nascom journal

Zeitschrift für Anwender des NASCOM 1 oder NASCOM 2

2. Jahrgang · Fabruar 1982 · Ausgabe 2

Herausgeber:
MK-SYSTEMTECHNIK Michael Klein · Pater-Mayer-Straße 6 · 6728 Germersheim/Rhein
Telefon (0 72 74) 27 56 · Telex 0453500 mks d

MK-Systemtechnik Thomas Gräfenecker · Kriegsstraße 164 · 7500 Karlsruhe · Tel. 07 21 - 2 92 43
MK-Systemtechnik Michael von Keitz · Pfaffenberg 4 · 5650 Solingen 1 · Tel. 0 21 22 - 4 72 67

Der Heftpreis beträgt DM 4,—. Ein Abonnement erhalten Sie für DM 48,— im Jahr. Dafür bekommen Sie 12 Hefte pro Jahr, bzw. 10 Hefte (zwei dicke Doppelausgaben).
Die Autoren sind für den Inhalt ihrer Beiträge selbst verantwortlich.

INHALT

nascom journal

INTERN

Liebe Leser,
nicht nur Nostradamus kann Voraussagungen machen, auch wir scheinen in die Zukunft blicken zu können; denn wie Sie sich erinnern, haben Sie die letzte Ausgabe des Journals, wie vorausgesehen, sehr verspätet erhalten. Hier nun unser Blick in die weitere Zukunft: da Sie ja Anspruch auf 12 Hefte bzw. eine entsprechende Anzahl von Doppelheften haben, wollen wir versuchen, durch die Ausgabe eines Doppelheftes wieder den Anschluß an die gewohnten Erscheinungstermine zu gewinnen. Sie erhalten diese (Februar-)Ausgabe also im März, und gleich darauf Ende April die Doppelausgabe 3/4. Als Redaktionsschluß haben wir uns den 10.April vorgenommen. Wir wären sehr dankbar, wenn wir vor diesem Termin noch eine Menge interessanter Beiträge erhalten würden; besonders Hardware und Anwenderprogramme. Bisher liegen uns sehr viele Spiele vor, die zum Teil aus unserem Wettbewerb hervorgingen. Viele von Ihnen können sich jetzt schon darauf freuen, es sind einige sehr ansprechende Sachen dabei (Grafik!).
Wenn es zeitlich klappt, können wir auch eine Beschreibung für den Anschluß einer Neckermann bzw. Quelle Typenradschreibmaschine von Günter Kreidl erwarten. Weiterhin eine schnelle Tastaturabfrage von Josef Zeller und noch einiges mehr. Aber bevor Sie an die nächste Ausgabe denken, wollen Sie sich bestimmt zunächst einmal mit diesem Heft beschäftigen. Hier hoffen wir, daß wieder für jeden etwas dabei ist. Besonders möchten wir auf die Grafik-Routine von Michael Bach hinweisen: vielleicht bildet sie den Anstoß für weitere interessante Grafikanwendungen in Maschinencode, vielleicht auch als Userprogramm in BASIC.
Hier noch einmal der Hinweis auf drei "Rundläufe", die wir augenblicklich auf Leserwunsch anbieten: YATZI, FORMAT und "SCHRÖDER-SPIELE". Bitte melden Sie sich rechtzeitig an, damit wir die Arbeit nicht mehrmals machen müssen.
Drücken Sie die Daumen, daß von nun an der Erscheinungstermin am Monatsende eingehalten werden kann; wir tun auch unser Bestes.
Ihr Günter Böhm

Uns liegt ein kleiner LISP-Interpreter vor, den Michael Bach nach einer Vorlage in "Dr. Dobb's" an den NASCOM angepaßt hat. (Was heißt hier"kleiner"? Das Ding umfaßt 16 K im Assembler ! G.B.)
LISP (von "LISt Processing") ist eine ganz eigenartige Programmiersprache, deren wichtigste Eigenschaft vielleicht die Gleichwertigkeit von Daten und Programmen ist. LISP-Programme können sich dadurch selbst verändern und sogar Programme entwikkeln und eignen sich damit vor allem zur Simulation von Lernvorgängen ("Artificial Intelligence").
Leider ist der Interpreter in der vorliegenden Form kaum zu gebrauchen. Daher die Frage an die Leser des Journals: Besteht Interesse an einem solchen Interpreter? Wer hat schon einmal mit LISP gearbeitet und/oder möchte an der Weiterentwicklung des vorliegenden Interpreters mitarbeiten?
Ein schöner einführender Aufsatz über LISP erschien übrigens in Heft 1 und 2/82 der Zeitschrift "ELEKTRONIK".
Ihr Günter Kreidl

LESERBRIEFE

Nachdem ich im Dezember die Hefte 8-12/81 bekommen habe, muss ich der Redaktion (und den sich immer mehr engagierenden Lesern) meinen Dank fuer die sehr interessanten Artikel aussprechen. Im Heft 11/12 waren es gleich 2 Beitraege, die mich dazu brachten, auch mal etwas zu 'leisten'.
Als erstes war da die Umschaltung des NASCOM1 auf 2400 Baud. Ich verwende diese Geschwindkeit schon seit ca. 1 Jahr auf meinem NASCOM2. Die Umschaltung auf 2400 Baud ist in ca. 5 Minuten erledigt. Zuerst sind die Testpunkte 4,5 und 20 miteinander zu verbinden und danach nur noch die Schalter LSW2/2 und LSW2/5 in die Stellung 'up' zu bringen. Wenn Sie nun noch eventl. den Aufnahmepegel etwas erhoehen, sollte die Sache richtig (und mit nicht mehr Lesefehlern als vorher) funktionieren. Zum Kopieren Ihrer alten Programme von 300/1200 Bd. auf 2400 Bd. lassen Sie einfach den Schalter LSW2/5 in der Stellung 'down', lesen das Programm ein und schreiben es wieder neu auf's Band. Wollen Sie nun einen Verify Durchgang machen, legen Sie LSW2/5 nach 'up' (aber nicht vergessen, ihn wieder zurueckzustellen).
Als zweites fiel mir das Programm QUEST von Herrn Bach ins Auge. Dieses Spiel habe ich nun umgeschrieben und auf 110(!) Kammern/Orte erweitert. Es benoetigt nun allerdings 32k Speicher und noch den Bereich von 0CA0H bis 0FA0H. Ich habe es mal mit auf diese Kassette geschrieben plus eine kleine Anleitung dazu. Wer daran Interesse hat und sich die Eintipperei der 950 Statements sparen will, kann mir eine Kassette zusenden (Rückporto bitte nicht vergessen) und ich kopiere es dann, allerdings kann ich nur NASCOM2 Format. Geben Sie dann bitte auch die gewuenschte Baudrate an (300/1200/2400), 300 Baud nur, wenn's absolut nicht anders geht, ich brauche dann naemlich ca. 1 Std. um das Programm abzuspeichern und zu ueberpruefen.
Ausser diesem Spiel befinden sich noch 4 weitere mit Anleitung auf der Kassette:
1.) AWARI, ein afrikanische Strategie-Spiel.
2.) MONDLANDUNG
3.) GELDSPIELAUTOMAT mit 'richtigen' Walzen
4.) STOCK-CAR, ein Action-Spiel mit Bewegung

Soweit fuer heute, und ich hoffe, wenn ich meine ersten Gehversuche mit FORTH hinter mir haben, auch dort mal ein Programm veröffentlichen zu können.
Wolfgang Schroeder, [redacted] Reutlingen

Vielen Dank, Herr Schröder! Wir veröffentlichen gerne obige Spiele. Die Arbeit des Kopierens können wir Ihnen aber abnehmen: alle Programme werden mit Beschreibung auf Cassette im NASCOM 1 oder NASCOM 2 Format als "Rundlauf" angeboten. Interessierte Leser sollten sich mit dem Stichwort "UMLAUF Schröder-Spiele ....Format" bei der Redaktion in nächster Zeit melden. Red.

Es ist sehr erfreulich, dass wir fuer unseren NASCOM eine solch schoene Zeitschrift haben. Um so unverstaendlicher ist es, dass manche Leute sich so negativ aeussern. Dem einen ist sie zu hoch, dem anderen zu nieder, ich finde so wie sie im Moment ist, ist sie gut durchwachsen und bringt jedem etwas. Auch die neue Gestaltung des Heftes ist eine gelungene Sache.
Unter der Rubrik "Bemerkungen zu Leserbriefen" finde ich es unschoen, sich so abweisend zu verhalten dem Herrn Mombaur gegenueber. Wenn sich jemand an einer Sache so begeistert, warum nicht? Das Spiel "Seeschlacht",das er gebracht hat ist wirklich eine Meisterleistung.
Nun nochwas: Warum wird denn immer an unserem lieben NASCOMPL herumgemeckert? Er lockert die doch manchmal trockene Materie etwas auf, und wem das nicht gefaellt,der moege eben darueber hinwegsehen.
Anbei finden Sie des weiteren auf der Cassette das praktische Anwenderprogramm des Quicksort, vielleicht kann man sowas bringen unter dem Motto "Von dem Praktiker fuer den Praktiker". Es waere von Vorteil, wenn es so eine Rubrik gaebe mit Programmen,die man im taeglichen Leben anwenden kann. Dies sollen keine Superprogramme sein, sie sollten lediglich die universelle Einsetzbarkeit eines Computers aufzeigen. Es ist auch zu bedenken, dass sich unter uns viele Newcomer befinden,die gerade ueber einfache und ausbaufaehige Programme erfreut sind.
Heinrich Auge, Künzelsau

Sortier-Programm

von Heinrich Auge

```
1 REM * SORTIERPROGRAMM MIT      *
2 REM * "QUICKSORT" unter       *
3 REM * Verwendung der Routine  *
4 REM * nach   W.Mayer-Gürr  von *
5 REM * Heinrich Auge, Künzelsau *
100 CLS
110 B$="**Quicksort** Eingabe mit '.' beenden!"
120 FOR I=1 TO LEN(B$)
130 POKE 3020+I,ASC(MID$(B$,I,1)):NEXT
140 CLEAR(2000):N=1
150 DIM I$(2000),A(2000),S(30,2)
160 PRINT N;:INPUT I$(N)
170 IF I$(N)="."THEN200
180 N=N+1
190 GOTO160
200 N=N-1:CLS
210 FOR I=1 TO N:PRINT I;I$(I)
220 IF I/14=INT(I/14) THEN GOSUB770
230 NEXT I
240 GOSUB770
250 IF X$="N" GOTO240
260 INPUT"Sortieren ab Stelle....:";Q
270 CLS:SCREEN 17,5:PRINT"Bitte Warten!"
280 FOR I=1 TO N:A(I)=I:NEXT
290 S1=1:S(1,1)=1:S(1,2)=N
300 L=S(S1,1):R=S(S1,2):S1=S1-1
310 I=L:J=R
320 H=A(INT(L+R)/2)
330 IF MID$(I$(A(I)),Q)<=MID$(I$(H),Q)THEN370
340 IF I<=R THEN370
350 I=I+1
360 GOTO330
370 IF MID$(I$(A(J)),Q)>=MID$(I$(H),Q)THEN410
380 IF J<=L THEN410
390 J=J-1
400 GOTO370
410 IF I<J GOTO430
420 Z=A(I):A(I)=A(J):A(J)=Z:I=I+1:J=J-1
430 IF I<=J GOTO330
440 IF(R-1)<=(J-L) GOTO490
450 IF L>=J GOTO470
460 S1=S1+1:S(S1,1)=L:S(S1,2)=J
470 L=I
480 GOTO520
490 IF I>=R GOTO510
500 S1=S1+1:S(S1,1)=I:S(S1,2)=R
510 R=J
520 IF R>L GOTO310
530 IF S1<>0 GOTO300
540 CLS:INPUT"Mit Drucker J/N..:";A$
550 IF A$="J"THEN580
560 IF A$="N"THEN600
570 GOTO540
580 POKE3084,0:DOKE4100,-10201:A=USR(0)
590 PRINT CHR$(1)
600 PRINT
610 FOR I=1 TO N:PRINT I$(A(I)):NEXT
620 IF A$="J"THEN PRINTCHR$(4):DOKE3189,1922
630 IF A$="J"THEN DOKE3187,1919
640 INPUT"Weitere Ausgabe..(J/N):";A$
650 IF A$="N"THEN670
660 GOTO540
670 INPUT"Neusortierung....(J/N):";A$
680 IF A$="N"THEN700
690 GOTO260
700 INPUT"Datenaenderung...(J/N):";A$
710 IF A$="J"THEN CLS:GOTO210
720 INPUT"Datenerweiterung.(J/N):";Y$
730 IF Y$="J"THEN CLS:N=N+1:GOTO160
740 INPUT"Neueingabe.......(J/N):";A$
750 IF A$="N"THEN CLS:END
760 GOTO100
770 INPUT"Alles richtig J/N:";X$
780 IF X$="N"THEN810
790 IF X$="J"THEN CLS:RETURN
800 GOTO770
810 INPUT"Welche Nr. soll berichtigt werden";I
820 PRINT I;:INPUT I$(I):RETURN
```

Zeile 580 und 590 schalten den Drucker ein, das hier ein normaler Fernschreiber ist (T100s). Das Druckerprogramm ist das TTY-SYS vom Herrn Ploss und liegt hier auf der Adresse (D800 - DFFFh). Die Aktivierung liegt im Befehl DOKE 4100,-10201. Sollte das TTY - SYS in einem anderen Bereich liegen, so ist die zweite Ziffer in diesem DOKE Befehl dementsprechend zu aendern. In den Zeilen Nr., 620 und 630, das entspricht dem "N" Befehl, wird der Drucker wieder abgeschaltet.

Fußball-Tabelle

von Klaus Mombaur

Dieses BASIC - Programm errechnet aus den eingegebenen Ergebnissen den Tabellenstand einer beliebigen Liga nach Punkten, Tordifferenz und erzielten Toren. Es berücksichtigt ausgefallene Spiele entsprechend und liefert 2 verschiedene Tabellen sowie die Spielpaarungen der nächsten Wochen.
Die Namen der Vereine und die Paarungen werden einmal jährlich eingegeben. Die Ergebnisse werden direkt in dem Speicherbereich 0C80 bis 0DFF abgelegt. Daher speichert man das gesamte Programm am besten von NAS - SYS aus mit dem W - Befehl 0C80 40C0. Geladen wird dann mit dem R - Befehl, danach Z - Befehl und RUN ! Damit man sich möglichst nicht vertippt, wird das Ergebnis z.B. 2:4 wie folgt eingegeben: 2 ENTER 4 ENTER S ENTER
Wer die DATA - Inhalte so stehen läßt und das HEX - Listing am Schluß noch lädt, der hat den Tabellenstand der 1. Fußballbundesliga nach dem 20. Spieltag vom 30.1.1982.

