Treiber-Software ist in zwei Versionen (Demo bzw. Einsatz) verfuegbar.

Die Demo-Version unterscheidet sich von der Einsatz-Version dahingehend, das mit ihr keine Hardware angesteuert werden kann.

Alle sonstigen Funktionen sind vollkommen identisch. Mit der Demoversion koennen Schaltzeiten erfasst, veraendert, gespeichert und geladen werden.

Bei der Ausfuehrung dieser Schaltzeiten (des Steuerprogramms) werden alle durchgefuehrten Schaltungen auf dem Bildschirm angezeigt (wie auch bei der Einsatz-Version).

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 0000-37FF 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 0000 liegt, und CP/M lädt alle Programme, die unter ihm laufen, in den Speicher ab 100 hexadezimal. CP/M und alle Programme, die unter ihm laufen, brauchen einen durchgehenden Speicher von 0000 bis zum Ende des Speichers, wo das Betriebssystem steht.

Abgesehen von dieser Speicherabweichung ist die Modell III Hardware durchaus dazu geeignet, leicht modifizierte 8080 oder Z80 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-Speicher-bänke 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 0000-3FFF-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

EIGHT KILOBYTE DEVICE SELECT BY ROM ENABLE* AND A14

ROM ENABLE*	A14	DEVICE SELECTED
HIGH	EITHER	NONE
LOW	LOW	8 KILOBYTE PROM
LOW	HIGH	PROM, KBD, VIDEO

Table 2. Segment Selection in Bank Containing ROM

Eine Hardware-Lösung

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 RAMs 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.

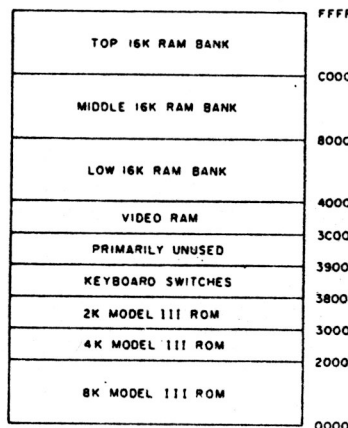


Fig. 1. Model III Memory Map

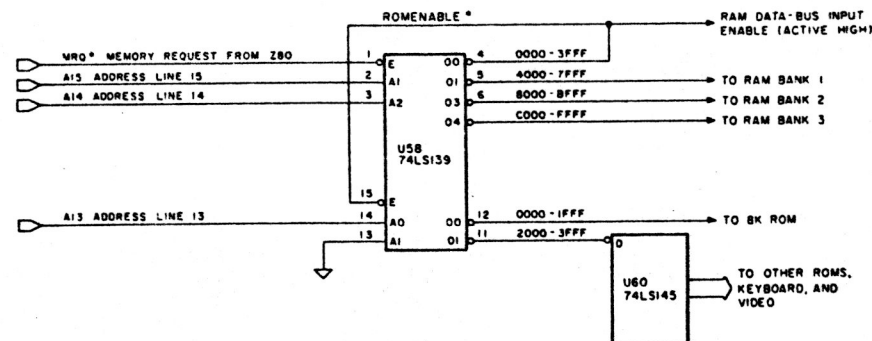


Fig. 2. Model III Address Decoder Circuit

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 Vorbearbeiten 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.

Ein neuer Bank-Wahl-Decoder

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 langsamer im Ansprechen 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 C000-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.

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.

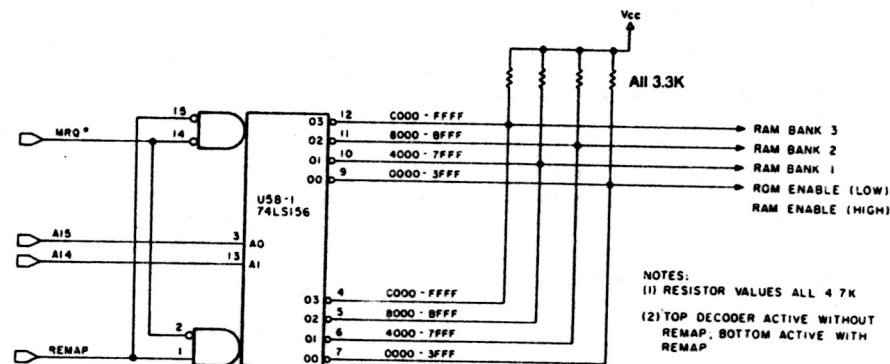


Fig. 3. Bank Decoder with External Request Line

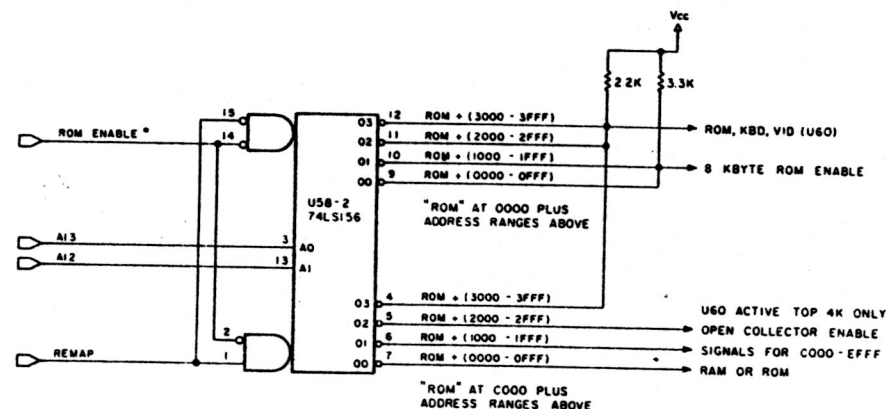


Fig. 4. Decoder for Bank Containing ROM, Keyboard, and Video

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 Einschaltssignale 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 744LS156 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)

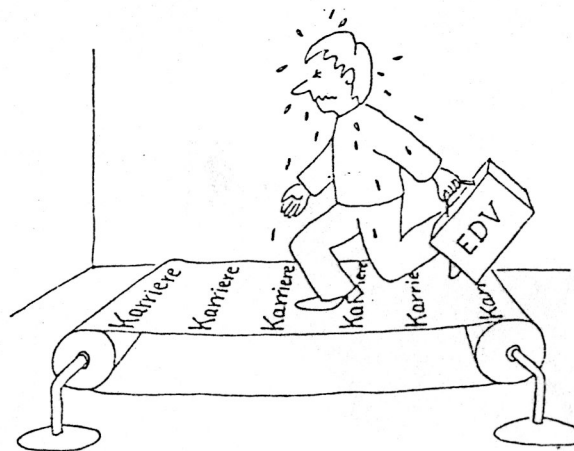
Das REMAP-Signal, das zwischen den beiden Speicher Konfigurationen umschaltet, könnte von einem Schalter außerhalb des Modell III-Gehäuses ausgehen. Aber der Schalter kann nur betätigt werden, wenn der Z80 ein Programm ausführt, das in den Speicher 4000-BFFF geladen wurde (oder im ROM ist). Sonst "vergißt"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