Für Journalleser kann ich noch anbieten: Wer mir eine Cassette schickt (mit Rückporto), der bekommt das Gesamtprogramm auf dem allerletzten Spielstand im Format NASCOM 2 mit 1200 oder 300 Baud.
Wollen Sie die Tabelle für andere Sportarten, so sollten Sie noch wissen, daß bis zu 22 Vereine bei gerader Anzahl verarbeitet werden.

```
50 REM C by Kl. Mombaur Nuernberg
100 REM 1.Seite = Angebot
110 CLS
120 PRINT"                Fussballbundesliga
130 PRINT"                ~~~~~~~~~~~~~~~~~~
140 PRINT"Waehlen Sie aus folgendem Angebot:
145 PRINT
150 PRINT"Ergebnisse eingeben . . . . . .
 - E -
160 PRINT"Aktuelle Tabelle ausgeben . . . . .
 - T -
170 PRINT"Spiele der naechsten Wochen . . . . .
 - W -
180 PRINT"Programm beenden  . . . . . .
 - P -
185 PRINT
190 PRINT     "    Start einer neuen Saison:
200 PRINT"Vereinsnamen eingeben . . . . . .
 - V -
210 PRINT"Spielpaarungen eingeben . . . . . .
 - S -
215 PRINT:CLEAR
220 INPUT"Geben Sie eine der Kennziffern an:";K
$
230 DIMV1(12),V2(12),E1(12),E2(12),P1(12),P2(12
)
231 DIMV(23),P(23),TD(23),M(23),G(23),K(23)
233 DIMW(23),S(23),GS(23),VS(23),US(23)
250 IF K$="V"THEN1000
260 IF K$="S"THEN2000
270 IF K$="E"THEN3000
280 IF K$="T"THEN5000
290 IF K$="W"THEN7000
300 IF K$="P"THENCLS:END
340 GOTO100
1000 REM N
1005 REM Speicher loeschen
1006 FOR A=1TO350
1007 POKE3200+A,00
1008 NEXT
1010 CLS
1020 INPUT"Wieviel Vereine hat die neue 1.Liga?
";V1
1025 POKE3200,V1
1030 PRINT
1040 PRINT"Schreiben Sie nun die Namen
1050 PRINT"der Vereine der gewuenschten Liga:
1060 PRINT"(max. 15 Zeichen je Verein, 22 Verei
ne)
1070 PRINT
1080 PRINT"Ab Zeile 10000 zuerst DATA, dann NR,
1090 PRINT"dann KOMMA, dann VEREIN!
1100 LIST10000
2000 REM - S -
2010 CLS
2020 PRINT"Geben Sie nun die Spielpaarungen ein

2030 PRINT"Schreiben Sie nicht den Verein, sond
ern
2040 PRINT"die Nr., unter der Sie den Verein
2050 PRINT"ab Zeile 10000 abgelegt haben.
2060 PRINT"Fuer jeden Spieltag: 1:Spieldatum,
2070 PRINT"                    2:Spielpaarunge
n
2075 PRINT
2080 PRINT"Ab Zeile 10100 zuerst DATA, dann Dat
um
2090 PRINT"dann KOMMA, dann 3-7,4-12 usw.!
2095 PRINT"(siehe folgendes Vorjahresbeispiel)
2100 LINES1:LIST 10100
3000 REM E
3005 F=3320:G=3344:H=3368:I=3392
3010 CLS:RESTORE
3020 B=PEEK(3200):Z=PEEK(3417)
3030 FOR A=1TO B*2:READD$:NEXT:REM Datasprg.
3032 CLS:PRINT:PRINT"Normale Ergebnisse?      (
= E)
3033 PRINT:PRINT"Oder Nachholspiele?      (= N)
3034 PRINT:INPUT"Oder zurueck zum Anfang? (= A)
";J$
3036 IFJ$="N"THEN4300
3037 IFJ$="A"THEN100
3040 FOR A=1TO(Z+1)*(B/2+1)-B/2
3050 READD$
3060 NEXT
3065 POKE3417,Z+1
3070 PRINT:PRINT:PRINT"Geben Sie mir die Ergebn
isse
3080 PRINT"des Spieltages ";D$;":?
3085 UP=0
3090 FOR A=1TO B/2: REM evtl +1 wenn ungerade
3100 READ V$
3110 V1(A)=VAL(LEFT$(V$,2))
3120 V2(A)=VAL(MID$(V$,3)):V2(A)=ABS(V2(A))
3130 NEXT:ONUPGOTO7150
3140 REM
3150 FOR A=1 TO B/2:RESTORE
3170 FOR C=1TOV1(A)*2
3180 READV1$
3190 NEXT
3200 RESTORE
3210 FOR C=1TOV2(A)*2
3220 READV2$
3230 NEXT:ONUPGOTO7170
3232 IFJ$<>"N"THEN3240
3234 GOTO4100
3240 PRINT:PRINT:PRINTV1$;"  gegen  ";V2$
3250 INPUT"     ";E1(A)
3255 IFLEN(STR$(E1(A)))>3THEN3240
3260 PRINT"      zu
3270 INPUT"     ";E2(A)
3275 IFLEN(STR$(E2(A)))>3THEN3240
3290 J$="E":GOTO3960
3295 GOSUB3750
3300 REM
3310 IFE1(A)>E2(A)THEN P1(A)=2:P2(A)=0:GOSUB380
0
3320 IFE1(A)<E2(A)THEN P2(A)=2:P1(A)=0:GOSUB384
0
3330 IFE1(A)=E2(A)THEN P1(A)=1:P2(A)=1:GOSUB387
0
3335 IFJ2=1THENJ2=0:GOTO3350
3340 NEXT
3350 REM
3360 FORA=1TOB/2
```

```
3370 C=PEEK(3200+V1(A))
3380 C=C+P1(A)
3390 POKE3200+V1(A),C
3400 C=PEEK(3200+V2(A))
3410 C=C+P2(A)
3420 POKE3200+V2(A),C
3430 NEXT
3440 FORA=1TOB/2
3450 C=PEEK(3272+V1(A))
3460 C=C+E1(A)
3470 POKE3272+V1(A),C
3480 C=PEEK(3272+V2(A))
3490 C=C+E2(A)
3500 POKE3272+V2(A),C
3510 NEXT
3520 FORA=1TOB/2
3530 C=PEEK(3248+V1(A))
3540 C=C+E2(A)
3550 POKE3248+V1(A),C
3560 C=PEEK(3248+V2(A))
3570 C=C+E1(A)
3580 POKE3248+V2(A),C
3590 NEXT
3650 REM
3660 FORA=1TOB/2
3670 C=PEEK(3224+V1(A))
3680 C=C+P2(A)
3690 POKE3224+V1(A),C
3700 C=PEEK(3224+V2(A))
3710 C=C+P1(A)
3720 POKE3224+V2(A),C
3730 NEXT
3740 GOTO3900
3750 REM U-Pro
3760 D=PEEK(F+V1(A)):POKEF+V1(A),D+1
3770 D=PEEK(F+V2(A)):POKEF+V2(A),D+1
3780 RETURN
3800 REM U-Pro
3810 D=PEEK(G+V1(A)):POKEG+V1(A),D+1
3820 D=PEEK(I+V2(A)):POKEI+V2(A),D+1
3830 RETURN
3840 D=PEEK(G+V2(A)):POKEG+V2(A),D+1
3850 D=PEEK(I+V1(A)):POKEI+V1(A),D+1
3860 RETURN
3870 D=PEEK(H+V1(A)):POKEH+V1(A),D+1
3880 D=PEEK(H+V2(A)):POKEH+V2(A),D+1
3890 RETURN
3900 REM
3910 C=3418:D=3297
3920 FORA=1TOB
3930 POKED,PEEK(C):C=C+1:D=D+1
3940 NEXT
3950 GOTO5000
3960 PRINT:PRINT"Eingabe abspeichern? (= S)
3961 PRINT"Eingabe verbessern?  (= E)
3962 IFJ2=1THENINPUT"                ";J$:GOTO3966
3963 INPUT"Spiel ausgefallen?   (= A)";J$
3964 CLS
3965 IFJ$="A"THENE1(A)=0:E2(A)=0:GOTO4000
3966 IFJ$="S"THEN3295
3967 IFJ$="E"THEN3240
3968 GOTO3960
4000 K=3442:M=1
4020 D=PEEK(K)
4030 IFD=0THEN4050
4040 M=M+3:K=K+3:GOTO4020
4050 POKE3441+M,Z+1
4060 POKE3441+M+1,V1(A)
4070 POKE3441+M+2,V2(A)
4080 GOTO3340
4100 K=3442
4110 L1=PEEK(K+1):L2=PEEK(K+2)
4120 IFK=3550THEN3340
4130 IFL1=V1(A)THENIFL2=V2(A)THEN4150
4140 K=K+3:GOTO4110
4150 IFV3$=V1$THENIFV4$=V2$THEN4200
4160 GOTO3340
4200 POKEK,00:POKEK+1,00:POKEK+2,00
4220 GOTO3240
4300 CLS
4310 PRINT"Folgende Nachholspiele sind offen:
4320 B=PEEK(3200):A1=0
4330 FORNA=3442TO3550STEP3
4340 N1=PEEK(NA+1):N2=PEEK(NA+2)
4350 IFN1<>0THEN4370
4360 NEXT:GOTO4460
4370 RESTORE:A1=A1+1
4380 FORA=1TON1
4385 READV$,V3$
4390 NEXT
4400 RESTORE
4410 FORA=1TON2
4420 READV$,V4$
4430 NEXT
4440 PRINTA1;": ";V3$;" - ";V4$
4442 PRINT:INPUT"Haben Sie das Ergebnis ? (J/N)
";J$
4444 IFJ$="J"THEN4470
4450 NEXTNA
4460 PRINT:PRINT"Keine ausstehenden Spiele !"
4465 FORA=1TO4000:NEXT:GOTO100
4470 J2=1:J$="N"
4480 RESTORE
4490 FORA=1TOB*2:READD$:NEXT
4500 FORA=1TOPEEK(NA)*(B/2+1)-(B/2):READD$:NEXT
4520 GOTO3085
5000 REM T
5005 CLS
5006 PRINT:PRINT:PRINT:PRINT
5007 PRINTTAB(10)"Etwas Geduld . . . ":PRINT
5008 PRINTTAB(14)          "ich errechne jetzt
5009 PRINTTAB(14)"aus den abgelegten Daten
5010 PRINTTAB(14)     "           die Tordifferen
z,
5011 PRINTTAB(14)     "sortiere nach Punkten
5012 PRINTTAB(14)     "          nach Tordifferen
z
5014 PRINTTAB(14)     "      und nach erzielten T
oren!
5030 REM
5040 B=PEEK(3200)
5050 C=3201
5060 FORA=1TOB
5070 P(A)=PEEK(C):V(A)=A
5080 C=C+1
5090 NEXT
5100 REM
5110 FORK=1TOB-1
5120 FORA=1TOB-1
5130 C=P(A)
5140 D=P(A+1)
5150 IFC>DTHEN5180
5160 P(A)=D:E=V(A):F=V(A+1)
5170 P(A+1)=C:V(A+1)=E:V(A)=F
5180 NEXTA:NEXTK
5182 C=3224
5184 FORA=1TOB
5185 M(A)=PEEK(C+V(A)):NEXT
5190 REM
5192 C=3272
5194 FORA=1TOB
5196 D=PEEK(C+V(A)):E=PEEK(C-24+V(A))
5198 TD(A)=D-E:NEXT
5200 REM
5202 FORA=1TOB-1
5204 C=P(A):D=P(A+1)
5205 IFS(A)=ATHEN5208
5206 IFC=DTHEN5209
5208 NEXT:GOTO5250
5209 IFM(A)>M(A+1)THENS(A)=A:GOTO5214
5210 IFTD(A)<TD(A+1)THEN5214
5212 NEXT:GOTO5250
5214 E=V(A):F=V(A+1):V(A)=F:V(A+1)=E
5216 GOTO5182
5250 C=3224
5260 FORA=1TOB
5270 M(A)=PEEK(C+V(A))
5280 NEXT
5300 C=3272
5310 FORA=1TOB
5320 G(A)=PEEK(C+V(A))
5330 NEXT
5340 REM
5345 FORA=1TOB-1
5350 C=P(A):D=P(A+1)
5353 IFS(A)=ATHEN5360
5355 IFC=DTHEN5365
5360 NEXT:GOTO5400
5365 IFTD(A)=TD(A+1)THEN5375
5370 NEXT:GOTO5400
5375 IF G(A)<G(A+1)THEN5214
5380 NEXT
5400 C=3320
5410 FORA=1TOB
5420 S(A)=PEEK(C+V(A))
5430 NEXT
5440 C=3344
5450 FORA=1TOB
```

```
5460 GS(A)=PEEK(C+V(A))
5470 NEXT
5480 C=3392
5490 FORA=1TO8
5500 VS(A)=PEEK(C+V(A))
5510 NEXT
5520 C=3368
5530 FORA=1TO8
5540 US(A)=PEEK(C+V(A))
5550 NEXT
5560 C=3248
5570 FORA=1TO8
5580 K(A)=PEEK(C+V(A)):NEXT
5700 REM
5710 CLS
5720 PRINT"Wollen Sie die Tabelle mit:
5721 PRINT
5725 PRINT"Anzahl der Spiele,
5726 PRINT"Gewonnene, Unentschiedene, Verlorene
5727 PRINT"und der Punktzahl?        Geben Sie:
 -1-
5728 PRINT
5730 PRINT"Tabellenplatz der Vorwoche,
5731 PRINT"Anzahl der geschossenen Tore,
5732 PRINT"        der kassierten Tore,
5733 PRINT"        der Tordifferenz
5734 PRINT"   und der Punktzahl?     Geben Sie:
 -2-
5736 PRINT
5738 PRINT"Wollen Sie zurueck zum Anfang?
5739 PRINT"                          Geben Sie:
 -3-
5740 INPUT U
5742 IF U<1THEN5700
5745 IF U>3THEN5700
5748 IFU=3THEN100
5750 REM
5755 CLS:RESTORE:C=3296
5758 ON U GOSUB 5900,5950
5760 FORA=1TO8/2:REM 1.Haelfte
5762 ON U GOSUB 5830,5775
5764 NEXT
5765 INPUT"       naechste Haelfte?";J$
5766 CLS:ON U GOSUB5900,5950
5767 FORA=8/2+1TO8:REM 2.Haelfte
5768 ON U GOSUB 5830,5775
5769 NEXT
5770 INPUT"      nochmal die 1.Haelfte? (J/N)";
 J$
5771 IFJ$="J"THENCLS:GOTO5755
5772 GOTO5850
5774 REM U-Pro
5775 READX,V$
5778 W(A)=PEEK(C+V(A))
5780 IFX=V(A)THEN5810
5800 GOTO5775
5810 PRINTA;W(A)TAB(8)V$TAB(23)G(A)TAB(28)":";
5811 PRINTK(A)TAB(34)TD(A)TAB(38)P(A)TAB(42)":"
M(A)
5820 RESTORE:RETURN
5830 READX,V$
5834 IFX=V(A)THEN5840
5836 GOTO5830
5840 PRINTATAB(4)V$TAB(20)S(A)TAB(25)GS(A)TAB(2
9);
5841 PRINTUS(A)TAB(33)VS(A)TAB(38)P(A)TAB(42)":
;M(A)
5845 RESTORE:RETURN
5850 REM
5860 C=3417
5870 FORA=1TO8
5880 POKEC+V(A),A
5890 NEXT
5895 GOTO5700
5900 PRINT"PL  V e r e i n   Spiele  G   U   V"
;
5905 PRINTTAB(39)"Punkte
5910 FORN=1TO47:PRINTCHR$(61);:NEXT
5920 RETURN
5950 PRINT"PL  VW  V e r e i n     T o r e    T
D";
5955 PRINTTAB(39)"Punkte
5960 FORN=1TO47:PRINTCHR$(61);:NEXT
5970 RETURN
7000 REM  W
7020 B=PEEK(3200):Z=PEEK(3417)
7030 CLS:RESTORE
7040 FORA=1TO8*2:READD$:NEXT
7050 FORA=1TO(Z+1)*(B/2+1)-B/2:READD$:NEXT
7060 PRINT"Die Spielpaarungen der naechsten Woc
he: ";
7070 PRINTD$
7075 IFD$="AUS"THENPRINT"Saisonende!":GOTO 180
7080 FORA=1TO47:PRINTCHR$(61);:NEXT
7090 UP=1:GOTO3090
7150 FORA=1TO8/2:RESTORE
7160 GOTO3170
7170 PRINTV1$;TAB(17)"gegen   ";V2$
7180 NEXT
7190 PRINT
7200 PRINT"Noch eine Woche weiter?  (= W)
7210 INPUT"Oder zurueck zum Anfang? (= A)";J$
7220 IFJ$()"W"THEN100
7230 Z=Z+1:J$="A":GOTO7030
10000 DATA 1,Bayern Muenchen
10001 DATA 2,Arm. Bielefeld
10002 DATA 3,VfL Bochum
10003 DATA 4,Werder Bremen
10004 DATA 5,E. Braunschweig
10005 DATA 6,Darmstadt 98
10006 DATA 7,Bor. Dortmund
10007 DATA 8,Fortuna D'dorf
10008 DATA 9,MSV Duisburg
10009 DATA10,Bor.M'gladbach
10010 DATA11,Eintr.Frankfurt
10011 DATA12,Hamburger SV
10012 DATA13,1.FC K'lautern
10013 DATA14,Karlsruher SC
10014 DATA15,1.FC Koeln
10015 DATA16,Bay.Leverkusen
10016 DATA17,1.FC Nuernberg
10017 DATA18,VfB Stuttgart
10020 REM DATA19,Darmstadt 98
10021 REM DATA20,Offenbacher Kl.
10022 REM DATA21,Stuttgarter Kl.
10023 REM DATA22,Hessen Kassel
10024 REM DATA23,SSV Ulm
10025 REM DATA24,SC Freiburg
10100 DATA 8.8.,1-16,2-6,10-4,18-8,9-14,12-5
10110 DATA 11-13,15-7,3-17
10120 DATA 15.8.,17-15,13-12,5-9,14-18,8-10
10130 DATA 4-2,6-1,16-3,7-11
10140 DATA 22.8.,1-4,2-8,10-14,18-5,9-13
10150 DATA 12-7,11-17,16-6,3-15
10160 DATA 25.8.,17-12,13-18,5-10,14-2,8-1
10170 DATA 4-16,6-3,15-11,7-9
10180 DATA 5.9.,2-5,10-13,18-7,9-17,12-15,6-4
10190 DATA 16-8,3-11,1-14
10200 DATA 12.9.,17-18,13-2,5-1,14-16,8-6
10210 DATA 4-3,11-12,15-9,7-10
10220 DATA 19.9.,1-13,2-7,10-17,18-15,9-11
10230 DATA 4-8,6-14,16-5,3-12
10240 DATA 26.9.,17-2,13-16,5-6,14-4,12-9
10250 DATA  11-18,15-10,7-1,8-3
10260 DATA 3.10.,1-17,2-15,10-11,18-12,8-14
10270 DATA 4-5,6-13,16-7,3-9
10280 DATA 17.10.,17-16,13-4,5-8,14-3,9-18
10290 DATA 12-10,11-2,15-1,7-6
10300 DATA 24.10.,1-11,2-12,10-9,14-5,8-13
10310 DATA 4-7,6-17,16-15,3-18
10320 DATA 31.10.,17-4,13-14,5-3,18-10,9-2
10330 DATA 12-1,11-16,15-6,7-8
10340 DATA 7.11.,1-9,2-18,5-13,14-7,8-17
10350 DATA4-15,6-11,16-12,3-10
10360 DATA 14.11.,17-14,13-3,10-2,18-1,9-16
10370 DATA 12-6,11-4,15-8,7-5
10380 DATA 28.11.,1-10,2-3,5-17,14-15,8-11
10390 DATA 4-12,6-9,16-18,7-13
10400 DATA 12.12.,17-13,2-1,10-16,18-6,9-4
10410 DATA 12-8,11-14,15-5,3-7
10420 DATA 19.12.,1-3,13-15,5-11,14-12,8-9
10430 DATA 4-18,6-10,16-2,7-17
10440 DATA16.1.,16-1,6-2,4-10,8-18,14-9,5-12
10450 DATA 13-11,7-15,17-3
10460 DATA 23.1.,15-17,12-13,9-5,18-14,10-8
10470 DATA 2-4,1-6,3-16,11-7
10480 DATA 30.1.,4-1,8-2,14-10,5-18,13-9
10490 DATA 7-12,17-11,6-16,15-3
10500 DATA 6.2.,12-17,18-13,10-5,2-14,1-8
10510 DATA 16-4,3-6,11-15,9-7
10520 DATA 13.2.,5-2,13-10,7-18,17-9,15-12,4-6
10530 DATA 8-16,11-3,14-1
10540 DATA 27.2.,18-17,2-13,1-5,16-14,6-8
10550 DATA 3-4,12-11,9-15,10-7
10560 DATA 6.3.,13-1,7-2,17-10,15-18,11-9
10570 DATA 8-4,14-6,5-16,12-3
10580 DATA 13.3.,2-17,16-13,6-5,4-14,9-12
10590 DATA 18-11,10-15,1-7,3-8
```

```
10600 DATA 27.3.,17-1,15-2,11-10,12-18,14-8
10610 DATA 5-4,13-6,7-16,9-3
10620 DATA 3.4.,16-17,4-13,8-5,3-14,18-9
10630 DATA 10-12,2-11,1-15,6-7
10640 DATA 17.4.,11-1,12-2,9-10,5-14,13-8
10650 DATA 7-4,17-6,15-16,18-3
10660 DATA 24.4.,4-17,14-13,3-5,10-18,2-9
10670 DATA 1-12,16-11,6-15,8-7
10680 DATA 28.4.,9-1,18-2,13-5,7-14,17-8
10690 DATA 15-4,11-6,12-16,10-3
10700 DATA 8.5.,14-17,3-13,2-10,1-18,16-9
10710 DATA 6-12,4-11,8-15,5-7
10720 DATA 15.5.,10-1,3-2,17-5,15-14,11-8
10730 DATA 12-4,9-6,18-16,13-7
10740 DATA 22.5.,13-17,1-2,16-10,6-18,4-9
10750 DATA 8-12,14-11,5-15,7-3
10760 DATA 29.5.,3-1,15-13,11-5,12-14,9-8
10770 DATA 18-4,18-6,2-16,17-7
11000 DATA "AUS"
```

```
0C80 12 1C 0D 10 17 14 0E 16 |26
0C88 0F 0A 1C 14 16 11 0C 1C |2C
0C90 0D 0F 12 00 00 00 00 00 |DA
0C98 00 0A 1B 16 0F 10 18 12 |28
0CA0 17 1A 0C 10 0C 11 16 0A |36
0CA8 17 17 12 00 00 00 00 00 |F4
0CB0 00 1B 20 1E 1D 1C 2A 19 |91
0CB8 27 2D 1A 24 17 21 23 0E |BF
0CC0 26 28 1C 00 00 00 00 00 |36
0CC8 00 2C 13 1C 1F 20 16 23 |A7
0CD0 1F 17 26 2E 2E 24 19 27 |F8
0CD8 16 1C 19 00 00 00 00 00 |2F
0CE0 00 01 10 0A 04 08 0D 05 |25
0CE8 0E 12 02 07 06 0B 11 03 |42
0CF0 0F 0C 09 00 00 00 00 00 |20
0CF8 00 13 14 13 13 12 13 14 |8A
0D00 13 12 14 12 11 11 11 13 |9E
0D08 12 13 12 00 00 00 00 00 |4C
0D10 00 0D 04 05 09 0A 04 09 |53
0D18 05 04 0A 09 09 05 04 0C |5F
0D20 04 06 06 00 00 00 00 00 |3D
0D28 00 02 05 06 05 00 06 04 |51
0D30 05 02 08 02 04 07 04 04 |61
0D38 05 03 06 00 00 00 00 00 |53
0D40 00 04 0B 08 05 08 09 07 |81
0D48 09 0C 02 07 04 05 09 03 |88
0D50 09 0A 06 00 00 00 00 00 |76
0D58 00 14 02 10 0B 04 08 0E |B0
0D60 06 0C 12 03 07 05 0A 11 |BB
0D68 01 0F 0D 09 00 00 00 00 |9B
0D70 00 00 12 0E 09 12 0D 0B |D0
0D78 10 0C 08 11 0D 0F 11 0E |F5
0D80 0C 13 0C 0D 13 09 05 13 |F9
0D88 12 0E 13 03 10 14 04 01 |F4
0D90 14 05 12 14 11 0B 14 06 |12
0D98 10 00 00 00 00 00 00 00 |B5
```