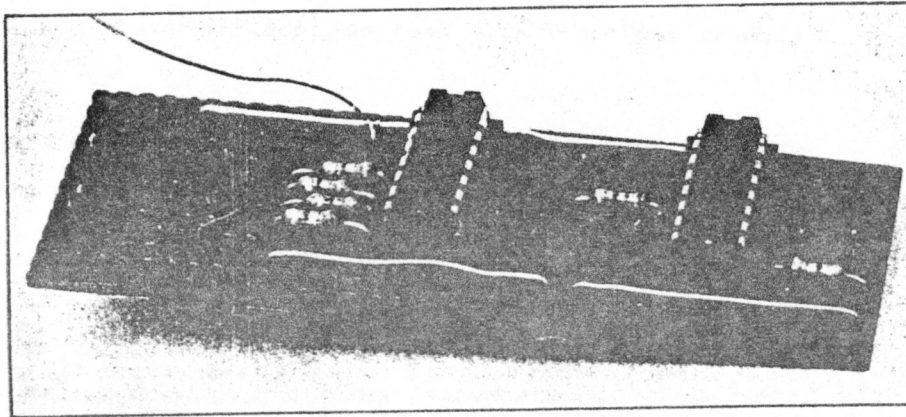


Photo 2b. Solder Side of Controller Board

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

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 CASX-Signal (CASX - Column Adress Strobe pin) der acht übrigen Sockel (für die acht zusätzlichen 4116s) parallel mit entsprechenden Steckern. Eine CAS-Leitung, die mit allen CASX 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 CASX zu erhalten, wird das daraus resultierende Signal mit dem Modell III CASX 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.

Außerste 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.

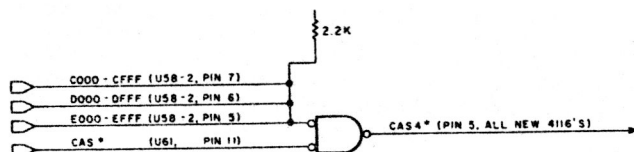


Fig. 6. Preliminary Logic for a New RAM Bank-Select Signal

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 angewählt.

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 vorhandenes 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.

Bild 1: Modell III Speicher Aufbau

Bild 2: Modell III Adressen Decoder Schaltkreis

Tabelle 1: 16K Bank-Wahl beim U58

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

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^{*1}. 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^{*2}. Als Speicherchips werden zwei 2114s (1024*4 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^{*2} 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.

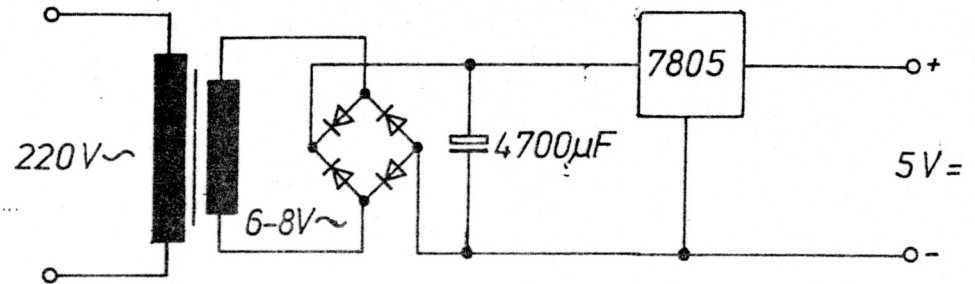
Nach dem Zusammenbau von Powersupply und Memory Board sollte als erstes eine genaue Sichtkontrolle der Schaltung erfolgen. Auf der Speicherplatine kann man außerdem mit einem Ohmmeter feststellen, ob die 5 Volt-Leitungen einen Kurzschluß haben. Erst nach diesen Kontrollen sollte man die Stromversorgung 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.

^{*1} Natürlich ist auch der Anschluß über den geplanten ECB-Bus realisierbar

^{*2} 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 ";
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.

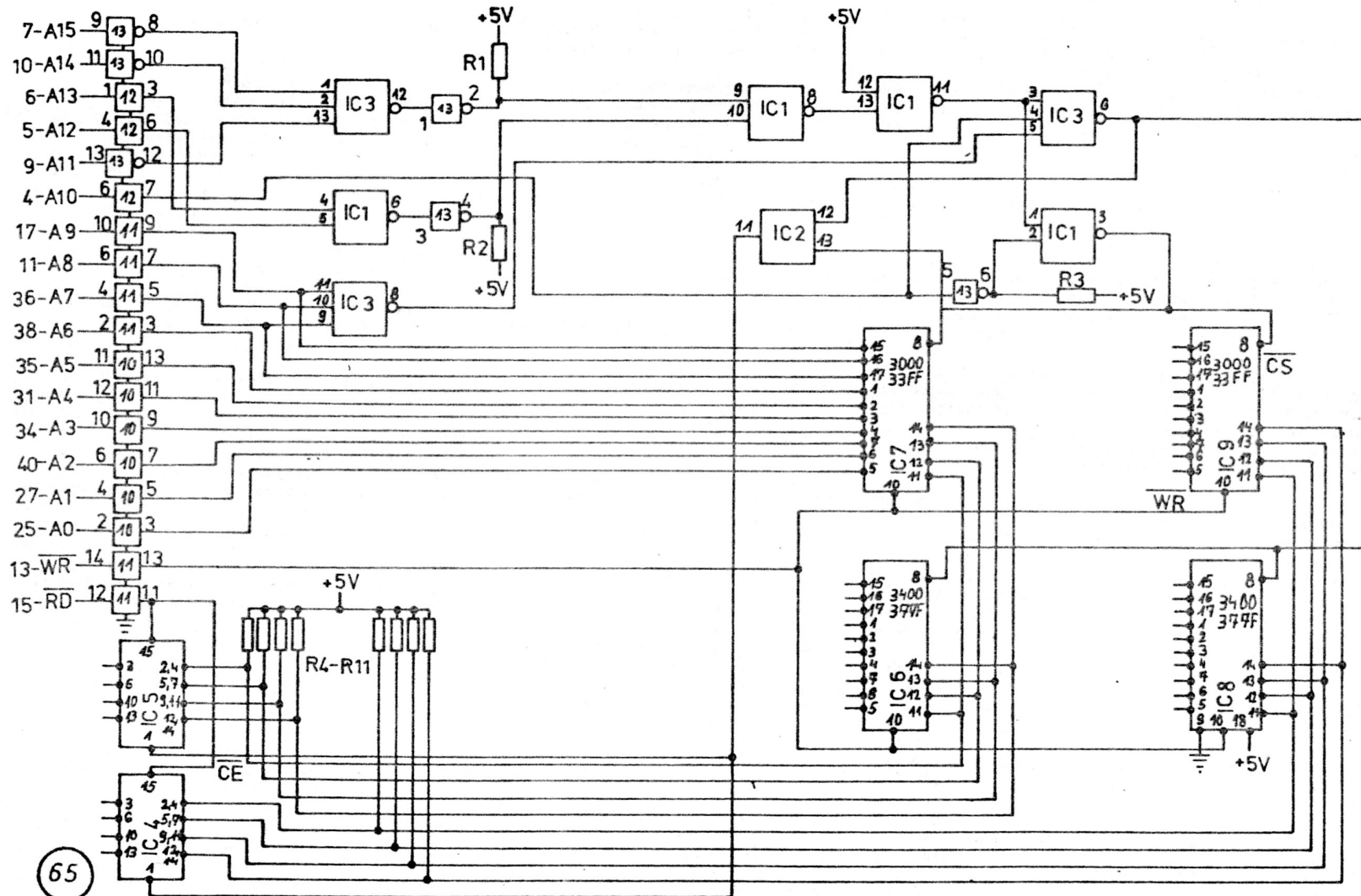
Die Zeichnung des Netzteils stammt von mir. Wie man feststellen kann, sind meine Fähigkeiten auf dem Gebiet des technischen Zeichnens nicht allzusehr ausgeprägt. Ich hoffe, man kann erkennen was gemeint ist.

Hartmut Obermann

San Diego – Sechs Kühlschränke, die hier in einer Oberschule aufgestellt werden sollten, waren mit Drei-Pol-Steckern ausgerüstet. In der Schule gab es jedoch nur Zwei-Pol-Steckdosen. Am nächsten Tag kamen zwei Elektriker. Der eine versah die Kühlschränke mit Zwei-Pol-Steckern. Unterdessen tauschte der andere die Steckdosen an der Wand aus und montierte überall Drei-Pol-Steckdosen.

Ein »völlig neues Frühstücksgefühl« verspricht eine amerikanische Elektrowarenfabrik den Käufern ihres Toaströsters, der je nach Wahl byzantinische, florentinische oder spätgotische Ornamente ins Brot brennt und überdies während des Röstens eine Mozartmelodie abklimpert.

Als der Lehrer einer Münchner Berufsschule die Prüfungsfrage »Was ist ein Lichtjahr?« stellte, antwortete der Prüfling: »Das ist die Stromrechnung für 365 Tage.«



Bauteile:

R1-R11 =	1KΩ
IC 1	= 74LS00
IC 2	= 74LS08
IC 3	= 74LS10
IC 4	= 8216
IC 5	= 8216
IC 6	= 2114
IC 7	= 2114
IC 8	= 2114
IC 9	= 2114
IC 10	= LS367
IC 11	= LS367
IC 12	= LS367
IC 13	= LS368

zu IC 8216:

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

Wie wohl inzwischen ein großer Teil von Euch mitbekommen hat, habe ich dauernd mit technischen Mängeln an meinem Genie I gekämpft. Der letzte Stand stellte sich so dar: die Laufwerke konnten sich nicht auf eine einheitliche Sourhaltung einigen und ein Wackelkontakt an der Video-Platine verwandelte den Bildschirm in ein Tumfeld aller möglichen Grafikzeichen, wobei nur ein kräftiger Schlag auf das Gehäuse Abhilfe schaffte. Entnervt beschloß ich gemeinsam mit meinem elterlichen Geldgeber, die gesammelten Neuheiten der Technik in Form eines Genie IIs zu erwerben.

Glücklicherweise fand sich ein Händler in der Nähe (nur ca. 60 km entfernt), der uns bei einer Besichtigung der Anlage diese gleich andrehte. Für vier Riesen durften wir einen Computer mit zwei eingebauten Slimline-Laufwerken je 80 Track DS/DD und einen Taxan-Monitor mitnehmen.

Zum Aufbau müssen nur die separate Tastatur und der Monitor werden an das massive (und dementsprechend schwere) Metallgehäuse angeschlossen werden. Die Anschlüsse sind absolut narrensicher, nur die Kabel leider nicht: das Tastaturkabel gehört zur Gattung der Flachkabel und ist dementsprechend inflexibel. Die Länge des Kabels eröffnet drei Möglichkeiten: die Tastatur hinter das Hauptgehäuse zu stellen, genau davor oder etwas rechts versetzt. Die erste Möglichkeit scheidet aus, und die beiden anderen sollte man auch vergessen, denn die Laufwerke liegen (im wahrsten Sinne des Wortes) auch rechts und damit ist das Laufwerk 0 mit davorstehender Tastatur nicht mehr erreichbar. Also das Kabel dehnen so weit es geht und die Tastatur nach links.

Aber wir haben ja noch ein Kabel: das vom Monitor zum Computer. Die Verbindung am Computer rechtfertigt den Slogan des Herstellers TCS: "Made in Germany" \Leftrightarrow Qualität, doch die andere Seite ist noch original Hongkong-made, was wohl erklärt, warum erst das dritte (!!) Kabel nicht auseinanderfiel. Allerdings erklärt es nicht die laschen Qualitätskontrollen (von denen später noch die Rede sein wird).

Fertig angeschlossen bietet sich ein tolles Bild: eine große, halbwegs flache Tastatur, ein wuchtiges Hauptgehäuse und ein flimmerfreier Bildschirm (Video-Ausgang verbessert); dazu der Sound eines leicht rauschenden Lüfters und der leise ratternden Laufwerke. Doch der Schein trügt etwas: die Tastatur erfüllt nicht alle Erwartungen. Quantität ging vor Qualität. Sehr schön die Zusatz Tasten: zwei Funktionstasten ergeben in Verbindung mit den Ziffern im abgesetzten Ziffernblock (leider ohne zweite Enter-Taste) sechzehn Funktionen. Reset wird über zwei gleichzeitig zu betätigende Tasten gegeben, die Umlaute sind gut erreichbar, im Gegensatz zu den Cursortasten zur Horizontalsteuerung im Ziffernblock.

Die Qualität läßt zu wünschen übrig: ähnlich dem C64 "schwimmen" die Tasten etwas wie in Summi, außerdem habe ich (subjektiv) den Eindruck, als ob sie "haken": die alte Genie-Tastatur war tipp-freundlicher...