Eprommer Ergänzung

von Otto Fößel

Hier ein Ergänzungsvorschlag zum EPROMMER, den Rüdiger Maurer in Heft 11/12-81 vorstellte.

All diejenigen, die selbst nicht schon den 2708-EPROMMER besitzen, müssen mit einem zusätzlichen Trafo, Gleichrichter und Stabilisierungsschaltung die 25V Programmierspannung erzeugen. Dazu kommt noch, daß die 25V-Leitung im PIO-Bus eine Leitung für eventuell andere Anwendung blockiert.

Diese beiden Nachteile macht ein einziges IC wett, das jederzeit noch auf der PROMMER Platine Platz findet.

Mit diesem IC und ein paar Beschaltungsbauteilen wird die vorhandene 5V-Spannung auf 25V hochtransformiert. Die Beschaltung dazu zeigt Bild 1.

Bild 2 ist ein Vorschlag, wie das Layout der Ätzfolie erweitert werden könnte.

Bild 1: Beschaltung des TL 497

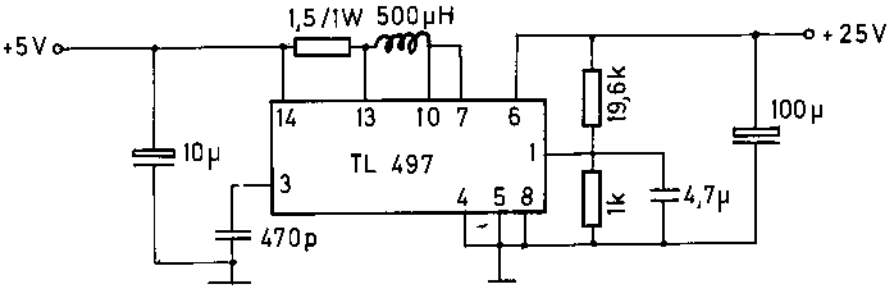
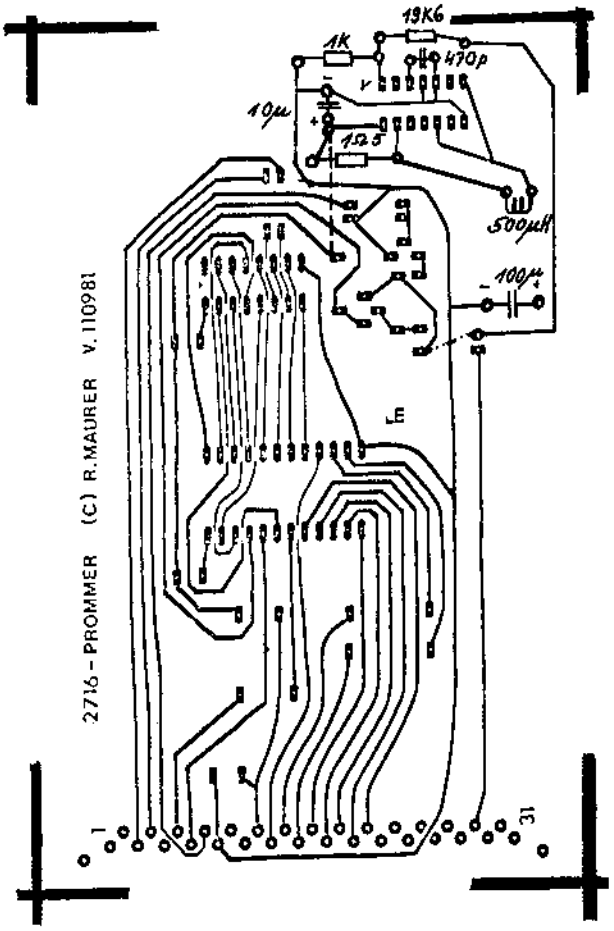


Bild 2: Erweitertes Platinen-Layout (Lötseite)



Der Preis für das IC (TL 497 von Texas) beträgt ca. 5.-DM (z.B. bei Fa Frank/ Nürnberg). Dies ist also eine echte Alternative zur herkömmlichen Methode mit Trafo.

Literatur: Manual ECB 85

Menue-Programm

von Rüdiger Maurer

```
                0010 ;   !! MENUEPROGRAMM Vers.3 - 19.01.82 !!
                0020 ;   Geaenderte Portadressen fuer
                0030 ;   den Drucker !!!!!!!!!!!!!!!!

                0050 ;   (c) Ruediger Maurer, [redacted]
                0060 ;       [redacted] Taunusstein [redacted]

                0080 ;  Folgende Bedienungsprogramme muessen
                0090 ;  auf den angegeben Adressen stehen :

                0110 ;  BASIC        = EE000 - EFFFF
                0120 ;  ZEAP 2.0     = ED000 - EDFFF
                0130 ;  DEBUGGER     = EC000 - EC3FF
                0140 ;  DISASSEMBLER = EC400 - ECFFF
                0150 ;  NASPEN       = EB800 - EBFFF
                0160 ;  TOOLKIT      = EB000 - EB7FF
                0170 ;  2708-PROMMER = E9800
                0180 ;  2716-PROMMER = E9C00
                0190 ;  MENUE        = E9000
                0195 ;  MDCR-MONIOR  = EA000

3E8F 9C00       0210 PRO6   EQU  E9C00
3E8F 9800       0220 PRO8   EQU  E9800

                0240 ;  Wenn die Bedienprogramme an anderer
                0250 ;  Stelle in Menueprogramm stehen, sind
                0260 ;  die Einsprungsadressen entsprechend zu
                0270 ;  aendern (Siehe Kommentare)!
                0280 ;  Um die NASPEN-AUSGABE ueber den Drucker
                0290 ;  zu ermoeglichen muessen im NASPEN-EPROM
                0300 ;  folgende Speicherplaetze geaendert
                0310 ;  werden :
                0320 ;           EB859 von EDF in EC3
                0330 ;           EB85A von E6F in EB8  *
                0340 ;           EB85A von EC9 in EA5  *
                0350 ;  *  Achtung ! EA5B8 = Label AUSG
                0360 ;  Bei Programmaenderung beachten !

                0380 ;  STARTADRESSE=E9000
                0390 ;  Wenn Assemblierung auf E7000 gewuenscht
                0400 ;  dann mit "P E000" Offset zwischen
                0410 ;  Assemblierungsadresse und Ablageadresse
                0420 ;  vorgeben !

9000            0440 START  ORG  E9000 ; ASSEMBLIERT AUF E9000
9000 21DF6F     0450        LD   HL,E6FDF    ; INITIALISIERE ASSEMBLER
9003 22800C     0460        LD   (E0C80),HL  ; FUER SERIELL-OUT
9006 21C900     0470        LD   HL,E00C9    ; OHNE DRUCKERAUSGABE
9009 22820C     0480        LD   (E0C82),HL  ;
900C EF         0490        RST  E28
900D 0C         0500        DEFB E0C ;      CLEAR SCREEN
```



```
900E 204D4443 0510          DEFM ! MDCR-Monitor ..... M!
     522D4D6F
     6E69746F
     72202E2E
     2E2E2E2E
     2E2E2E2E
     2E2E2E20
     4D
902B 0D       0520          DEFB E0D
902C 20546F6F 0530          DEFM ! Tool-Kit ..... T!
     6C2D4B69
     74202E2E
     2E2E2E2E
     2E2E2E2E
     2E2E2E2E
     2E2E2E20
     54
9049 0D       0540          DEFB E0D
904A 20426173 0550          DEFM ! Basic Cold / Warm ..... BC / B!
     69632043
     6F6C6420
     2F205761
     726D202E
     2E2E2E2E
     2E2E2E20
     4243202F
     2042
906C 0D       0560          DEFB E0D
906D 204E6173 0570          DEFM ! Naspen Cold / Warm ..... PC / P!
     70656E20
     436F6C64
     202F2057
     61726D20
     2E2E2E2E
     2E2E2E20
     5043202F
     2050
908F 0D       0580          DEFB E0D
9090 20417373 0590          DEFM ! Assembler Cold / Warm .... AC / A!
     656D626C
     65722043
     6F6C6420
     2F205761
     726D202E
     2E2E2E20
     4143202F
     2041
90B2 0D       0600          DEFB E0D
90B3 20446562 0610          DEFM ! Debugger / Disassembler .. DB / D!
     75676765
     72202F20
     44697361
     73736D6D
     626C6572
     202E2E20
     4442202F
```

```
     2044
90D5 0D       0620        DEFB £0D
90D6 20323730 0630        DEFM ! 2708 / 2716 Programmer ... 8 or 6!
     38202F20
     32373136
     2050726F
     67726160
     6D657220
     2E2E2E20
     38206F72
     2036
90F8 0D       0640        DEFB £0D
90F9 20526574 0650        DEFM / Return to NAS-SYS ........ N/
     75726E20
     746F204E
     41532D53
     5953202E
     2E2E2E2E
     2E2E2E20
     4E
9116 0D       0660        DEFB £0D
9117 2052414D 0670        DEFM / RAM Test ................. R/
     20546573
     74202E2E
     2E2E2E2E
     2E2E2E2E
     2E2E2E2E
     2E2E2E20
     52
9134 0D       0680        DEFB £0D
9135 20436C65 0690        DEFM / Clear or Compare Memory .. C/
     6172206F
     7220436F
     6D706172
     65204D65
     6D6F7279
     202E2E20
     43
9152 0D       0700        DEFB £0D
9153 20507269 0710        DEFM / Printer Option ........... X/
     6E746572
     204F7074
     696F6E20
     2E2E2E2E
     2E2E2E2E
     2E2E2E20
     58
9170 0D       0720        DEFB £0D
9171 00       0730        DEFB £00 ; END OF MESSAGE

9172 DF       0750 ANFNG  RST  £18
9173 63       0760        DEFB £63 ; INPUT LINE
9174 1A       0770        LD   A,(DE)
9175 47       0780        LD   B,A   ; 1. BUCHSTABE IN A
9176 13       0790        INC  DE
9177 1A       0800        LD   A,(DE)

9178 4F       0810        LD   C,A   ; 2. BUCHSTABE IN C
9179 78       0820        LD   A,B

917A FE54     0840        CP   "T
917C 2017     0850        JR   NZ,WEIT1
917E 3EFF     0860        LD   A,£FF     ; DIE PORTS WERDEN AUF EIN-
9180 D312     0870        OUT  (£12),A ; GABE GESETZT UM DEN DRUCKER
9182 D312     0880        OUT  (£12),A ; ABZUSCHALTEN, DA TOOLKIT
9184 D313     0890        OUT  (£13),A ; DIE USR-OUT ADRESSE AENDERT
9186 D313     0900        OUT  (£13),A
9188 212F00   0910        LD   HL,£002F    ; DRUCKEROPTION WIRD
918B 22780C   0920        LD   (£0C78),HL ; AUSGESCHALTET F. TOOLKIT
918E 210080   0930        LD   HL,£8000    ;!IN £0C0C MUSS EINPRUNG-
9191 220C0C   0940        LD   (£0C0C),HL  ;ADRESSE STEHEN DA TOOLKIT
9194 E9       0950        JP   (£HL);T.KIT ;RELATIV VERSCHIEBBAR IST

9195 FE50     0970 WEIT1  CP   "P
9197 2009     0980        JR   NZ,PROG
9199 79       0990        LD   A,C
919A FE43     1000        CP   "C
919C CA00B8   1010        JP   Z,£B800   ; NASPEN COLD
919F C306B8   1020        JP   £B806     ; NASPEN WARM

91A2 FE38     1040 PROG   CP   "8      ;!IN £0C0C MUSS EINSPRUNG-
91A4 2007     1050        JR   NZ,WEIT2    ;ADRESSE STEHEN DA PROGR.
91A6 210098   1060        LD   HL,£PRO8    ;VOLL VERSCHIEBBAR IST UND
91A9 220C0C   1070        LD   (£0C0C),HL  ;SPRUNGADRESSEN INDIZIERT
91AC E9       1080        JP   (HL);2708   ;UEBER £0C0C BERECHNET !
91AD FE36     1090 WEIT2  CP   "6
91AF 2007     1100        JR   NZ,WEIT3
91B1 21009C   1110        LD   HL,£PRO6   ; ADRESSE FUER 2716-PROMMER
91B4 220C0C   1120        LD   (£0C0C),HL
91B7 E9       1130        JP   (HL)     ; 2716 PROGR.
91B8 FE41     1140 WEIT3  CP   "A
91BA 200B     1150        JR   NZ,NASYS
91BC DF       1160        RST  £18  ; Setzt NASYS in den N-Mode
91BD 4E       1170        DEFB "N  ; damit Assemblerfunktionen OK
91BE 79       1180        LD   A,C
91BF FE43     1190        CP   "C
91C1 CA00D0   1200        JP   Z,£D000 ; ASSEMBLER COLD
91C4 C303D0   1210        JP   £D003   ; ASSEMBLER WARM
91C7 FE4E     1220 NASYS  CP   "N
91C9 2002     1230        JR   NZ,BASIC
91CB DF       1240        RST  £18
91CC 5B       1250        DEFB £5B   ; NASYS
91CD FE42     1260 BASIC  CP   "B
91CF 2009     1270        JR   NZ,DISAS
91D1 79       1280        LD   A,C
91D2 FE43     1290        CP   "C
91D4 CAFAFF   1300        JP   Z,£FFFA ; BASIC COLD
91D7 C3FDFF   1310        JP   £FFFD   ; BASIC WARM
91DA FE44     1320 DISAS  CP   "D
91DC 2068     1330        JR   NZ,JUMP
91DE 79       1340        LD   A,C
91DF FE42     1350        CP   "B
91E1 CA00C0   1360        JP   Z,£C000 ; DEBUGGER
```

```
91E4 EF       1370        RST  £28
91E5 0D       1380        DEFB £0D    , NEW LINE
91E6 20446973 1390        DEFM / Disassembler Option Mode .. O/
     61737365
     6D626C65
     72204F70
     74696F6E
     204D6F64
     65202E2E
     204F
9204 0D       1400        DEFB £0D
9205 20202020 1410        DEFM /      "       Direct Mode .. D/
     20202220
     20202020
     20204469
     72656374
     204D6F64
     65202E2E
     2044
9223 0D       1420        DEFB £0D
9224 20526574 1430        DEFM / Return to Menue ........... M/
     75726E20
     746F204D
     656E7565
     202E2E2E
     2E2E2E2E
     2E2E2E2E
     204D
9242 0D00     1440        DEFB £0D,£00
9244 1802     1450        JR   CONT
9246 1858     1460 JUMP   JR   CLEAR
9248 DF       1470 CONT   RST  £18
9249 63       1480        DEFB £63
924A 1A       1490        LD   A,(DE)
924B FE4F     1500        CP   "O
924D CA00C4   1510        JP   Z,£C400   ; OPTION MODE
9250 FE44     1520        CP   "D
9252 2803     1530        JR   Z,DISDI
9254 C30090   1540        JP   START     ; RETURN TO MENUE
9257 EF       1550 DISDI  RST  £28
9258 0D       1560        DEFB £0D
9259 20537461 1570        DEFM / Start Adress and Number of Lines/
     72742041
     64726573
     7320616E
     64204E75
     6D626572
     206F6620
     4C696E65
     73
927A 20284865 1580        DEFM / (Hex!)/
     782129
9281 0D00     1590        DEFB £0D,£00
9283 DF       1600 DIS2   RST  £18  ; INPUT LINE
9284 63       1610        DEFB £63
9285 DF       1620        RST  £18
```

```
9286 79       1630        DEFB £79
9287 38FA     1640        JR   C,DIS2
9289 2A0E0C   1650        LD   HL,(£0C0E)  ; SCHIEBE ARG2 NACH ARG3
928C 22100C   1660        LD   (£0C10),HL
928F 2A0C0C   1670        LD   HL,(£0C0C)  ; SCHIEBE ARG1 NACH ARG2
9292 220E0C   1680        LD   (£0C0E),HL
9295 2A0B0C   1690        LD   HL,(£0C0B)  ; ADDIERE 2 ZU ARG-NUMBER
9298 23       1700        INC  HL
9299 23       1710        INC  HL
929A 220B0C   1720        LD   (£0C0B),HL
929D C300C4   1730        JP   £C400   ; DISASSEMBLER DIRECT MODE

92A0 FE43     1750 CLEAR  CP   "C
92A2 C29F94   1760        JP   NZ,GOON   ; WEITERE UEBERPRUEFUNGEN
92A5 EF       1770        RST  £28
92A6 0C0D     1780        DEFB £0C,£0D
92A8 20436C65 1790        DEFM / Clear All ................ A/
     61722041
     6C6C202E
     2E2E2E2E
     2E2E2E2E
     2E2E2E2E
     2E2E2041
92C4 0D       1800        DEFB £0D
92C5 20436C65 1810        DEFM / Clear Range .............. R/
     61722052
     616E6765
     202E2E2E
     2E2E2E2E
     2E2E2E2E
     2E2E2052
92E1 00       1820        DEFB £00
92E2 0D       1830        DEFB £0D
92E3 EF       1840        RST  £28
92E4 0D       1850        DEFB £0D
92E5 20436F6D 1860        DEFM / Compare 1k Block ........ C/
     70617265
     20316B20
     426C6F63
     6B202E2E
     2E2E2E2E
     2E2E2043
9301 0D       1870        DEFB £0D
9302 20526574 1880        DEFM / Return to Menue ......... M/
     75726E20
     746F204D
     656E7565
     202E2E2E
     2E2E2E2E
     2E2E204D
931E 0D00     1890        DEFB £0D,£00
9320 DF       1900        RST  £18
9321 63       1910        DEFB £63
9322 1A       1920        LD   A,(DE)
9323 FE41     1930        CP   "A
9325 280C     1940        JR   Z,ALL
```

```
9327 FE52     1950        CP   "R
9329 281C     1960        JR   Z,RANGE
932B FE43     1970        CP   "C
932D CAAC93   1980        JP   Z,COMP
9330 C30090   1990        JP   START     ; RETURN TO MENUE!
9333 3EFF     2000 ALL    LD   A,£FF
9335 21800C   2010        LD   HL,£0C80  ; STARTADRESS OF CLEAR
9338 06A0     2020        LD   B,£A0     ; ANZAHL 1/4 K-BLOEKE
933A C5       2030 LOOP1  PUSH BC
933B 0600     2040        LD   B,£00     ; 256 BYTES
933D 77       2050 LOOP2  LD   (HL),A
933E 23       2060        INC  HL
933F 10FC     2070        DJNZ LOOP2
9341 C1       2080        POP  BC
9342 10F6     2090        DJNZ LOOP1
9344 C30090   2100        JP   START
9347 EF       2110 RANGE  RST  £28
9348 0D       2120        DEFB £0D
9349 20456E74 2130        DEFM / Enter Start Adress and/
     65722053
     74617274
     20416472
     65737320
     616E64
9360 204C656E 2140        DEFM / Lenght of Memory (Hex!)/
     67687420
     6F66204D
     656D6F72
     79202848
     65782129
9378 0D       2150        DEFB £0D
9379 20314B20 2160        DEFM / 1K = 03FF  4K = 0FFF/
     3D203033
     46462020
     344B203D
     20304646
     46
938E 0D00     2170        DEFB £0D,£00
9390 DF       2180 INP    RST  £18
9391 63       2190        DEFB £63
9392 DF       2200        RST  £18
9393 79       2210        DEFB £79  ;ARG1=START ARG2=LENGHT
9394 38FA     2220        JR   C,INP
9396 3A0F0C   2230        LD   A,(£0C0F) ; LENGHT IN BC
9399 47       2240        LD   B,A
939A 3A0E0C   2250        LD   A,(£0C0E)
939D 4F       2260        LD   C,A
939E 3EFF     2270        LD   A,£FF      ; CLEAR A
93A0 2A0C0C   2280        LD   HL,(£0C0C)
93A3 77       2290        LD   (HL),A
93A4 54       2300        LD   D,H
93A5 5D       2310        LD   E,L
93A6 13       2320        INC  DE
93A7 EDB0     2330        LDIR
93A9 C30090   2340        JP   START      ; RETURN TO MENUE!
93AC EF       2350 COMP   RST  £28
93AD 0C0D     2360        DEFB £0C,£0D
93AF 31682053 2370        DEFM /1k Speicherversleich/
     70656963
     68657276
     6572676C
     65696368
93C3 0D       2380        DEFB £0D
93C4 45696E67 2390        DEFM /Eingabe Anfangsadressen:/
     61626520
     416E6661
     6E677361
     64726573
     73656E3A
93DC 0D00     2400        DEFB £0D,£00
93DE DF       2410 BACK   RST  £18
93DF 63       2420        DEFB £63
93E0 DF       2430        RST  £18
93E1 79       2440        DEFB £79
93E2 38FA     2450        JR   C,BACK
93E4 F5       2460        PUSH AF
93E5 C5       2470        PUSH BC
93E6 D5       2480        PUSH DE
93E7 E5       2490        PUSH HL
93E8 010300   2500        LD   BC,£0003
93EB C5       2510        PUSH BC
93EC ED580C0C 2520        LD   DE,(£0C0C)
93F0 2A0E0C   2530        LD   HL,(£0C0E)
93F3 1800     2540        JR   SPRG1
93F5 1A       2550 SPRG1  LD   A,(DE)
93F6 46       2560        LD   B,(HL)
93F7 B8       2570        CP   B
93F8 282A     2580        JR   Z,SPRG2
93FA E5       2590        PUSH HL
93FB D5       2600        PUSH DE
93FC E1       2610        POP  HL
93FD F5       2620        PUSH AF
93FE CD3E94   2630        CALL ROUT1
9401 DF       2640        RST  £18
9402 69       2650        DEFB £69 ;i
9403 00       2660        NOP
9404 F1       2670        POP  AF
9405 F5       2680        PUSH AF
9406 CD3B94   2690        CALL ROUT2
9409 CD4994   2700        CALL ROUT3
940C F1       2710        POP  AF
940D 90       2720        SUB  B
940E CD3894   2730        CALL ROUT2
9411 CD4994   2740        CALL ROUT3
9414 78       2750        LD   A,B
9415 CD3B94   2760        CALL ROUT2
9418 DF       2770        RST  £18
9419 69       2780        DEFB £69 ;i
941A 00       2790        NOP
941B E1       2800        POP  HL
941C CD3E94   2810        CALL ROUT1
941F CD4F94   2820        CALL ROUT4
```

```
9422 1800     2830        JR   SPRG2
9424 C1       2840 SPRG2  POP  BC
9425 100F     2850        DJNZ SPRG5
9427 AF       2860        XOR  A
9428 B9       2870        CP   C
9429 2805     2880        JR   Z,SPRG4
942B 0D       2890        DEC  C
942C 0600     2900        LD   B,£00
942E 1806     2910        JR   SPRG5
9430 E1       2920 SPRG4  POP  HL
9431 D1       2930        POP  DE
9432 C1       2940        POP  BC
9433 F1       2950        POP  AF
9434 1831     2960        JR   ENDE ; RUECKSPRUNG
9436 C5       2970 SPRG5  PUSH BC
9437 23       2980        INC  HL
9438 13       2990        INC  DE
9439 18BA     3000        JR   SPRG1
943B DF       3010 ROUT2  RST  £18
943C 68       3020        DEFB £68 ;h
943D C9       3030        RET
943E F5       3040 ROUT1  PUSH AF
943F E5       3050        PUSH HL
9440 7C       3060        LD   A,H
9441 DF       3070        RST  £18
9442 68       3080        DEFB £68 ;h
9443 7D       3090        LD   A,L
9444 DF       3100        RST  £18
9445 68       3110        DEFB £68 ;h
9446 E1       3120        POP  HL
9447 F1       3130        POP  AF
9448 C9       3140        RET
9449 DF       3150 ROUT3  RST  £18
944A 69       3160        DEFB £69 ;i
9448 00       3170        NOP
944C DF       3180        RST  £18
944D 69       3190        DEFB £69 ;i
944E C9       3200        RET
944F 00       3210 ROUT4  NOP
9450 E5       3220        PUSH HL
9451 D5       3230        PUSH DE
9452 DF       3240        RST  £18
9453 6A       3250        DEFB £6A ;j
9454 DF       3260        RST  £18
9455 61       3270        DEFB £61 ;a
9456 3008     3280        JR   NC,SPRG6
9458 DF       3290 SPRG7  RST  £18
9459 61       3300        DEFB £61 ;a
945A 00       3310        NOP
945B 30F8     3320        JR   NC,SPRG7
945D FE0D     3330        CP   £0D ; VERGLEICH OB NEW LINE (WEITER)
945F 2802     3340        JR   Z,SPRG6
9461 18F5     3350        JR   SPRG7
9463 D1       3360 SPRG6  POP  DE
9464 E1       3370        POP  HL
9465 00       3380        NOP
```

```
9466 C9       3390        RET
9467 EF       3400 ENDE   RST  £28
9468 536F6E73 3410        DEFM /Sonst keine Fehler/
     74206B65
     696E6520
     4665686C
     6572
947A 0D       3420        DEFB £0D
947B 5A757275 3430        DEFM !Zurueck ins MENUE ? (J)!
     65636B20
     696E7320
     4D454E55
     45203F20
     284A29
9492 0D00     3440        DEFB £0D,£00
9494 DF       3450        RST  £18
9495 63       3460        DEFB £63
9496 1A       3470        LD   A,(DE)
9497 FE4A     3480        CP   "J
9499 CA0090   3490        JP   Z,START
949C C3AC93   3500        JP   COMP
949F FE52     3510 GOON   CP   "R  ; RAM - TEST ?
94A1 2803     3520        JR   Z,RAMTE
94A3 C3A395   3530        JP   DRUCK  ; WEITERE UEBERPRUEFUNGEN

94A6 EF       3550 RAMTE  RST  £28 ; RAMTEST AUS 32K RAM MANUAL
94A7 0C0D     3560        DEFB £0C,£0D
94A9 53746172 3570        DEFM /Start- and Endadress of Memorytest/
     74204164
     6E642045
     6E646472
     65737373
     206F6620
     4D656D6F
     72797465
     7374
94CB 0D00     3580        DEFB £0D,£00
94CD DF       3590 RAMT1  RST  £18
94CE 63       3600        DEFB £63
94CF DF       3610        RST  £18
94D0 79       3620        DEFB £79
94D1 38FA     3630        JR   C,RAMT1
94D3 EF       3640        RST  £28
94D4 0C00     3650        DEFB £0C,£00 ;..
94D6 EF       3660        RST  £28
94D7 4D656D6F 3670        DEFM /Memory Test Part 1/
     72792054
     65737420
     50617274
     2031
94E9 0D00     3680        DEFB £0D,£00 ;..
94EB 2A0E0C   3690        LD   HL,(£0C0E); LADE ARG2
94EE AF       3700        XOR  A
94EF ED580C0C 3710        LD   DE,(£0C0C); LADE ARG1
94F3 ED52     3720        SBC  HL,DE
94F5 E5       3730        PUSH HL
```

```
94F6 44       3740        LD   B,H
94F7 4D       3750        LD   C,L
94F8 2A0C0C   3760        LD   HL,(£0C0C); LADE ARG1
94FB E5       3770        PUSH HL
94FC 54       3780        LD   D,H
94FD 5D       3790        LD   E,L
94FE 13       3800        INC  DE
94FF 3600     3810        LD   (HL),£00
9501 EDB0     3820        LDIR
9503 E1       3830        POP  HL
9504 C1       3840        POP  BC
9505 1600     3850 ZCHK   LD   D,£00
9507 AF       3860        XOR  A
9508 BE       3870        CP   (HL)
9509 3E41     3880        LD   A,£41
950B C49395   3890        CALL NZ,ERR
950E 3E01     3900        LD   A,£01
9510 77       3910 WALK   LD   (HL),A
9511 BE       3920        CP   (HL)
9512 F5       3930        PUSH AF
9513 3E42     3940        LD   A,£42
9515 C49395   3950        CALL NZ,ERR
9518 F1       3960        POP  AF
9519 2003     3970        JR   NZ,ALFF
951B 17       3980        RLA
951C 30F2     3990        JR   NC,WALK
951E 3EFF     4000 ALFF   LD   A,£FF
9520 77       4010        LD   (HL),A
9521 BE       4020        CP   (HL)
9522 3E43     4030        LD   A,£43
9524 C49395   4040        CALL NZ,ERR
9527 7A       4050        LD   A,D
9528 B7       4060        OR   A
9529 C4A095   4070        CALL NZ,NEWLI
952C EDA0     4080        LDI
952E EA0595   4090        JP   PE,ZCHK
9531 EF       4100        RST  £28
9532 0D       4110        DEFB £0D  .
9533 50617274 4120        DEFM /Part 2  Op-Code Fetch Test/
     20322020
     4F702D43
     6F646520
     46657463
     68205465
     7374
954D 0D00     4130        DEFB £0D,£00 ;..
954F 2A0E0C   4140        LD   HL,(£0C0E); LADE ARG2
9552 AF       4150        XOR  A
9553 ED5B0C0C 4160        LD   DE,(£0C0C); LADE ARG1
9557 ED52     4170        SBC  HL,DE
9559 44       4180        LD   B,H
955A 4D       4190        LD   C,L
955B 0B       4200        DEC  BC
955C 0B       4210        DEC  BC
955D EB       4220        EX   DE,HL
955E E5       4230 OPLOP  PUSH HL
```

```
955F 3EC3     4240        LD   A,£C3
9561 77       4250        LD   (HL),A
9562 23       4260        INC  HL
9563 117A95   4270        LD   DE,RETUR
9566 73       4280        LD   (HL),E
9567 23       4290        INC  HL
9568 72       4300        LD   (HL),D
9569 E1       4310        POP  HL
956A ED5B290C 4320        LD   DE,(£0C29)
956E C5       4330        PUSH BC
956F DF       4340        RST  £18
9570 66       4350        DEFB £66 ;f
9571 22800C   4360        LD   (£0C80),HL
9574 ED53290C 4370        LD   (£0C29),DE
9578 C1       4380        POP  BC
9579 E9       4390        JP   (HL)
957A EDA0     4400 RETUR  LDI
957C EA5E95   4410        JP   PE,OPLOP
957F EF       4420        RST  £28
9580 0D       4430        DEFB £0D;£0D
9581 4D656D6F 4440        DEFM /Memory is OK/
     72792069
     73204F4B
958D 0D0D00   4450        DEFB £0D,£0D,£00 ;..
9590 DF       4460        RST  £18
9591 5B       4470        DEFB £5B ;[
9592 00       4480        NOP
9593 F5       4490 ERR    PUSH AF
9594 F7       4500        RST  £30
9595 DF       4510        RST  £18
9596 69       4520        DEFB £69 ;i
9597 C5       4530        PUSH BC
9598 DF       4540        RST  £18
9599 66       4550        DEFB £66 ;f
959A C1       4560        POP  BC
959B DF       4570        RST  £18
959C 69       4580        DEFB £69 ;i
959D 14       4590        INC  D
959E F1       4600        POP  AF
959F C9       4610        RET
95A0 DF       4620 NEWLI  RST  £18
95A1 6A       4630        DEFB £6A ;j
95A2 C9       4640        RET

95A3 FE58     4660 DRUCK  CP   "X
95A5 2808     4670        JR   Z,INIT
95A7 FE4D     4680        CP   "M
95A9 CA00A0   4690        JP   Z,£A000 ; MDCR-Monitor
95AC C30090   4700        JP   START ; WEITERE UEBERPRUEFUNGEN!!

95AF F5       4720 INIT   PUSH AF ;DRUCKER VERS.1 26.6.81 RM
95B0 E5       4730        PUSH HL     ;CENTRONICS SCHNITTSTELLE
95B1 216FC3   4740        LD   HL,£C36F   ; INITIALISIERE ASSEMBLER
95B4 22810C   4750        LD   (£0C81),HL ; FUER SERIELLE AUSGABE
95B7 21DD95   4760        LD   HL,AUSG    ; INCL. DRUCKERAUSGABE
95BA 22830C   4770        LD   (£0C83),HL ; (ROUT. AUF £0C80)
```

```
95BD 3EFF     4780        LD   A,£FF      ; INITIALISIERE PORT A
95BF D312     4790        OUT  (£12),A    ; ZUR DATENAUSGABE
95C1 AF       4800        XOR  A
95C2 D312     4810        OUT  (£12),A
              4820 ; Das Initial-Signal ist nicht unbedingt
              4830 ; erforderlich.
95C4 3EFF     4840        LD   A,£FF      ; INITIALISIERE PORT B
95C6 D313     4850        OUT  (£13),A    ; BIT 0 = STROBEIMPULS
95C8 3E02     4860        LD   A,£02      ; BIT 1 = BUSYSIGNALEING.
95CA D313     4870        OUT  (£13),A    ;
95CC 3E01     4880        LD   A,£01      ; SETZE STROBE AUF HIGH
95CE D311     4890        OUT  (£11),A    ;
              4900 ; Es wird nur die TOOLKIT Outputroutine
              4910 ; geloescht. Inputroutine (£0C78) bleibt
              4920 ; erhalten
95D0 21DD95   4930        LD   HL,AUSG    ; INITIALISIERE MONITOR
95D3 22780C   4940        LD   (£0C78),HL ; FUER DRUCKERAUSGABE
95D6 DF       4950        RST  £18
95D7 55       4960        DEFB £55        ; FUEHRE DAS U-KOMMANDO AUS
95D8 E1       4970        POP  HL
95D9 F1       4980        POP  AF
95DA C37291   4990        JP   ANFNG      ; ZURUECK INS MENUE
95DD F5       5000 AUSG   PUSH AF  ; STAMMROUTINE
95DE C5       5010        PUSH BC
95DF D5       5020        PUSH DE
95E0 E5       5030        PUSH HL
95E1 FE70     5040        CP   "p
95E3 2829     5050        JR   Z,AUSGP
95E5 FE67     5060        CP   "g
95E7 2829     5070        JR   Z,AUSGG

95E9 FE0A     5090        CP   £0A         ; UNTERDR. LINEFEED
95EB 2804     5100        JR   Z,NULLA     ; FUER ASSEMBLER
95ED FEA0     5110        CP   £A0         ; SETZE £0D FUER £A0
95EF 2002     5120        JR   NZ,AUACI    ; AUSGABE ASCII
95F1 3E00     5130 NULLA  LD   A,£00 ; SKIP CODE FUER ASSEMBLER
95F3 D310     5140 AUACI  OUT  (£10),A; DATEN AUF PORT A LEGEN
95F5 DB11     5150 BUSY1  IN   A,(£11) ; DRUCKER BESCHAEFTIGT?
95F7 CB4F     5160        BIT  1,A          ; WENN JA, DANN WARTE
95F9 20FA     5170        JR   NZ,BUSY1
95FB 3E00     5180        LD   A,£00        ; SETZE STROBE IMPULS
95FD D311     5190        OUT  (£11),A
95FF 3E01     5200        LD   A,£01
9601 D311     5210        OUT  (£11),A
9603 DB11     5220 BUSY2  IN   A,(£11) ; DRUCKER BESCHAEFTIGT?
9605 CB4F     5230        BIT  1,A          ; WENN JA, DANN WARTE
9607 20FA     5240        JR   NZ,BUSY2
9609 E1       5250        POP  HL
960A D1       5260        POP  DE
960B C1       5270        POP  BC
960C F1       5280        POP  AF
960D C9       5290        RET  ; ENDE DER DRUCKROUTINE

              5310 ; Grafikberechnung spez. fuer SEIKO GP80
              5320 ; Hier Berechnung fuer Zeichentabelle
960E 3E01     5330 AUSGP  LD   A,£01    ; ADRESSE FUER "p"
9610 1804     5340        JR   AUGRA
9612 3E02     5350 AUSGG  LD   A,£02    ; ADRESSE FUER "g"
9614 1800     5360        JR   AUGRA
9616 214296   5370 AUGRA  LD   HL,TABEL ; TABELLENANFANG IN HL
9619 110900   5380        LD   DE,£09   ; ANZAHL GRAF.ZEICHEN
961C 43       5390        LD   B,E
961D 3D       5400 ZEICH  DEC  A  ; BERECHNE TABELLEN-
961E 2803     5410        JR   Z,GRAAF  ; ANFANG FUER DAS GE-
9620 19       5420        ADD  HL,DE    ; WUENSCHTE ZEICHEN
9621 18FA     5430        JR   ZEICH
9623 7E       5440 GRAAF  LD   A,(HL)  ; LADE TABELLE
9624 D310     5450        OUT  (£10),A      ; DATEN AUF PORT A LEGEN
9626 DB11     5460 BUSY3  IN   A,(£11) ; DRUCKER BESCHAEFTIGT?
9628 CB4F     5470        BIT  1,A          ; WENN JA, DANN WARTE
962A 20FA     5480        JR   NZ,BUSY3
962C 3E00     5490        LD   A,£00        ; SETZE STROBE-IMPULS
962E D311     5500        OUT  (£11),A
9630 3E01     5510        LD   A,£01
9632 D311     5520        OUT  (£11),A
9634 DB11     5530 BUSY4  IN   A,(£11) ; DRUCKER BESCHAEFTIGT
9636 CB4F     5540        BIT  1,A          ; WENN JA DANN WARTE
9638 20FA     5550        JR   NZ,BUSY4
963A 23       5560        INC  HL
963B 10E6     5570        DJNZ GRAAF
963D E1       5580        POP  HL
963E D1       5590        POP  DE
963F C1       5600        POP  BC
9640 F1       5610        POP  AF
9641 C9       5620        RET  ; ENDE DER DRUCKROUTINE

9642 08       5640 TABEL  DEFB £08 ; GRAFIKTABELLE
9643 80F89494 5650        DEFB £80 £F8 £94 £94 £94 £88 £80 £0F
     9488800F
9648 0880C8D4 5660        DEFB £08 £80 £C8 £D4 £04 £D4 £EA £80 £0F
     D4D4EA80
     0F

              5690 ; PS. Eine einfache Moeglichkeit fuer
              5700 ;     NASCOM 1 - Benutzer zum Menuestart
              5710 ;     ist folgende :
              5720 ; Im Nassys Eprom wird im Speicherplatz
              5730 ; £070A = £00 und in £0708 = £A0
              5740 ; eingebrannt. Man kann nun das Menue
              5750 ; ueber die "J"-Taste aufrufen; bisher
              5760 ; Basic - Kaltstart.

ZEAP Z80 Assembler - Symbol Table

951EH 4000 ALFF         9333H 2000 ALL
9172H 0750 ANFNG        95F3H 5140 AUACI
9616H 5370 AUGRA        95DDH 5000 AUSG
9612H 5350 AUSGG        960EH 5330 AUSGP
93DEH 2410 BACK         91CDH 1260 BASIC
95F5H 5150 BUSY1        9603H 5220 BUSY2
9626H 5460 BUSY3        9634H 5530 BUSY4
```

```
92A0H 1750 CLEAR        93ACH 2350 COMP
9248H 1470 CONT         9283H 1600 DIS2
91DAH 1320 DISAS        9257H 1550 DISD1
95A3H 4680 DRUCK        9467H 3400 ENDE
9593H 4490 ERR          949FH 3510 GOON
9623H 5440 GRAAF        95AFH 4720 INIT
9390H 2100 INP          9246H 1460 JUMP
933AH 2030 LOOP1        9339H 3050 LOOP2
91C7H 1220 NASYS        95A0H 4620 NEW_I
95F1H 5130 NULLA        955EH 4230 OPLOF
9C00H 0210 PROG         9800H 0220 PROG
91A2H 1040 PROG         94C0H 3590 SORT1
94A6H 3550 RAMTE        9347H 2110 RANGE
957AH 4400 RETUR        943EH 3040 ROUT1
9438H 3010 ROUT2        9443H 3150 ROUT3
944FH 3210 ROUT4        93F5H 2550 SPRG1
9424H 2845 SPRG2        9430H 2920 SPRG4
9436H 2970 SPRG5        9463H 3360 SPRG6
9456H 3290 SPRG7        9000H 0440 START
9642H 5640 TABEL        9510H 3910 WALK
9195H 0970 WEIT1        91ADH 1090 WEIT2
91B8H 1140 WEIT3        9505H 3850 ZCHK
961DH 5400 ZEICH
```