Eine Sondertaste fehlt noch in der Aufzählung: die "LSP"-Taste (steht wohl für Lower - Speed), neben der rechten Shift-Taste. Sie rastet ein, wenn man sie drückt, und schaltet damit die Geschwindigkeit des Prozessors von 6 auf 1,77 Mhz zurück. Damit habe ich auch einen Hauptpunkt angesprochen, der dieses Gerät so attraktiv macht: die Geschwindigkeit. Scripsit ist unter "Höchstgeschwindigkeit" fast nicht benutzbar. Der Cursor springt nur so über den Bildschirm und ein einzelner Buchstabe kann kaum getippt werden. Also muß man sich umgewöhnen: schreiben mit 1,77 Mhz und alle zeitraubenden Aktionen (Replace, Blättern) mit 6. Selbst die Diskettenoperationen laufen etwas schneller ab, obwohl die Laufwerke natürlich hemmend wirken.

Offensichtlich ist die hohe Geschwindigkeit vor allem im Basic. Spiele erreichen leicht die Qualität der in Maschinensprache geschriebenen. Nur eines vermüßte ich sofort: den Sound. Kein Tongenerator lenkt vom "ernsthaften Arbeiten" ab. Vielleicht läßt sich der Ton über einen extern anschließbaren Recorder oder Lautsprecher ausgeben, aber dokumentiert ist diese Möglichkeit nicht.

Natürlich reicht die hohe Geschwindigkeit nicht aus, diesen Computer zu rechtfertigen. Also Punkt zwei: TCS ist einfach unserem Hardware-Zirkel unter Leitung von Walter Zwickel zuvorgekommen und hat das Steckkarten-System benutzt. Nach Öffnung des Gehäuses ist eine gähnende Leere sichtbar. In meinem Gerät sind noch fünf der zehn Steckplätze frei (Grundgerät ohne Floppy-Controller mit sechs freien). Bis jetzt bietet TCS folgende Steckkarten an: Grafikkarte (80x24 Zeichen, 480x192 Punkte), SID/FIQ-Karte (zwei serielle und zwei parallele Schnittstellen), RAM-Karte (192 Kb, höchste Speicherkapazität 256 Kb), Host-Karte (zwei Harddisks anschließbar), ROM-Karte (Erweiterung des ROM auf bis zu 4x32 Kb). Außerdem kann die Taktfrequenz auf 8 Mhz erhöht werden.

Übrigens: wer sich so etwas lieber mal kurz selbst zusammenlöten möchte, dem empfehle ich die Technische Beschreibung zum Genie IIs (39 DM, von Trommeschläger). Dort sind alle Platinen mit Bauteilen etc. auf ca. 100 Seiten (für mein Laienauge) ziemlich genau beschrieben.

Wie versprochen möchte ich noch einmal auf die Qualität eingehen. Natürlich wollte ich meinen Drucker nicht ungenutzt rumstehen lassen. Also besorgte ich mir ein Druckerkabel zum Preis von 25 DM bei meinem Händler (Materialkosten bei Conrad-electronic unter 50 DM). Doch mein Drucker souckte nur Müll aus und so schickte ich das Ding zurück. Das zweite Kabel verschwand angeblich auf dem Postweg, aber das dritte lief wenigstens, allerdings nur mit 7 Bit. Nun kam ich darauf, den Ausgang am Computer durchzuprüfen. Und siehe da: statt den versprochenen 8 kamen nur 7 Bit heraus. Ich kann noch nicht sagen, was das zu bedeuten hat, aber ich werde TCS einen netten Brief schreiben müssen. Nun will ich zum letzten Punkt kommen: der Software. In einem Zusatz-ROM befindet sich im Genie IIs ein tolles Software-Paket: Monitor, Texteditor und Assembler und ein kleiner Disassembler. Nach Aufruf wird dieses Paket in den oberen Bereich des Speichers gelegt. Allerdings fehlen hier die Diskettenbefehle, so daß ein Zusatzprogramm auf der mitgelieferten Systemdiskette zu finden ist, das diese beinhaltet. Doch dieses Super-Paket löst sich mit zunehmender Arbeitsdauer in Luft auf. Angeblich können Programme bearbeitet werden, die in den Bereich 5300-0000H laden. Doch auch bei Programmen, die diese Grenzen nicht erreichen, hängt sich der Monitor auf. Außerdem scheint der Assembler etwas gegen Source-Codes zu haben, die über ein NOF hinausgehen. Sind beim Start des Programmpakets noch andere DOS-Hilfsprogramme im Speicher, kann man herrliche Effekte erzeugen ("blindes" Eingeben von Filenamen ohne Cursor, toll für die Password-Freaks), von denen nichts im Handbuch steht.

Dieses Handbuch (besser gesagt deren zwei, für DOS und Basic) ist auch so 'ne Sache, natürlich auch die Technische Beschreibung. Angeblich können Controller und Laufwerke alles lesen, von 35/SD/SS bis 80/DD/DE, aber es ist mir mit keiner der angegebenen PDrive-Einstellungen gelungen, 40/SD/SS zu lesen. Auch die beschriebenen Prozeduren zum Booten von 40er-Disketten sind absolut wirkungslos. Das DOS-Handbuch weiß außerdem nur in einem Vorspann über das mitgelieferte GDOS 2.4 zu berichten, welches angeblich für alle Genie-Rechner geeignet ist. Das Handbuch selbst hört beim S-DOS 3.0b für den Speedmaster auf. Die letzte Auflage fehlt scheinbar noch. Dafür werden einige neue Programme vorgestellt: z.B. zwei Grafikpakete für hochauflösende Grafik bei Genie IIs und IIIs. Oder KEY24, ein Programm, das sich selbst nach HIMEM verschiebt und neben Druckertreiber einen Bildschirmeditor zur Verfügung stellt, der allerdings nicht sehr komfortabel ist, dafür aber im DOS, im Basic und in allen Programmen ohne eigene Eingabe-Routine zur Verfügung steht (da sollte sich Arnulf mit seinem HDOS noch mal hinterklemmen).

Andere Programme unterstützen den zusätzlichen Speicher (falls vorhanden) und die Funktionstasten (sehr einfach programmierbar). Bildschirm- und Tastatortreiber wurden abgeändert (steht da jedenfalls). Und scheinbar auch das Mini-Superzap, hier DDE (=Disk Daten Editor) genannt, denn es läuft nicht mehr. Im Modifiziermodus hängt es sich auf (wie vieles auf dieser Diskette).

Dann ist da natürlich noch die Kompatibilität: Im Basic geht es noch, obwohl Dancing Demon einige komische Sachen macht. Aber mit Assembler-Programmen sieht es schlimm aus. Auf dem Genie I vollkommen lauffähige Programme müssen über das Level II-Basic gestartet werden und die Abspeicherung von Highscores klappt auch nicht immer. Aber zum Laufen bekommen habe ich bis jetzt alle, wenn auch mit Komfort-Einbußen. Nur alles Bootbare kann ich vergessen.