Umrechnung für die Druckroutine:

ext.Port	int.Port
10	4
11	5
12	6
13	7

```
1 FOLGENDE ZEILEN SIND IM MENUEPROGRAMM
2 ZU AENDERN BZW. ZU ERGAENZEN, DAMIT KEINE
3 PROBLEME AUFTRETEN, WENN UNTER DER MDCR-
4 ROUTINE DIE DRUCKROUTINE AUFGERUFEN WIRD:
5 .
444 RST £18 ; SETZE NAS-SYS IN NORMAL-MODE
446 DEFB "N ; UND VERHINDERE INTERRUPTS
448 DI
4950 LD HL,AUSG    ; INITIALISIERE MONITOR
4940 LD (£0C78),HL ; FUER DRUCKERAUSGABE
4942 LD HL,£002F
4944 LD (£0C7B),HL
```

AD-Wandlung

von Michael Bach

Hier eine Anmerkung zum kürzlich veröffentlichten Analog-Interface (P.Bentz, Heft 11/12-81).

Es gibt eine meiner Meinung nach wesentlich einfachere Lösung, wie sie im Analog Devices AD7581 (8 Kanäle a 8 Bit, DM 50.-) verwirklicht ist: der ADC wandelt ständig der Reihe nach die Spannungen aller Kanäle um und schreibt sie in ein 2-Port-Ram (mit im IC), das nach Bedarf vom Rechner ausgelesen wird (8 I/O-Adressen). Dieser Wert ist nie älter als 0,7ms. Damit spart man sich den ganzen Aufwand für Start Conversion, Warten oder gar Interrupt. Für ganz schnelle Anwendungen natürlich nicht das Richtige.

Ich habe eine I/O-Platine (für ein 6809-System) damit gebaut, welches sehr zufriedenstellend fuktioniert.

Sortieren in Basic

Teil 7 von W. Mayer-Gürr

Was beim Heapsort schon zu einer Verbesserung der Effizienz - nämlich Vergleiche über eine größere Distanz - geführt hat, wird beim Quicksort noch konsequenter angewandt. Das mittlere Element eines Feldes dient als Bezugspunkt. Von links und rechts nähert man sich der Mitte und tauscht gegebenenfalls das größere Element auf die rechte Seite. Die Feldhälften werden wiederum geteilt und die Hälften, wie oben beschrieben, behandelt.

Der dazugehörige Algorithmus läßt sich in BASIC nur mit einigen Verrenkungen verwirklichen. Er führt aber zu der schnellsten (mir bekannten) Sortierung eines Feldes. Wegen des größeren Umfangs des Programms lohnt er sich bei kleineren Feldern wohl nicht.

Das hier aufgelistete Programm läßt sich auf Kosten der Lesbarkeit noch um ca. 10% schneller machen, wenn man die Tios aus Teil 2 berücksichtigt. Um 100 Integerzahlen zu sortieren, braucht meine Anlage mit CLD-DOS und dem MICROSOFT BASIC etwa 15 Sekunden.

```
100  REM  ********************************
110  REM  *       QUICK-SORT               *
120  REM  ********************************
130 N = 10
140  REM      N = ANZAHL DER ELEMENTE
150  DIM NA$(N)
160  FOR I = 1 TO N
170  PRINT "NR. ";I; TAB( 8);
180  INPUT NA$(I)
190  NEXT I
200  GOSUB 400: REM  * ZUM SORTIEREN
210  FOR I = 1 TO N
220  PRINT NA$(I)
230  NEXT I
240  END
400 M = 1
410 I = 1
420 J = N
430  IF I >  = J THEN 800
440 K = I
450 IN =  INT ((J + I) / 2)
460 HI$ = NA$(IN)
470  IF NA$(I) <  = HI$ THEN 510
480 NA$(IN) = NA$(I)
490 NA$(I) = HI$
500 HI$ = NA$(IN)
510 L = J
520  IF NA$(J) >  = HI$ THEN 630
530 NA$(IN) = NA$(J)
540 NA$(J) = HI$
550 HI$ = NA$(IN)
560  IF NA$(I) <  = HI$ THEN 630
570 NA$(IN) = NA$(I)
580 NA$(I) = HI$
590 HI$ = NA$(IN)
600  GOTO 630
```

```
610 NA$(L) = NA$(K)
620 NA$(K) = PL$
630 L = L - 1
640  IF NA$(L) > HI$ THEN 630
650 PL$ = NA$(L)
660 K = K + 1
670  IF NA$(K) < HI$ THEN 660
680  IF K <  = L THEN 610
690  IF L - I <  = J - K THEN 750
700 X(M) = I
710 Y(M) = L
720 I = K
730 M = M + 1
740  GOTO 840
750 X(M) = K
760 Y(M) = J
770 J = L
780 M = M + 1
790  GOTO 840
800 M = M - 1
810  IF M < 1 THEN  RETURN
820 I = X(M)
830 J = Y(M)
840  IF J - I > 10 THEN 440
850  IF I = 1 THEN 430
860 I = I - 1
870 S = I
880 I = I + 1
890  IF I = J THEN 800
900 HI$ = NA$(I + 1)
910 K = I
920  IF K = S THEN 940
930  IF HI$ < NA$(K) THEN NA$(K + 1) = NA$(K)
     :K = K - 1: GOTO 920
940 NA$(K + 1) = HI$
950  GOTO 880
```

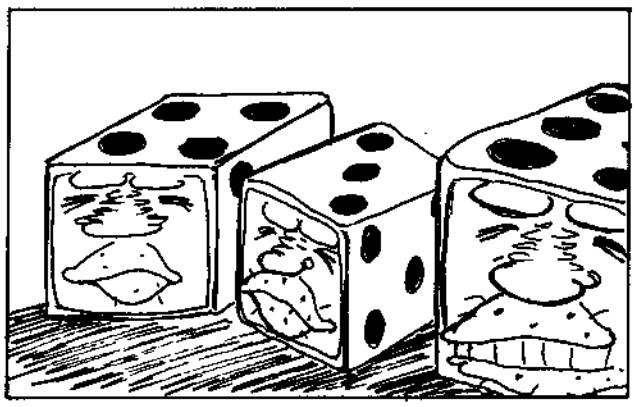
Wesentlich eleganter ist die Lösung in einer höheren Programmiersprache, die echte Rekursion ermöglicht. Dort gelten Variablen eines sich selbst aufrufenden Unterprogramms nur während der Ausführung dieses Programmteils. Die Prozedur zum Sortieren von 100 Integerelementen braucht bei meinem PASCAL nur 4 Sekunden.

```
PROCEDURE QUICKSORT ( NIEDRIG,HOCH : INTEGER ) ;
  VAR
      I,J,MITTE,TAUSCH : INTEGER ;
    BEGIN
      I:=NIEDRIG; J:=HOCH;
      MITTE:=LISTE[(I+J) DIV 2 ] ;
      REPEAT
        WHILE LISTE[I] < MITTE DO I:=I+1;
        WHILE LISTE[J] > MITTE DO J:=J-1;
        IF I <= J THEN
        BEGIN
          TAUSCH:=LISTE[I];
          LISTE[I]:=LISTE[J];
          LISTE[J]:=TAUSCH;
          I:=I+1;
          J:=J-1;
        END;
      UNTIL I > J ;
      IF NIEDRIG < J THEN  QUICKSORT(NIEDRIG,J);
      IF I < HOCH THEN QUICKSORT(I,HOCH);
    END;
```

Würfelspiel

von Ottmar Schweizer

```
10 DIM D(12) : CLS
11 PRINT:PRINT:PRINT
20 PRINT "W U E R F E L S P I E L"
21 PRINT: PRINT: PRINT
70 PRINT "DIE NAMEN VON"
80 INPUT "DREI SPIELERN:";A$(1),A$(2),A$(3)
90 CLS: Q=0 : L=0
100 FOR I=2 TO 12 :D(I)=0: NEXT I
110 U=1 : Y=1 : Z=40
120 FOR U =2 TO 12 : SCREEN 42,U-1 : PRINT U
 : NEXT U
130 V=INT(10*RND(1))+1: IF V>6 GOTO 130
140 W=INT(10 * RND (1))+1: IF W>6 GOTO 140
150 I=V+W: D(I)=D(I)+1: X=Z-D(I)
160 SCREEN X,I-1 : PRINT"X":Q=Q+1
170 IF X=20 GOTO 260
180 IF X=2 GOTO 200
190 GOTO 130
200 FOR B=1 TO 3
210 IF(A(B)=I) GOTO 340
220 NEXT B
230 IF L=0 THEN SCREEN 4,12 :PRINT "HA,HA"
240 SCREEN4,14 : PRINT Q"WUERFE": INPUT"WEIT
ER";Z
250 GOTO 90
260 IF Y>1 GOTO 130
270 SCREEN 4,12
280 PRINT"IHREN TIP, BITTE !"
290 SCREEN 30,12: PRINT A$(1);:INPUT A(1)
300 SCREEN 30,13: PRINT A$(2);:INPUT A(2)
310 SCREEN 30,14: PRINT A$(3);:INPUT A(3)
320 SCREEN 4,12: PRINT"                  "
330 Y=Y+1 : GOTO 130
340 SCREEN 1,10+L
350 PRINT"SIEGER IST:   "A$(B)
360 L=L+1 : GOTO 220
370 END
```



FORTH für den NASCOM

Teil 5 von Günter Kreidl

1) Fehlerkorrektur

Leider hat sich beim Abtippen der FORTH-Erweiterungen im Oktoberheft ein Fehler bei der Funktion MESSAGE eingeschlichen. Richtig lautet sie:

```
: MESSAGE  GETWORD ENTER VARBL CMPLW
CODEADR PEEKW
REPEAT CIN DUP COUT DUP 34 EQ UNTIL
DUP 8 EQ IF
POP DEC ELSE OVER POKEB INC THEN
LOOP
POP ZERO OVER POKEB INC CODEADR POKEW ;
```

Ich hoffe, der Fehler hat nicht zuviel Frust verursacht, und bedanke mich bei Herrn Waltenberger, der ihn entdeckt hat.

2) Verbesserung der Initialisierungsroutine

Wer sich bei den Erweiterungen bis zum Editor-Modus hindurchgearbeitet hat, der wird eine unangenehme Erfahrung machen, wenn er den Editor-Modus einmal mit RESET verlassen mußte; der Interpreter läßt sich nicht mehr starten - es hilft nur neues Laden von Cassette. Hier läßt sich leicht Abhilfe schaffen, da sich in der Initialisierungsroutine der erweiterten Fassung einige "NOP's" befinden (von 1003H bis 100EH). Hier kann man nun zwei Befehle eintragen, die den Interpreter auch nach RESET im Editor-Modus wieder neu starten lassen. Da die jeweiligen Adressen der Variablen INVAR und der Konstanten BLADR bei verschiedenen Benutzern je nach Reihenfolge der Eingabe unterschiedlich sein mögen, wollen wir sie durch den Interpreter suchen lassen:

```
BLADR PRINTHEX SPACE INVAR PRINTHEX
1747  1774          (Antwort des Interpreters)
```

Nun können wir bei 1003H eintragen:

```
1003 21 47 17 22 74 17.
```

Dadurch wird bei jedem Start als Einleseroutine BLINK aufgerufen. (Der Editor-Modus arbeitet mit BLINK und READSCR. Der Interpreter-Modus hingegen kann nicht mit READSCR arbeiten.)

3) Compilererweiterungen

Die folgende Funktion " :: " ermöglicht es, neue Compiler-Funktionen zu definieren. Zunächst wird eine Interpreter-Funktion compiliert, dann wird deren Name an das Compiler-Dictionary angehängt und aus dem Dictionary des Interpreters wieder gelöscht. Das Compiler-Dictionary wächst also "nach oben", das Interpreter-Dictionary dagegen "nach unten".

```
: :: : NAMES PEEKW DUP PEEKB 3 + OVER OVER
MEMORY PEEKW SWAP MOVEBYTES SWAP
OVER + NAMES POKEW MEMORY PEEKW +
DUP MEMORY POKEW ZERO SWAP POKEB ;
```

Um die folgenden Funktionen zu verstehen, die dem Compiler hinzugefügt werden, muß man sich noch einmal die Arbeitsweise des Compilers vergegenwärtigen: Außer Zahlen erkennt er Interpreterfunktionen, die dann compiliert werden, und Compilerfunktionen, die sofort ausgeführt werden. Manchmal ist es aber wünschenswert, im Compiler-Modus eine Interpreterfunktion direkt ausführen zu lassen, bzw. im "::"-Modus eine Compilerfunktion eben nicht ausführen, sondern compilieren zu lassen. Dies geschieht durch Voranstellen der Zeichen (! , ").

```
:: !  GETWORD NAMES PEEKW LOOKUP
IF  EXECUTE
ELSE ERROR TYPE PRINTS THEN  ;  ,
4759 CONSTANT CNAMES  zeigt auf variable
:: "  GETWORD CNAMES PEEKW LOOKUP
IF  CMPLW
ELSE ERROR TYPE PRINTS THEN  ;
```

Nicht als bleibende Erweiterung des Systems, sondern als ein Beispiel für die Anwendung der oben definierten Funktionen sei eine Kommentarfunktion mit "(" und ")" definiert:

```
:: (  REPEAT GETWORD INC PEEKB 41 EQ UNTIL
LOOP ;
```

Innerhalb der Klammern kann man nun beliebigen Text in die Funktion einfügen, der vom Compiler ignoriert wird (Vergl. "REM" in BASIC!). Ich persönlich ziehe aber die Funktion MISCHTEXT vor, die Kommentare in Kleinschrift erlaubt.

Ein zweites Beispiel (ebenfalls nicht als bleibende Erweiterung gedacht) verwendet die Klammern als Wiederholungsfunktion:

```
:: (  ONE 4302    adresse der funktion tpush
CMPLW CMPLW " FOR  ;
:: )  " LOOP  ;
```

Beispiel für die Anwendung:

```
: COUNT   ( I = SPACE ) ;
```

Auf die Eingabe "5 COUNT" antwortet der Interpreter mit:

```
1 2 3 4 5
```

Die "FORTH-Maschine"

Die folgenden Hinweise sind für diejenigen gedacht, die sich nicht nur für die Anwendung sondern auch für den inneren Aufbau des FORTH-Interpreters interessieren.

Die Routinen NEXT, TCALL und TRET bilden zusammen einen kleinen rekursiven Interpreter. NEXT ruft jeweils das nächste Unterprogramm auf (sowohl Maschinencode als auch Threaded Code) und ist somit die am häufigsten benutzte Routine. TCALL und TRET leiten jede Threaded-Code-Routine ein bzw. beenden sie. Es ist klar, daß die Arbeitsgeschwindigkeit des ganzen Systems hauptsächlich von der Geschwindigkeit dieser drei Routinen bestimmt ist. Hier die Arbeitsgeschwindigkeit der jeweiligen Routinen in Taktzyklen:

NEXT: 90 TCALL: 104 TRET: 88

Christoph Rau, der Ihnen allen durch sein schönes REVERSI-Programm bekannt ist, arbeitet ebenfalls an einer FORTH-Version. Seine entsprechenden Routinen sind um ein Erhebliches schneller, haben aber den Nachteil, daß die Register BC und IX ständig von diesen Routinen belegt sind und deshalb nicht anderweitig verwendet werden können. Hier die Arbeitsgeschwindigkeit in Taktzyklen:

NEXT: 62 TCALL: 74 TRET: 58

Bei dem Versuch, einen entsprechenden Interpreter für einen anderen Prozessor zu entwerfen, kam ich noch zu einer anderen Form eines rekursiven Interpreters, der aber mit dem CALL-Befehl arbeitet, wodurch der Stack nicht mehr für die Parameterübergabe benutzt werden kann. Es muß dann eine Art "Ersatz-Stack" konstruiert werden, der zwar langsamer arbeitet, aber auch für größere Datenformate (Strings, BCD-Zahlen) geeignet ist. Ein weiterer Vorteil ist, daß alle Unterprogramme (sowohl in Maschinen- als auch im Threaded Code) von anderen Programmen mit einfachen CALL-Befehlen aufgerufen werden können. Dies ist möglich, weil bei dieser Interpreterversion Maschinencode-Unterprogrammen nicht mehr ihre Startadresse vorangestellt werden muß, und weil Threaded-Code-Routinen mit "CALL TCALL" statt mit "DEFW TCALL" beginnen. Beide Interpreter folgen unten im Assemblertext, und ich mache hier lieber Schluß, da sonst das NASCOM-Journal womöglich noch des "esoterischen Rekursivismus" beschuldigt wird.

Arbeitsgeschwindigkeit der Interpreterroutinen: NEXT: 80 TCALL: 33 TRET: 38

```
                        0010 ; Rekursiver Interpreter
                        0020 ;
                        0030 ;1) FORTH-Interpreter
                        0040 ;von Christoph Rau
                        0050 ;nach R.G.Loeliger
                        0060 ;BC = Programmzähler
                        0070 ;IX = Return-Stack-Pointer
0D80                    0080        ORG  #D80
0D80 820D               0090 TRET   DEFW $+2
0D82 DD4E00             0100        LD   C,(IX+0)
0D85 DD23               0110        INC  IX
0D87 DD4600             0120        LD   B,(IX+0)
0D8A DD23               0130        INC  IX
0D8C 03                 0140 NEXT   INC  BC
0D8D 0A                 0150        LD   A,(BC)
0D8E 6F                 0160        LD   L,A
0D8F 03                 0170        INC  BC
0D90 0A                 0180        LD   A,(BC)
0D91 67                 0190        LD   H,A
0D92 5E                 0200        LD   E,(HL)
0D93 23                 0210        INC  HL
0D94 56                 0220        LD   D,(HL)
0D95 EB                 0230        EX   DE,HL
0D96 E9                 0240        JP   (HL)
0D97 DD2B               0250 TCALL  DEC  IX
0D99 DD7000             0260        LD   (IX+0),B
0D9C DD2B               0270        DEC  IX
0D9E DD7100             0280        LD   (IX+0),C
0DA1 4B                 0290        LD   C,E
0DA2 42                 0300        LD   B,D
0DA3 C38C0D             0310        JP   NEXT

0D80                    0370        ORG  #D80
0D80 D9                 0380 TCALL  EXX
0D81 D1                 0390        POP  DE
0D82 E5                 0400        PUSH HL
0D83 EB                 0410        EX   DE,HL
0D84 D9                 0420        EXX
0D85 21850D             0430 NEXT   LD   HL,NEXT
0D88 E5                 0440        PUSH HL
0D89 D9                 0450        EXX
0D8A 5E                 0460        LD   E,(HL)
0D8B 23                 0470        INC  HL
0D8C 56                 0480        LD   D,(HL)
0D8D 23                 0490        INC  HL
0D8E D5                 0500        PUSH DE
0D8F D9                 0510        EXX
0D90 E1                 0520        POP  HL
0D91 E9                 0530        JP   (HL)
0D92 D9                 0540 TRET   EXX
0D93 D1                 0550        POP  DE
0D94 E1                 0560        POP  HL
0D95 D9                 0570        EXX
0D96 C9                 0580        RET

0320 ;2) Fädelcode-Interpreter
0330 ;mit indirektem CALL
0340 ;benutzt die Austauschregister
0350 ;Fädelcode-Routinen müssen mit
0360 ;"CALL TCALL" beginnen
```

Februar 82 G.K.

Disketten-tauschservice

Beteiligung an der Tauschaktion:
Bitte eine (moeglichst mit vielen Programmen bespielte) Diskette einsenden und 5,-DM (Schein oder Briefmarken) fuer Verpackung, Porto usw. beilegen.

Wolfgang Mayer-Guerr
[redacted]
[redacted]Recklinghausen
Tel. [redacted]

Der Einsender eines Programms muss nicht immer auch der Urheber sein!

Stand: 10-FEB-82

Nr. Filename Sektoren Kurzbeschreibung

Hilfsprogramme fuer Maschinensprache

A001	CNVHD	.ACM	8	Konvertiert HEX in DEZ
A002	CNVBA	.ACM	3	" Bin in ASCII
A003	OUTWRD	.ACM	3	Word an das Terminal
A004	OUTBYT	.ACM	2	Gibt Byte aus
A005	OUTHEX	.ACM	3	Hex an Terminal
A006	CNVAB	.ACM	3	ASCII zu Hex
A007	INSTR	.ACM	5	Einlesen eines Strings
A008	INHEX	.ACM	5	Einlesen einer HexZahl
A009	RDFILE	.ASM	9	Einlesen eines Files
A010	WRFILE	.ASM	10	Abspeichern eines Files
A011	TD	.ASM	9	Umdefinieren von Tasten
A012	UHR	.ASM	10	Interrupt-Uhr
A013	START	.ACM	2	Obergrenze Workspace
A014	COMAND	.ACM	3	Befehlsschleife
A015	NOBLANK	.ACM	1	ignoriert Blanks
A016	INCHAR	.ACM	2	Zeichen v. Term.
A017	INLINE	.ACM	3	Zeile vom Terminal
A018	HEXOUT	.ACM	2	Ausgabe von Hexzahlen
A019	STRBUF	.ACM	14	Stringbuffer
A020	HEXNUM	.ACM	4	String in Zahl wandeln
A021	HEXIN	.ACM	2	Hexzahl lesen

Programme zur Diskettenverwaltung

B001	CHVOL	.ABS	10	Aendern der Disketten-nummer
B002	CLRFLA	.ABS	5	Loescht alle (auch L) Flags
B003	DIAGNOSE	.ABS	3	Erweiterter STATUS-Befehl
B004	FORMAT	.BAS	6	Rekonstruiert Formatspur

Hilfsprogramme in Maschinensprache

C001	REVAS	.ABS	13	NAS-DIS (nur zum Laden)
C002	DASM	.ABS	14	8080 Disassembler
C003	TF	.ASM	15	Devicedriver f. Lochstreifenleser
C004	FC	.ABS	7	Vergleicht 2 Files
C005	DISASM	.ABS	16	Komf. Z80 Disassembler
C006	SORT	.ABS	4	Sortierprogramm
C007	SUBMIT	.ABS	1	Verkettet DOS-Befehle
C008	PICPATCH	.ABS	4	Hilfe f. DBUG u. DISASM
C009	DISAS	.ABS	15	Z-80 Disassembler

Spiele in Maschinensprache

D001	KEKS	.ABS	2	Zweidimensional, NIM aehnlich
D002	RATE1	.ABS	2	Zahlenraten
D003	RATE2	.ABS	3	Zahlenraten
D004	CRAPS	.ABS	4	Wuerfelspiel
D005	BLACK	.ABS	8	Blackjack Kartenspiel
D006	DREH	.ABS	2	Zahlendreh
D007	HURKEL	.ABS	3	Ratespiel
D008	UFO	.ABS	2	Ufojagd
D009	STAR	.ABS	17	Enterprise jagt Klingons
D010	SUPER	.ABS	3	Mastermind gegen Computer
D011	KLINGON	.ABS	9	Klingons fangen
D012	KONZENTR	.ABS	7	Gedaechtnistraining
D013	REVERSI	.ABS	92	verbess. Othello (Fortran)

Hilfsprogramme fuer Microsoftbasic

E001	SHAKER	.BAS	3	Sortieralgorithmus
E002	HAURUCK	.BAS	3	"
E003	BUBBLE	.BAS	2	"
E004	INSERT	.BAS	3	"
E005	SHELL	.BAS	3	"
E006	HEAP	.BAS	3	"
E007	QUICK	.BAS	6	"
E008	EXPO	.BAS	12	Umrechnungen HEX,BIN,OCT
E009	UPSIRENE	.BAS	2	Unterprogramm Sirene
E010	RELOC	.BAS	9	Relociertes Maschinensprache programm mit Tastendefinition (erfoerdert E011)
E011	EXT	.ABS	1	
E012	RELOCA	.BAS	9	wie E010,laedt automatisch
E013	DATABANK	.BAS	6	Verwalten einer Datenbank
E014	ASMUT	.BAS	18	wandelt DISASM-Progr. in ASEM-faehigen Code
E015	RANDAUSG	.BAS	4	Druckausgabe m.Randausgleich
E016	WEEKDAY	.BAS	7	berechnet Wochentag

Mathematik/Naturwissenschaft CLD-BASIC

F001	TITRATIO	.BAS	20	Titration Saeure-Base
F002	KURDIS	.BAS	13	Kurvendiskussion
F003	KINETIK	.BAS	20	Reaktionskinetik
F004	CODON	.BAS	18	Aminosaeurensynthese

Spiele in CLD-BASIC

G001	WUMPUS	.BAS	32	Jagd in 20 Hoehlen
G002	WUMPUS2	.BAS	28	Mehrere Hoehlensysteme
G003	CSR	.BAS	8	Spektroskopie
G004	ELIZA	.BAS	29	Psychoanalyse
G005	OREGON	.BAS	72	Treck nach Oregon
G006	MONDLAN	.BAS	13	Mondlandung
G007	RODEO	.BAS	12	Cowboy faengt Pferd
G008	MILLI	.BAS	34	Wie werde ich Millionaer
G009	MILGEWIN	.DAT	1	Bester Spieler von G008
G010	INTERCEP	.BAS	14	Rette dich vor Robotern
G011	TICTAC	.BAS	6	Minimuehle
G012	WAHL	.BAS	57	Wahl des US-Praesidenten
G013	KRAFTWER	.BAS	43	Energieversorgungsbetrieb
G014	CLEWSO	.BAS	28	Detektivspiel
G015	SCHIFFE	.BAS	43	Schiffe versenken
G016	WALD	.BAS	63	Such den Schatz
G017	WALD	.DAT	1	Bester Spieler von G016
G018	KKW	.BAS	52	Kernkraftwerk
G019	KKW	.DAT	1	Bester Spieler von G018
G020	BOERSE	.BAS	19	Spekulieren mit Aktien
G021	HAMURABI	.BAS	16	Regiere eine Stadt
G022	SUPERMAN	.BAS	36	Fang den Boesewicht
G023	LIMONADE	.BAS	34	Verdiene viel
G024	APOLLO	.BAS	18	Mondlandung
G025	REVERS	.BAS	14	Reversi (Othello)
G026	DAME	.BAS	14	Dame
G027	STARTREK	.BAS	34	Enterprise jagt Klingons

Spiele in MBASIC

```
H001 ELIZA    .BAS  25 Psychoanalyse
H002 MONDLAN .BAS  11 Mondlandung
H003 RODEO    .BAS   8 Cowboy faengt Pferd
H004 MILLI    .BAS  27 Wie werde ich Millionaer
H005 MILGEWIN.DAT   1 Bester Spieler von H004
H006 INTERCEP.BAS  12 Entkomme den Robotern
H007 TICTAC   .BAS   4 Minimuehle
H008 CSR      .BAS   6 Computerspektroskopie
H009 VIERER   .BAS  10 Vier gewinnt
H010 ZONEX    .BAS   9 Vierfarbenfeld erraten
H011 HIRN     .BAS   8 Hirnzwirn
H012 LANDER   .BAS   8 Landung auch auf Planeten
H013 WUMPUS   .BAS  24 Jag den Wumpus
H014 WUMPUS2 .BAS  21 Noch mehr Hoehlen
H015 AWARI    .BAS   7 Mini-Kalah
H016 KRAFTWER.BAS  36 Energiebetrieb
H017 SCHIFF   .BAS  33 Schiffe versenken
H018 KKW      .BAS  42 Kernkraftwerk
H019 KKW      .DAT   1 Bester Spieler von H018
H020 LIMONADE.BAS  26 Verdiene viel
H021 TRUCK    .BAS  56 Fernfahrer in den USA
H022 TRUCK    .DAT   1 Bester Spieler von H021
H023 WALD     .BAS  51 Such den Schatz
H024 WALD     .DAT   1 Bester Spieler von H023
H025 BORNES   .BAS  43 Kartenspiel Mille Bornes
H026 BUNNY    .BAS   5 Druckt Hasen
H027 KURVE    .BAS   2   "    Kurve
H028 LOVE     .BAS   1   "
H029 BANNER   .BAS  10   "
H030 ZEIT     .BAS   2 Zeitzeichen
H031 WAHL     .BAS  47 Wahl des US-Praesidenten
H032 VERKEHR .BAS  90 Verkehrsminister
H033 FEUER    .BAS  10 Waldbrand bekaempfen
H034 FLUCHT   .BAS   5 Weich dem Monster aus
H035 INSEL    .BAS   9 Schatzsuche auf einer Insel
H036 QUEST    .BAS  32 Schatzsuche/M.Bach
```

.....