Fazit: Trotz aller negativen Punkte möchte ich nicht zu meinem alten Genie I zurückkehren. Das flimmerfreie Bild, die Umlaut- und Funktionstasten, der Bildschirmeditor, die 1.4 Megabyte Diskettenspeicher (mit den alten SD/SS-Disketten!) und die Erweiterungsmöglichkeiten ohne das finger- und nervtötende Löten überzeugen mich von dem System. Ab und zu vermisse ich die Töne, aber die kann ich auch selbst machen, und schließlich war die Geräuschkulisse oft nervtötend.

Außerdem: wer möchte nicht Intruders mit 6 Mhz spielen. So schnell bin ich noch nie gestorben.

Nachlese

Kurzerklärungen für Programme

Hallo Leute, mir scheint, daß viele von Euch die gleiche Idee mit den Programmkurzerklärungen auch schon hatten.

Ich werde mich also in nächster Zeit über die Kurzerklärungsbibliothek hermachen und Euch in den nächsten INFO's die Resultate, in Form einer Aufstellung von Kurzanleitungen auf Karton anbieten.

Bis dahin könnt Ihr Euch aus der Clubbibliothek die Anleitungen zukommen lassen. Die Auflistung der entsprechenden File findet Ihr in diesem INFO.

Henn Ihr im Besitz von Anleitungs-File seit, die nicht in der Auflistung angeführt sind, seit bitte so nett und sendet diese an die Clubbibliothek.

Bis zur 1. kartonierten Kurzerklärung

J. Meuder

Im Artikel "Shift-Trick" wird angesprochen, daß sich das L2-Basic unserer Computer verändern läßt und dann in 2532-Eprom's abgespeichert werden kann. Wenn im Club jemand solche Änderungen vornehmen will und keinen Eprommer besitzt, kann er sich an mich wenden. Ich bin gerne bereit, kostenlos Eproms nach eigenen Wünschen zu brennen. Als Beispiel habe ich mir den teilweise hässlichen Zeichensatz meines Genie II völlig neu gestaltet.

Peter Spieß

Zum Kampf gegen die Rattenplage wird in Japan ein elektrisch betriebener Apparat angeboten, der in regelmäßigen Abständen »Miau«-Geräusche ausstößt und außerdem mit funkelnden Katzenaugen bestückt ist.

Heilige Hardware

MATTHÄUS, Markus, Lukas und Johannes würden staunen über das, was Pfarrer John Derksen in Kemprville in Kanada mit einem Computer zuwege bringt. Der Geistliche hat Bibelstellen, Bücher aus seiner Bibliothek und eine Fülle weiterer biblischer Informationen in einen Computer eingespeist, mit dem er seine wöchentlichen Predigten recherchiert. Anschließend schreibt er die Reden auf einem mit dem Computer gekoppelten Textverarbeitungsgerät.

In seinem Studierzimmer hat Derksen Geräte im Wert von Tausenden von Dollar installiert. Mit Hilfe seiner Computerdateien lokalisiert er schriftliche Dokumente, Bibelstellen und Informationen für besondere Feiertage und Anlässe. Das geht schneller als das herkömmliche Suchen und Nachblättern.

Derksen macht sich zunächst zu seinem Thema eine Gliederung und skizziert in großen Zügen den Inhalt. „Allmählich wächst die Predigt, und mit dem Textprozessor kann man ganze Absätze hinzufügen, tilgen oder verschieben“, sagt er. Für ihn gehöre das handschriftliche Ausarbeiten von Predigten der Vergangenheit an.

Canadian Press

Notausrüstung

MEIN Mann war ganz unglücklich, als er einsehen mußte, daß die Taschenrechner und Computer seinen guten alten Rechenschieber zum Museumsstück gemacht hatten. Er warf ihn jedoch nicht weg, sondern ließ ihn an einer Wand seines Büros hinter Glas anbringen mit der Aufschrift: „Bei Stromausfall Scheibe einschlagen!“

Y.R.

MIR GEFÄHRT der Mann, der immer wenn er in einem Einreiseantrag auf die Frage nach der Rassenzugehörigkeit stieß, nur das Wort „Mensch“ eintrug.

P.T.

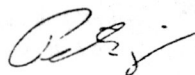
BÖRSE

WER HAT WAS -- WER SUCHT WAS --

In der heimischen Bastelstube entsteht zur Zeit ein Drucker-spooler mit max. 64k Speicher. Dieser Spooler ist für jeden Drucker verwendbar und wird zwischen Centronic's-Schnittstelle und Drucker gesteckt. Das Gerät arbeitet mit einer eigenen CPU und selbstständigen Monitorprogramm. Der Preis wird aber nur einen Bruchteil eines Industriegerätes kosten. Falls im Club Interesse besteht, bitte ich um Nachricht.

Weiterhin habe ich jetzt eine externe Grafikkarte mit einer Auflösung von 512 * 512 Bildpunkten lauffähig. Sie wird über die Ports 80H - 8FH angesprochen und ist daher universell für jeden Z80-Computer einsetzbar. Außerdem können unabhängig voneinander zwei Bildschirmseiten angesprochen werden. Die Umschaltung erfolgt über den Port 90H. Die Karte enthält einen eigenen ASCII-Zeichensatz und hat eine Zeichengeschwindigkeit von max. 1,5 Mill. Bildpunkte / Sekunde. Testbericht folgt in Kürze.

Mit besten Grüßen



Suche Kontakt zu Leuten,
die Spaß am Lösen von
Adventures
haben.

Gerald Schröder

Zu verkaufen !!!

1. Ein Expander EXP1 von RB-electronic
(Floppy- und Druckerschnittstelle)
2. Zwei Laufwerke Tandon TM100
40 Track / DS / DD, normale Bauhöhe
3. Ein Laufwerk Philips 3114
80 Track / DS / DD, 2/3 Bauhöhe

Alle Preise auf Anfrage bei: Gerald Dreyer
Am Speiergarten 8
6200 Wiesbaden/Bierstadt
☎ 06121 / 508218

Wer kann mir die Ausgaben August und November 1984 der
BOMicro fuer ein paar Tage ausleihen?