Anweisungen zu Programmen

```
I001 STAR     .DOC  13 Fuer D009
I002 BORNES   .DOC  11 Fuer H025
I003 WAHL     .DOC   5 Fuer G012 und H031
I004 SORT     .DOC  12 Fuer C006
I005 FC       .DOC   8 Fuer C004
I006 DISASM   .DOC  14 Fuer C005
I007 PICPATCH.DOC   8 Fuer C008
I008 SUBMIT   .DOC   9 Fuer C007
I009 DIAGNOSE.DOC   3 Fuer B003
I010 ASMUT    .DOC   9 Fuer E014
I011 RANDAUSG.DOC   5 Fuer E015
```

.....

Hilfsprogramme fuer CLD-BASIC

```
J001 SET      .BAU   5 Graphik SET,RESET,POINT
J002 HEXLIST .BAS   4 Hexausdruck eines Files
J003 DATEI    .BAS  15 Anlegen einer Datei
J004 SINGLE   .BAS   2 Einzeleingabe in File
J005 KBD      .BAS   1 Umwandlung in Hex
J006 UMCODE   .BAS   5 Umcodieren
```

Programmiersprachen

```
K001 FORTH    .ASM  43 nach G.Kreidl/Journal
```

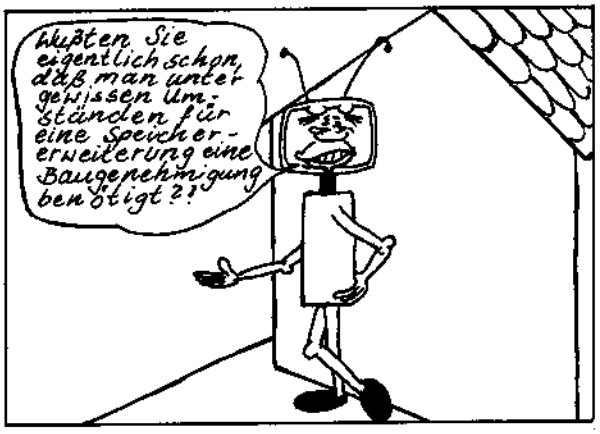
.....

Zur Bibliothek haben beigetragen

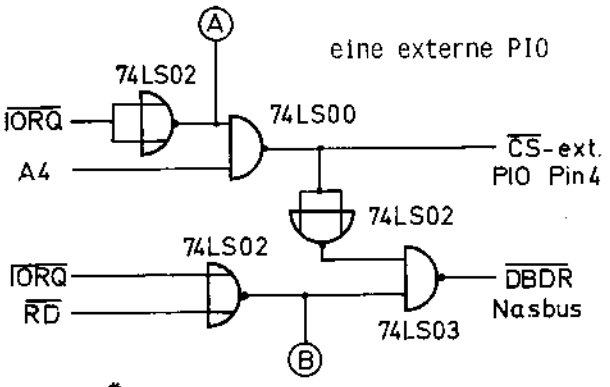
G.Baier	GB
H.Emmelmann	HE
G.Endert	GE
U.Kafka	UK
C.Lampson	CL
W.Mayer-Guerr	WMG
M.Reimer	MR

Die Mitarbeiter dieser Ausgabe waren:

Heinrich Auge, [redacted], [redacted] Künzelsau, Tel. [redacted]; Michael Bach, [redacted], [redacted] Stegen, Tel. [redacted]; Günter Böhm, siehe Impressum; Otto Fößel, [redacted], [redacted] Erlangen, Tel. [redacted]; Uwe Fricke, [redacted], [redacted] Troisdorf [redacted]; Eberhard Horch, [redacted], [redacted] Hannover [redacted]; Günter Kreidl, siehe Impressum; Rüdiger Maurer, [redacted], [redacted] Taunusstein; Klaus Mombaur, [redacted], [redacted] Nürnberg [redacted]; Ottmar Schweizer, Adresse leider verschollen - bitte bei der Redaktion melden; Josef Zeller, siehe Impressum; und wie immer perfekt getippt: Gabi Böhm.-



PIO Erweiterung für NASCOM 1



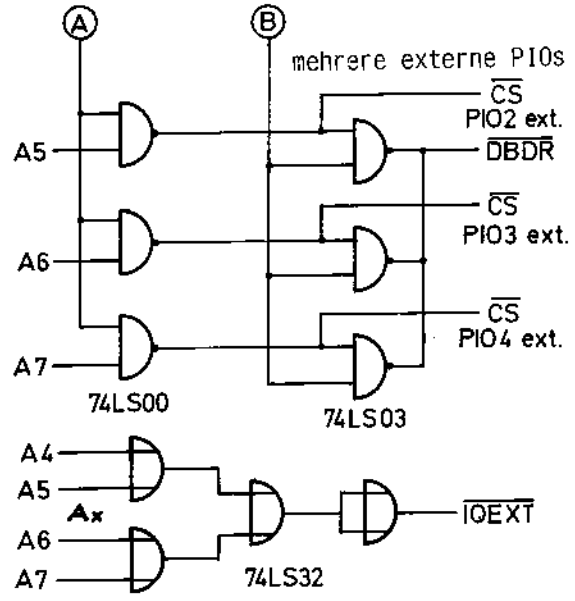
Folgende Leitungen der PIO sind noch mit dem NASBUS zu verbinden:

PIN	Bezeichnung
23	INT
36	IORQ
37	M1
35	RD
25	Ø
19	DØ
20	D1
1	D2
40	D3
39	D4
38	D5
3	D6
2	D7
4	AØ
5	A1
26	+5V
24	+5V
11	Masse

Die PIO hat dann folgende Steuerworte:

	PORT 1A	PORT 1B
CONT	12	13
DATA	10	11 (HEX)

Achtung: IC 46a PIN 2 auf dem NASCOM 1 - Board ist direkt an A2 der CPU anzuschließen, und die Brücke IO auf EXT zu schalten.



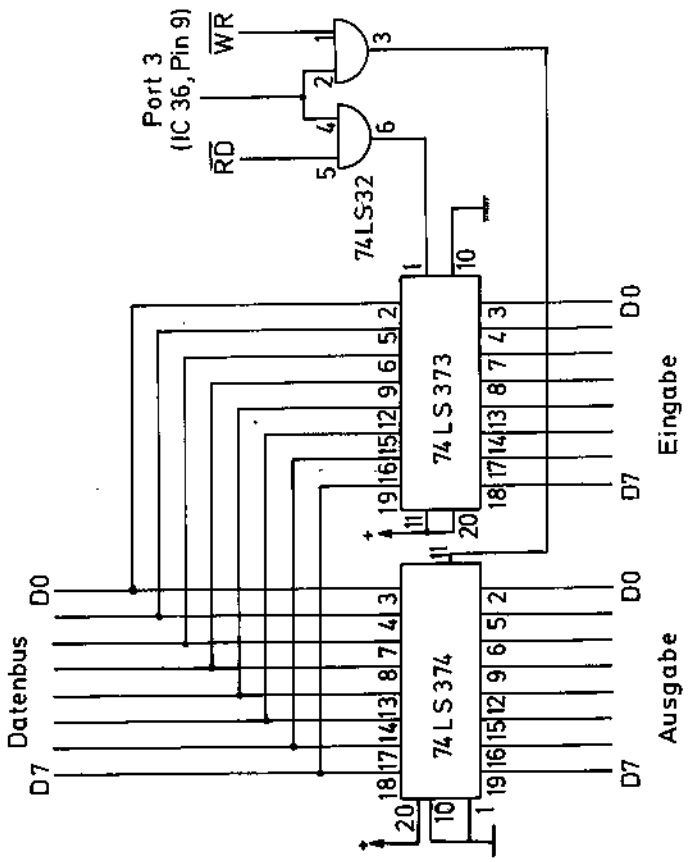
Steuerworte für die PIOs

Ax — Port	2A	2B	3A	3B	4A	4B
	5	5	6	6	7	7
CONTROL	22	23	42	43	82	83
DATA	20	21	40	41	80	81

Ich habe meine Schaltung frei verdrahtet, sodaß kein Lay Out existiert.

RÜDIGER MAURER

Mit drei ICs (ca.8.-DM) läßt sich der NASCOM 1 schon in der Grundversion um einen Eingabe- und einen Ausgabeport erweitern. Auf der NASCOM-Platine wird Port 3 zwar dekodiert (IC 36, Pin9), aber nicht benutzt. Durch zwei 8-Bit-Flipflops mit Tri-State-Ausgängen können zwei "Einweg"-Ports angeschlossen werden. Die Schaltung funktioniert bei mir schon seit ein paar Monaten problemlos. Es sind ein Drucker und eine zusätzliche Tastatur angeschlossen.



UWE FRICKE

Seite(n) für Einsteiger von Günter Böhm

Hier also der versprochene Beitrag über die Ladebefehle. (Wir wollten ja streng nach dem Z8Ø Manual vorgehen, aber Günter Kreidl hat mit der BCD-Serie etwas vorgegriffen). Zunächst ein Tip zur Arbeit im Assembler. Obwohl ich den ZEAP 2.Ø besitze (und vorher mit ASM Assembler gearbeitet habe), skizziere ich Programme immer zunächst auf dem Papier. Man überblickt dann größere Progamm-teile schneller und hat nicht nur 15 Bildschirmzeilen vor sich. Das Umschreiben der mnemonischen Assemblerbefehle in den Object-Code lasse ich allerdings den Rechner machen (obwohl mir das Assemblieren per Hand nichts geschadet hat. Im Gegenteil: nach häufigem Aufsuchen des Object-Codes in den Tabellen gewinnt man einen Blick für das Lesen von Hexdumps, so daß man z.B. schnell zwischen Befehlen, ASCII-Strings und Tabellen unterscheiden kann). Nun aber zur Praxis!

Ein Programm besteht eigentlich nur aus Daten, die zwischen verschiedenen Registern des Prozessors ausgetauscht, manipuliert, in verschiedenen Speichern abgelegt oder durch Ports ausgegeben werden. So ist das Laden von Registern und Speichern einer der wichtigsten Aufgaben eines Programms.

Betrachten wir uns die Matrix der 8 Bit-Ladebefehle. (In meinem MOSTEK Manual auf Seite 29). Ich habe sie allerdings wie alle anderen Tabellen fotokopiert und in Klarsichthüllen aufbewahrt, so daß ich immer die möglichen Befehle vor mir habe). In der linken Spalte stehen die Zielregister (Destination), die geladen werden sollen; in den Zeilen stehen die Quellregister (Source), aus denen geladen wird. Nun muß man nur noch wissen, daß ein Register ohne Klammer direkt geladen wird, ein Register mit Klammer aber auf die Adresse hinweist, die geladen werden soll. (Keine Angst, dies wird am Beispiel verdeutlicht).

Soll Register A mit dem Wert von Register B geladen werden (im Assembler heißt das LD A, B), so suche ich in der Tabelle A als Ziel und B als Quelle. An der Schnittstelle findet man den Code für den Rechner 78. Ich kann aber auch A direkt mit einer Zahl n laden. (n steht ganz oben rechts). Dann heißt der Befehl im Maschinencode 3E n.

Probieren wir das einmal praktisch aus. Mit dem MODIFY Befehl M C8Ø ENTER nehmen wir uns die Adresse ØC8Ø vor. Der Rechner zeigt nun auf dem Bildschirm, welchen Inhalt diese Adresse hat. Wir geben ihr nun den Inhalt Ø6, in dem wir diese Hexzahl und anschließend ENTER eintippen. Nun wird auf dem Schirm die nächste Adresse angezeigt. Die belegen wir mit z.B. FF + ENTER. Damit haben wir den Beginn unseres Programmes bereits festgelegt. Ø6 FF bedeutet LD B,FF hex (Ziel B wird mit Zahl n = FF geladen). Unser nächster Befehl ist 78 (Sie erinnern sich: LD A,B). Damit wir unser Programm auch testen können, geben wir zuletzt E7 ein. Dieser Befehl bewirkt einen Sprung in das Betriebssystem und zeigt alle Register an. Wir schließen das Laden des Programmes mit Punkt und ENTER ab. Wenn das Programm nun mit E C8Ø (execute = ausführen) gestartet wird, zeigt es augenblicklich die Register auf dem Bildschirm an.

```
Registeranzeige bei NAS-SYS 3
1ØØØ XXXX ØC83 XXXX FFXX XXXX XXXX XXXX
SP        PC        AF        HL
XXXX XXXX FFXX XXXX XX XXXX XXXX
DE        BC          IX   IY
```

Register A und B enthalten beide FF; denn unser Programm lautete

```
LD B,FF
LD A,B
RST 2ØH (Restart 2ØHex = E7 = Breakpoint)
```

Sie können obiges Programm auch mit den anderen Registern durchspielen. Die Registeranzeige gibt Ihnen jeweils das Ergebnis an. Mit dem S Befehl (Single Step = Einzelschritt) können Sie auch jede Programmzeile einzeln verfolgen. Achten Sie bei folgendem Programm darauf, wie sich die einzelnen Register verändern, und suchen Sie die Befehle in der Tabelle auf.

```
0C80           0001        ORG  #C80
0C80 3E00      0010        LD   A,0
0C82 67        0020        LD   H,A
0C83 6F        0030        LD . L,A
0C84 57        0040        LD   D,A
0C85 5F        0050        LD   E,A
0C86 47        0060        LD   B,A
0C87 4F        0070        LD   C,A
0C88 3EFF      0080        LD   A,#FF
0C8A C3820C    0090        JP   #C82
0C8D E7        0100        RST  #20
```

Der Pseudocode des Assemblers ORG #C8Ø (#↑= ↑=Hex) bedeutet, daß das Programm bei C8Ø beginnt (Origin = Ursprung). In Zeile 9Ø ist ein Sprung (JP = Jump) nach der Adresse C82 eingetragen. Unter normalen Umständen erreicht das Programm also nie die Adresse C8D. Es ist nur für den Single Step Betrieb vorgesehen. Der Breakpoint E7 dient nur der Sicherheit. Wenn Sie das Programm mit E C8Ø starten, wird es bis zum nächsten Stromausfall die Register immer wieder mit FF laden.