Klaus Hermann

Ich habe einen neuen und originalverpackten Magnetkartenleser
der Firma Redaktron zu verschenken(!). Der Neupreis ist 1200,-
Ich habe das Gerät vor ca. 2 Jahren fuer 280,- aus einer
Konkursmasse gekauft (inkl. 100 Magnetkarten mit 10.4 KB je
Karte). Der Leser ist einschliesslich Controller und Handbuch
verpackt und nach Anschluss einer Versorgungsspannung
betriebsbereit. Man muss sich allerdings ein
Anpassungs-Interface bauen und entsprechende Software
schreiben, also sehr viel Zeit investieren. Aus diesem Grunde
mochte ich das Gerät nur an einen wirklich interessierten
"Freak" abgeben. Da das Gerät und die Verpackung recht gross
sind, bitte ich um Erstattung der Porto-Kosten (ca. 20 bis 25
DM).

Harald Trapp

ist schon weg! P.S.

Wer hat folgende Sachen für 80 Track DS/DD:
Superutility (oder weiß die richtige Einstellung)
LDOS
Newdos

Wie lese ich auf 80 Track DS/DD stinknormale 40/SD/SS Disket-
ten? Ich bekomme nur "Lesefehler Inhaltsverzeichnis"

Wie boote ich auf Laufwerken mit 80/DS/DD Disketten mit 40
Spuren SD oder DD? Die Software-Lösung von Jens läuft bei mir
nicht.

Wer hat ein Programm, das den Start über das Kassettenbasic
nur simuliert, ohne über <Break und Reset> zu gehen? Das ROM-
Listing liefert leider nur Anhaltspunkte, denn viele Opera-
tionen sind scheinbar überflüssig (mir ist ihr Sinn jedenfalls
schleierhaft).

Gerald Schröder

PASCAL - PROBLEME =====

FRAGEN, FRAGEN, FRAGEKASTEN

Beim Arbeiten mit ALCOR-PASCAL Version 1.2 für das Modell I bin ich auf folgende Probleme gestoßen:

1. Offensichtlich erkennt der Compiler bei der Verwendung von Unterbereichstypen diese nicht immer richtig.
Um dies zu testen, habe ich folgendes kleine Programm zur Analyse eines einzugebenden Textes geschrieben und getestet:

```

(*****)
(*)
(*) PROGRAMNAME :   TEXTANAL
(*) DATUM :        3. 4. 1985
(*)
(*) BESCHREIBUNG :  DAS PROGRAMM LIEST EINEN TEXT EIN
(*) UND BESTIMMT DIE GESAMTZAHL DER ZEICHEN, DIE
(*) ANZAHL DER BUCHSTABEN UND DIE ANZAHL DER ZIFFERN.
(*) DER TEXT WIRD MIT EINEM $-ZEICHEN ABGESCHLOSSEN.
(*) DAS PROGRAMM WIRD DURCH EINGABE EINES '.' BEENDET.
(*)
(*)
(*****)
(*)
PROGRAM TEXTANAL (INPUT,OUTPUT);

VAR
  C1,C2   : CHAR;
  BUCHSTAB : INTEGER;
  ZIFFER   : INTEGER;
  GESAMT   : INTEGER;

BEGIN
  REPEAT
    BUCHSTAB := 0;
    ZIFFER   := 0;
    GESAMT   := 0;

    WRITELN ('ZUM BEENDEN DER EINGABE $ EINGEBEN UND ENTER !!');

    READ (C1);

    WHILE C1 <> '$' DO
      BEGIN
        IF (C1 IN ('A'..'Z')) THEN BUCHSTAB := BUCHSTAB + 1;
        IF (C1 IN ('0'..'9')) THEN ZIFFER := ZIFFER + 1;
        GESAMT := GESAMT + 1;
        READ (C1);
      END;

    WRITELN;
    WRITELN ('GELESEN WURDEN ',GESAMT,', ZEICHEN. ');
    WRITELN ('DAVON WAREN ',BUCHSTAB,', BUCHSTABEN UND ',ZIFFER,',
      ' ZIFFERN. ');

    READ(C2)
  UNTIL C2 = '.'

END.

```

Damit ergaben sich mit folgenden Eingaben die Ergebnisse (wobei man nach dem Abschluß der Eingabe des Strings mit (ENTER) nochmal (ENTER) für die READ(C2)-Anweisung drücken muß) :

ABCDEFGHIJ\$	10 ZEICHEN.	
GELESEN WURDEN	10 BUCHSTABEN UND	5 ZIFFERN.
DAVON WAREN		
ABCDEFGH\$	8 ZEICHEN.	
GELESEN WURDEN	8 BUCHSTABEN UND	5 ZIFFERN.
DAVON WAREN		
ABC\$	3 ZEICHEN.	
GELESEN WURDEN	3 BUCHSTABEN UND	1 ZIFFERN.
DAVON WAREN		
A\$	1 ZEICHEN.	
GELESEN WURDEN	1 BUCHSTABEN UND	1 ZIFFERN.
DAVON WAREN		
ABCDEFGHIJKLMN\$	14 ZEICHEN.	
GELESEN WURDEN	14 BUCHSTABEN UND	6 ZIFFERN.
DAVON WAREN		
1234567890\$	10 ZEICHEN.	
GELESEN WURDEN	0 BUCHSTABEN UND	10 ZIFFERN.
DAVON WAREN		
!"#\$%&'()*:-\$	10 ZEICHEN.	
GELESEN WURDEN	0 BUCHSTABEN UND	0 ZIFFERN.
DAVON WAREN		

2. Die Library mit den externen TRS-Funktionen bzw. Prozeduren (TRSLIB/OBJ) enthält wohl Fehler, da Programme, die sie benutzen, nach dem Linken nicht mehr einwandfrei funktionieren.

3. Der Overlay-Compiler steigt immer mit Fehlermeldungen aus, obwohl es sich um einwandfreie Programme handelt.

Falls jemand Patches kennt, um diese oder auch andere Fehler zu beheben oder aber diese Probleme mit dem ALCOR-PASCAL nicht hat, so soll er sich doch bitte bei mir melden.

Wer arbeitet mit
L I S P ???
Bitte bei Josef Konrad
melden !!

Wer hat Interesse daran,
daß die Inhaltsverzeichnisse
der 80 MICRO
im INFO abgedruckt
werden sollen?
Ulrich Böckling

Zur Zeit sind Foto-Kopien
dieser Verzeichnisse
bei Josef Konrad
erhältlich.

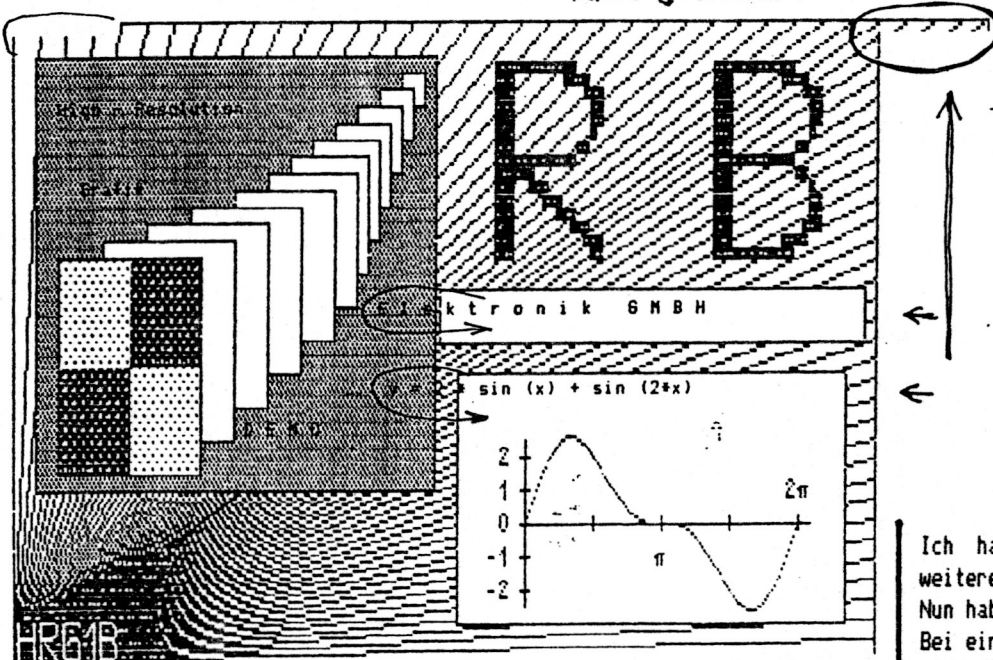
Wer kann helfen!

Ich bin nun glücklicher Besitzer der HRG1b von RB-Elektronik und möchte natürlich Graphiken erstellen. Dazu habe ich folgendes Problem. Nach dem Laden des Treiberprogr. von RB und dem nachladen des Basicprogr. "Graflix/BAS" aus der C.Persönlich Nr.21 v.3.10.84, lassen sich die Graphiken einwandfrei erstellen, nur ich kann die erstellten Graphiken nicht Speichern und Laden. Der Computer bleib hängen.

Wer kann mir auch noch in dem Progr. Graflix stehenden Druckerzeilen von ITOH auf EPSON RX umschreiben. * oder NEC??
LPRINTCHR\$(27)"E"CHR\$(27)"T16"CHR\$(27);
LPRINTCHR\$(27)"S0384";

Diese beiden Programme habe ich nun auch noch auf das hervorragende H-DOS von Arnulf Sopp kopiert. Mit der JKL Funktion besteht nämlich die Möglichkeit die HRG mit auszudrucken. Beim Druck habe ich aber folgende Ergebnisse. (siehe Abbildung) Wer kann hier helfen.

Hans-J. König



Wie ist es möglich Sonderzeichen (speziell Umlaute) in Filenamen zu verwenden? Das NEWDOS lässt ja nur "reine Buchstaben" bzw. ab der 2. Stelle auch Ziffern zu.

Umlaute für Filenamen sind sicher interessant, da sich dadurch die Filenamen genauer angeben lassen und man durch weitere Zeichen mehr Codierungsmöglichkeiten hat.

Jens Neuder

??? Frage ???

Ich möchte mir eine CP/M - fähigen Computer zulegen. Trotzdem will ich nicht von der TRS 80 - Linie abweichen. Aus diesem Grund suche ich Leute die entweder:

- ihr TRS 80 Model 1 auf 64k RAM "aufgebohrt" haben (zum Erfahrungsaustausch) oder
- ein VideoGenie I oder II mit 64k RAM zu verkaufen haben.

Vor allem bin ich an einer Anleitung zur Erweiterung des TRS 80 Model 1 auf 64k interessiert. Schaltpläne für einen Mapper besitze ich bereits.

Hartmut Overmann

Wer hat Erfahrungen mit einem VOICE SYNTHI ???

Wie wird er eingebaut ???

Wer kennt Bezugsquellen ???

Lohnt sich der Kauf ???

Patrick Perschbach

Wer hat Erfahrungen oder Bauanleitungen über einen "Lichtgriffel" für TRS-80 Mod. I ?

Ist der Light-Pen auch für die HRG verwendbar ?

Jens Neuder

Ich habe seit längerem die Groß-/Kleinschreibung sowie weitere Tasten zur Erreichung der Sonderzeichen eingebaut.

Nun habe ich damit folgendes Problem :

Bei eingeschalteter Kleinschreibung (im NEWDOS oder Basic) sind die Kleinbuchstaben der Umlaute nur über SHIFT zu erreichen. Alle anderen Buchstaben werden aber ohne SHIFT Klein dargestellt -wie es auch sein sollte-.

Bei Arbeiten mit Textverarbeitung oder anderen Betriebssystemen taucht der Fehler nicht auf.

Wer kennt die richtige Einstellung bzw. den Zap um Umlautkleinbuchstaben ohne SHIFT zu erzeugen ?

Jens Neuder

Bierlexicon

S 0166 A 01 /BAS

Mit Hilfe dieses Programms kann man sich ausfuehrlich ueber die deutschen Biersorten informieren, ohne dabei Gefahr zu laufen, wegen Alkoholismus seine Stellung zu verlieren.

Fuenfzehnerspiel

G 0167 A 03 /BAS

Wer kennt nicht das unterhaltsame Spiel, bei dem es gilt fuenfzehn in Unordnung gebrachte Zahlen in einem 4*4 grossen Quadrat durch verschieben der leeren Stelle wieder zu ordnen.

LISP

S 0168 A 03 /BAS

Diese Implementierung der Sprache fuer die Programmierung von Problemen der kuenstlichen Intelligenz entstammt der Micro 80 Maerz 83 S.176

LOCEDIT

S 0170 A 02 /BAS

Das Programm entstammt der Micro 80 und ermittelt die genaue Stelle, an der ein Syntaxfehler in einer laengeren BASIC-Zeile auftrat. Das Programm ist beim Editieren von Programmen sehr hilfreich!

MODEM via Cassettenport

S 0171 B 05 /BAS

Dieses Programm (Micro 80) ermoeoglicht mit Hilfe einer kleinen Zusatzschaltung den Betrieb eines Modems an der Kassettenschnittstelle des TRS80 M1. Man erspart sich also die Anschaffung einer RS232.

PILOT

S 0172 A 03 /BAS

Quelle: Micro 80. Dieses Programmpaket (3 Progr.) implementiert nicht nur einen PILOT-Interpreter sondern stellt gleich einen kurzen Einfuehrungskurs fuer diese Sprache dar.

RX 80 - Einsteller

S 0173 B 01 /CMD

Mit diesem Programm kann man sehr leicht die Parameter (Schriftart, Randeinstellung usw.) des EPSON RX 80 einstellen. Das Programm ist ein compiliertes BASIC-Programm.

Telefonwaehler

S 0175 B 03 /BAS

Das Programm ermoeoglicht, nach einem geringfuegigen Eingriff in das Telefon, mit dem Computer Telefonnummern zu waehlen. Autor: J. Herl

User Definable Keys

S 0176 B 05 /CMD

Mit diesem Programm (Quelle: ELCOMP) kann man die meisten der Tasten der Tastatur mit DOS- und BASIC- Befehlen belegen.

77

Unnummer

S 0177 B 02 /CMD

Das Programm macht, vom BASIC aus geladen und mit "NAME" aufgerufen, alle nicht als Sprungadressen benoetigten Zeilennummern zu 0 und erschwert somit die Veraenderung von Programmen durch Unbefugte.

QUADRATO

G 0178 A 01 /BAS

Das Programm verwaltet das Spielbrett fuer das bekannte Quadratospiel. Es ermoeoglicht nicht Quadrato gegen den Computer zu Spielen! Quelle: Homecomputer

SuperTape

S 0180 B 05 /CMD

Supertape ist ein Programm zur schnellen Speicherung von Programmen und Daten auf Tape (siehe Clubinfo Nr.7). Quelle: c't

Westwaerts 1847

B 0181 A 03 /BAS

16

Ein gutes Adventure-aehnliches Spiel mit Grafik.

CMDPATCH - BASIC Erweiterung

S 0182 B 03 /CMD

Dieses Programm erweitert den BASIC - Befehlssatz um die Befehle: CLIC K, FLASH, MELT, RVS und PIEP. Quelle: ELCOMP 4/5 84

DEFINPUT

S 0183 B 03 /CMD

Dieses Programm erlaubt es, den BASIC-Befehl Input mit Parametern (max . Eingabelaenge usw.) zu versehen.

SPOOLER

S 0184 B 03 /CMD

Ein 4k RAM-Spooler der die Programmausfuehrung von Programmen mit viel Textausgabe erheblich beschleunigt. Da der Sourcecode vorhanden ist, kann der Spooler leicht veraendert (Groesse usw.) werden.

TOTO

S 0185 A 03 /BAS

Ein Programm, mit dem eigentlich jeder zum TOTO-Millionaer werden muesste! Autor: Guenther Wagner

INPUT

S 0186 B 03 /CMD

Das Programm erfuehlt die gleichen Aufgaben wie das Programm DEFINPUT (S0183B03/CMD) jedoch mit anderem Syntax. Autor: Gerald Schroeder

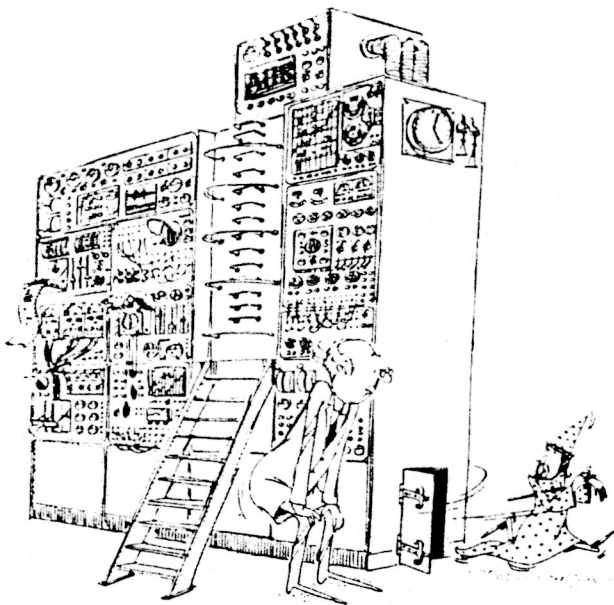
78

Aehnlich dem Adventure Powerplant jedoch eingedeutscht, erheblich erweitert und mit Hilfsfunktionen versehen von Gerald Schroeder

TRANSMUT

S 0189 A 01 /BAS --

Dieses Programm erleichtert die Aenderung von BASIC-Programmen. Es ermöglicht z.B. die globale Aenderung von Variablenamen und das Auffinden von BASIC-Befehlsworten.



»Feterabend!«

CLUB 80 - Programmbibliothek exklusiv

Heute möchte ich euch wieder ein paar Programme aus der Bibliothek ganz besonders ans Herz legen.

Da wäre als erstes einmal die wirklich sehr gute Übersetzung des Superutility+ 3.2 - Handbuches von Gerald Schröder zu nennen. Sie ermöglicht es auch dem, der englischen Sprache weniger mächtigen, Besitzer der Superutility diese voll auszunutzen. Die Übersetzung liegt als Scripsit-File vor und kann so von jedem selbst ausgedruckt oder direkt am Bildschirm gelesen werden. Das Handbuch hat übrigens ca. 70 Seiten!!!

Weiterhin wären zwei Programme zu nennen, die schon länger in der Bibliothek vorhanden sind. Es handelt sich dabei um EDITOR (S0007A01/BAS) und AktienTabelle (S0018A02/BAS), beide von unserem Clubgründer Günther Wagner. Diese Programme wurden von Günther überarbeitet und sind nun in ihrer neuesten Version zu haben.

Zum Thema Aktienverwaltung kann ich noch berichten, daß unser Mitglied Ulrich Böckling ein sehr gutes Programm zur Lösung dieses Problems geschrieben hat. Es ist sehr umfangreich und mit vielen Hilfsfunktionen gespickt. Leider läuft es nur mit einer Highresolution Grafik-Erweiterung. Wer es haben möchte, sollte sich bitte beim Ulrich direkt melden. Er bekommt dann nämlich noch die Aktienkurse einer Unmenge von Firmen aus den letzten 1 1/2 Jahren mitgeliefert.

Zu guter Letzt noch einmal ein Hinweis zum inzwischen abgeschafften Punktesystem. Immer wieder schreiben mir Mitglieder, ich solle ihnen ihren aktuellen Punktestand mitteilen. Hier also noch einmal der Hinweis: Das Punktesystem ist abgeschafft!!! Jedes Mitglied kann so viele Programme bestellen wie es mag. Dies kann z.B. so weit gehen, daß man mir eine entsprechende Anzahl Disketten (die Bibliothek umfaßt z.Z. ca. 5 DS/DD/80 Track-Disketten) zuschickt und dann die komplette Bibliothek zurückbekommt. Trotzdem sollte natürlich niemand vergessen, seine selbstgeschriebenen oder abgetippten Programme in die Bibliothek einzubringen.

Das wars mal wieder. Ich hoffe, ich höre bald mal von Euch, Euer

Karlmut Obermann