In gleicher Weise wird mit den 16 Bit - Ladebefehlen verfahren. So lädt der Befehl Ø1FFFF das Registerpaar BC mit der Hexzahl FFFF. Schreiben Sie sich selbst ein Programm, das die Register BC, DE und HL mit AABB lädt. (Sie werden feststellen, daß man beim Laden immer erst das niederwertige Byte eingeben muß).
Nun wollen wir aber endlich zu den eingeklammerten Registern kommen. Diese Art des Ladens nennt sich indirekte Adressierung; d. h. nicht das Register selbst, sondern die Adresse, die es enthält, wird geladen. Verfolgen Sie dieses Programm:

```
LD HL, #ØDØØ
LD (HL), #FF
RST 2ØH
```

Das Registerpaar HL wird mit der Adresse DØØ geladen. Daraufhin lädt man diese Adresse indirekt (Sie steht ja in HL) mit dem Wert FF. Sie können das nach dem Programmende durch M DØØ (Modify- Befehl) selbst feststellen. Die Adresse DØØ muß FF enthalten. Den gleichen Effekt sollte der Befehl LD (#DØØ), #FF bewirken, aber er existiert weder in der Tabelle noch im Z8Ø Chip. Man könnte die Adresse allerdings auf dem Umweg über das A Register laden.

```
LD A,#FF
LD (#DØØ),A
```

Glücklicherweise ist unser Bildschirmspeicher genau wie ein RAM-Speicher zu behandeln, und wir können direkt auf dem Schirm sehen, was geladen wird, um unsere Theorien praktisch zu erproben. Allerdings werden auf dem Schirm nicht die Hexzahlen, sondern die zugehörigen ASCII-Codes abgebildet. (Im Construction Article des NASCOM finden Sie eine Übersicht über die Bildschirmadressen und die ASCII-Codes; für NASCOM 1 Seite 32 und 33). Spielen Sie das Laden von Bildschirmadressen mit verschiedenen Registern durch.

```
0C80          0010      ORG  #C80
0C80 3EB7     0020      LD   A,#B7 ;GRAFIK MÄNNCHEN
0C82 210A08.  0030      LD   HL,#080A      ODER 7
0C85 11E209   0040      LD   DE,#09E2
0C88 01B90B   0050      LD   BC,#0BB9
0C8B 77       0060      LD   (HL),A
0C8C 12       0070      LD   (DE),A
0C8D 02       0080      LD   (BC),A
0C8E 76       0090      HALT
```

Der HALT Befehl (76) bringt den Prozessor in einen Wartezustand. Er kann ihn nur durch RESET verlassen.
Außer den 3 Registerpaaren und dem Akku (Register A) finden Sie noch die Indexregister IX und IY in der Tabelle. Man kann sie mit einer Adresse laden und dann Werte an Adressen übergeben, die durch die Addition des Index d zur Adresse ermittelt werden. Die Ausführung des folgenden Programms macht den Vorgang deutlich.

```
0C80           0010      ORG  #C80
0C80 DD210A08  0020      LD   IX,#080A
0C84 3EB7      0030      LD   A,#B7
0C86 DD360030  0040      LD   (IX+0),"0
0C8A DD360131  0050      LD   (IX+1),"1
0C8E DD360232  0070      LD   (IX+2),"2
0C92 DD360333  0080      LD   (IX+3),"3
0C96 DD362F34  0090      LD   (IX+47),"4
0C9A 76        0100      HALT
```

Man kann die Indexregister aber auch wie das HL Register behandeln, d.h. nicht nur als Registerpaar sondern auch als Einzelregister. Dazu setzt man vor den entsprechenden Code für H oder L einfach DD (entspricht IX) oder FD (entspricht IY). Führen Sie folgendes Programm in Einzelschritten aus, und vergleichen Sie die Registeranzeigen von IX und IY.

```
LD IX high Byte,"H  DD 26 48
LD IX low  Byte,"L  DD 2E 4C
LD IY high Byte,"H  FD 26 48
LD IY low  Byte,"L  FD 2E 4C
```

Diese Manipulation ist nicht im Manual vorgesehen, auch der Assembler versteht sie nicht; aber sie läuft auf den meisten Z8Ø Chips und leistet gute Dienste, wenn man zusätzliche Register benötigt.
Eine reguläre Art, zusätzliche Register zu gewinnen, ist die Benutzung des zweiten Registersatzes (Alternate Register Set). Diese Register werden einfach durch den

Befehl EXX (exchange = austauschen), oder D9 in Maschinensprache, ausgetauscht und wie der Hauptregistersatz verwendet. Beim Ausführen des folgenden Programms stellen Sie fest, daß tatsächlich die Inhalte der Register ausgetauscht wurden - bis auf das A Register. Dessen Austausch wird durch den Befehl EX AF,AF' (Ø8) bewirkt.

```
0C80            0010        ORG  #C80
                0020 ;NORMALER REGISTERSATZ
0C80 3E31       0030        LD   A,"1
0C82 1632       0040        LD   D,"2
0C84 1E33       0045        LD   E,"3
0C86 0634       0050        LD   B,"4
0C88 0E35       0055        LD   C,"5
0C8A 2636       0060        LD   H,"6
0C8C 2E37       0065        LD   L,"7
                0070 ;ERSTES AUSTAUSCHEN
0C8E D9         0080        EXX
0C8F 3E38       0090        LD   A,"8
0C91 1639       0100        LD   D,"9
0C93 1E41       0105        LD   E,"A
0C95 0642       0110        LD   B,"B
0C97 0E43       0115        LD   C,"C
0C99 2644       0120        LD   H,"D
0C9B 2E45       0125        LD   L,"E
                0130 ;ANZEIGE DER REGISTER
0C9D DD210A08 0140          LD   IX,#080A
0CA1 DD7500     0150        LD   (IX),L
0CA4 DD7401     0160        LD   (IX+1),H
0CA7 DD7102     0170        LD   (IX+2),C
0CAA DD7003     0180        LD   (IX+3),B
0CAD DD7304     0190        LD   (IX+4),E
0CB0 DD7205     0200        LD   (IX+5),D
0CB3 DD7706     0210        LD   (IX+6),A
                0220 ;ERNEUTER TAUSCH
0CB6 D9         0230        EXX
0CB7 DD7507     0240        LD   (IX+7),L
0CBA DD7408     0250        LD   (IX+8),H
0CBD DD7109     0260        LD   (IX+9),C
0CC0 DD700A     0270        LD   (IX+10),B
0CC3 DD730B     0280        LD   (IX+11),E
0CC6 DD720C     0290        LD   (IX+12),D
0CC9 DD770D     0300        LD   (IX+13),A
0CCC 76         0310        HALT
```

Eine weitere Tauschmöglichkeit zwischen HL und HD kann man durch EX DE,HL (EB) erreichen.
Diese Ausführungen sind nun schon zu solchen Ausmaßen angewachsen, daß sie den Rahmen des Heftes fast überschreiten. Testen Sie die besprochenen Möglichkeiten durch und machen Sie sich mit den Befehlen fest vertraut. Im nächsten Heft folgt dann der Abschluß der Ladebefehle (I und R Register, PUSH - und POP - Befehle) und möglicherweise die Beantwortung Ihrer Fragen zu diesem Thema.

Laufschrift

von Eberhard Horch

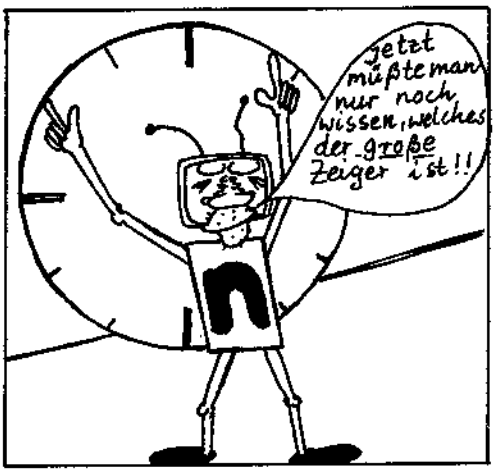
```
10 REM       * LAUFSCHRIFTPROGRAMM  *
11 REM
12 REM * Eberhard Horch,[redacted] Hannover [redacted]     *
13 REM *                              [redacted]  *
14 REM
15 REM
20 CLS:CLEAR1000 :B$=" "
30 READ A$
40 A=LEN(A$):B=47-A
50 FOR C=1 TO B
60 D$=D$+B$:NEXT C
70 C$=A$+D$:CLS
80 FOR X=47 TO 1 STEP-1
90 D=ABS(X-48)
100 SCREENX,8:PRINTMID$(C$,1,D)
110 FOR M=1TO33:NEXTM:NEXT X
120 FOR Y=A TO 1 STEP-1
130 E=ABS(Y-(A+2))
140 SCREEN1,8:PRINTMID$(C$,E,A)
150 FOR M=1TO50:NEXTM:NEXT Y
160 L=L+1
170 IFL=5THEN PRINT"Danke für Ihre Geduld !!!":
171 IFL=5THEN END
180 GOTO30
190 REM * DATA *
200 DATA"Dies ist ein Probetext."
210 DATA"Hiermit wird die Funktion ....."
220 DATA"...des Laufschrift Programms getestet"
230 DATA"Der Text kann durch weitere DATA "
240 DATA"Statements verlaengert werden."
Ok
```

Durch Ändern der Zeile 170
170 IFL=5THEN20
beginnt das Programm nach Durchlaufen des Textes von neuem. In der vorliegenden Form bricht es nach dem Text (hier 5 Zeilen) ab.
Beim Verlängern des Textes muß natürlich die Länge L in Zeile 170/171 geändert werden.

UHR mit Großschrift

von Eberhard Horch

```
10 REM        ***      UHR      ***
20 REM * Modifiziert nach CBM Unterlagen *
30 REM
40 REM * Eberhard Horch, Hannover *
50 REM
60 REM               * 5.Jan. 1982 *
70 REM -----------------------------------
80 CLS:CLEAR 1000:A=0
90 Z=239:REM * Z=Zeitbestimmender Faktor *
100 REM * Muss eventuell angepasst werden *
110 DIM A$(60)
120 FOR I=0 TO 59
130 READ A$(I):NEXT I
140 B$(0)="       ":B$(1)="       "
150 B$(2)="       ":B$(3)="       "
160 B$(4)="       ":B$(5)="       "
170 C$=" "
180 INPUT"Stunde,Minute";H,M
190 IF H>24 OR M>59 THEN 1340
200 CLS
210 GOSUB 1070
220 FOR S=0 TO 59
230 FOR N=1 TO Z:NEXT N:REM * Zeitschleife *
240 GOSUB 1220
250 IFS<59 GOTO 280
260 IF S=59 THEN M=M+1
270 GOSUB 1070
280 IF M=60 THEN 320
290 IF H=24 THEN 330
300 IF S=59 THEN 220
310 NEXT S
320 M=0:GOTO 290
330 H=0:M=0:S=0:GOTO 220
340 REM * DATA fuer Zifferndarstellung *
350 DATA "JJJJ "
360 DATA "J  J "
370 DATA "J  J "
380 DATA "J  J "
390 DATA "J  J "
400 DATA "JJJJ "
410 REM
420 DATA" JJ  "
430 DATA"  J  "
440 DATA"  J  "
450 DATA"  J  "
460 DATA"  J  "
470 DATA" JJJ "
480 REM
490 DATA"JJJJ "
500 DATA"   J "
510 DATA"   J "
520 DATA"JJJJ "
530 DATA"J    "
540 DATA"JJJJ "
550 REM
560 DATA"JJJJ "
570 DATA"   J "
580 DATA"   J "
590 DATA"  JJ "
600 DATA"   J "
610 DATA"JJJJ "
620 REM
630 DATA"J    "
640 DATA"J    "
650 DATA"J J  "
660 DATA"JJJJ "
670 DATA"  J  "
680 DATA"  J  "
690 REM
700 DATA"JJJJ "
710 DATA"J    "
720 DATA"J    "
730 DATA"JJJJ "
740 DATA"   J "
750 DATA"JJJJ "
760 REM
770 DATA"JJJJ "
780 DATA"J    "
790 DATA"J    "
800 DATA"JJJJ "
810 DATA"J  J "
820 DATA"JJJJ "
830 REM
840 DATA"JJJJ "
850 DATA"   J "
860 DATA"   J "
870 DATA"  JJ "
880 DATA"   J "
890 DATA"   J "
900 REM
910 DATA"JJJJ "
920 DATA"J  J "
930 DATA"J  J "
940 DATA"JJJJ "
950 DATA"J  J "
960 DATA"JJJJ "
970 REM
980 DATA"JJJJ "
990 DATA"J  J "
1000 DATA"J  J "
1010 DATA"JJJJ "
1020 DATA"   J "
1030 DATA"JJJJ "
1040 REM
1050 END
1060 REM * Zeitumrechnung fuer Darst
ellung *
1070 T=M:IF T=60 THEN T=00
1080 IF T=00 THEN H=H+1
1090 IF H=24 THEN H=00
1100 T1=INT(T/10):T0=T-T1*10
1110 H1=INT(H/10):H0=H-H1*10
1120 T1=T1*6:T0=T0*6
1130 H1=H1*6:H0=H0*6
1140 FOR I=0 TO 5
1150 Z$(I)=C$+A$(H1+I)+A$(H0+I)+B$(I
)+A$(T1+I)+A$(T0+I)
1160 NEXT I
1170 CLS:PRINT:PRINT
1180 FORI=0TO5
1190 PRINTTAB(8)Z$(I):NEXT I
1200 RETURN
1210 REM * Darstellung Sekunde und P
endel *
1220 SCREEN 21,8:PRINT S
1230 IF A=1 THEN 1290
1240 SCREEN 20,3:PRINT" *    "
1250 SCREEN 20,4:PRINT"  *   "
1260 SCREEN 20,5:PRINT"   *  "
1270 SCREEN 20,6:PRINT"    * "
1280 SCREEN 20,7:PRINT"     *":A=1:R
ETURN
1290 SCREEN 20,3:PRINT"    * "
1300 SCREEN 20,4:PRINT"   *  "
1310 SCREEN 20,5:PRINT"  *   "
1320 SCREEN 20,6:PRINT" *    "
1330 SCREEN 20,7:PRINT"*     ":A=0:R
ETURN
1340 PRINT"Bitte richtige Uhrzeit ei
ngeben !"
1350 GOTO 180
1360 REM
1370 REM * Ende Uhrenprogramm *
```



Grafikroutinen in Assembler

von Michael Bach

Nachdem ich die sehr empfehlenswerte 2732-Grafik von B.Ploss in meinen Nascom 1 eingebaut hatte, stellte ich fest, daß in BASIC zwar die Grafik schön ansteuerbar ist, für schnelle Bewegungen aber zu langsam. Also habe ich Assembler- Grafik- Routinen geschrieben, die ähnliches tun wie die SET, RESET & POINT in BASIC. Näheres geht aus dem beiliegenden Listing hervor. Eine Anwendung findet sich in meinem Spiel "Doppelwurm". Es wäre schön, wenn jemand mal als nächstes VEKTOR programmierte, wofür ich folgende Parameterübergabe vorschlage: Endkoordinate in HL, Anfangskoordinate ist der Endpunkt beim vorherigen Aufruf. Carry Set bedeutet Vektor hellzeichnen, Carry gelöscht (durch OR A oder so) bedeutet dunkler Vektor (um einen neuen Vektor anzufahren). Diese Konvention im Gegensatz zur Übergabe der Anfangs- und Endkoordinate erleichtert das Zeichnen geschlossener Linien. Für eine einzelne Linie muß man VEKTOR allerdings zweimal aufrufen.

ZEAP Z80 Assembler - Source Listing

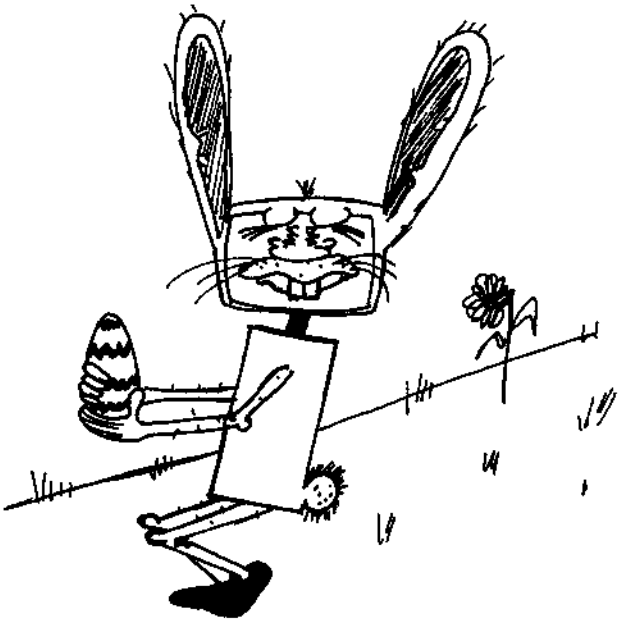
```
                0010 ;*** GRAFIK-ELEMENTE ***   3.3.82
                0020 ;
                0030 ;Michael Bach, [redacted],
                0040 ;[redacted] Stegen; Tel. [redacted].
                0050 ;
                0060 ;Elementare Grafik-Routinen für die
                0070 ;Nascom-2-Klötzchen-Grafik.
                0080 ;
                0090 ;3 Funktionen (wie im Basic):
                0100 ;SET: Setzen eines Punktes
                0110 ;RESET: Löschen eines Punktes
                0120 ;POINT: Testen eines Punktes
                0130 ;Bereichsüberschreitungen werden abgefange
                0140 ;
                0150 ;Mit einem simplen Demoprogramm vorne dran
                0160 ;
2E9F 0038       0170 RDEL   EQU  38H
2E9F 0030       0180 ROUT   EQU  30H
2E9F 005B       0190 MRET   EQU  5BH
2E9F 0061       0200 KBD    EQU  61H
2E9F 000C       0210 FF     EQU  0CH
2E9F 080A       0220 VL1    EQU  080AH
                0230 ;
1000            0240        ORG  1000H
1000            0250        ENT
1000 3E0C       0260        LD   A,FF
1002 F7         0270        RST  ROUT;Schirm löschen
                0280 ;Rahmen malen
                0290 ;------------
                0300 ;linke Kante
1003 210000     0310        LD   HL,0;Oben links: Y=0, X=0
1006 062C       0320        LD   B,44;44 Punkte
1008 CD4F10     0330 BGL    CALL SET
100B 24         0340        INC  H;abwärts
100C 10FA       0350        DJNZ BGL
                0360 ;unten
100E 065F       0370        LD   B,95
1010 CD4F10     0380 BGU    CALL SET
1013 2C         0390        INC  L
1014 10FA       0400        DJNZ BGU
                0410 ;rechts
1016 062C       0420        LD   B,44
1018 CD4F10     0430 BGR    CALL SET
101B 25         0440        DEC  H
101C 10FA       0450        DJNZ BGR
                0460 ;oben
101E 065F       0470        LD   B,95
1020 CD4F10     0480 BGO    CALL SET
1023 2D         0490        DEC  L
1024 10FA       0500        DJNZ BGO
```

```
                0510 ;
                0520 ;Ping-Pong-Bewegung
                0530 ;-----------------
1026 210118     0540        LD   HL,1801H;Anfangspos: X=1,Y=24
1029 D724       0550 PP1    RCAL SET;Punkt an
102B D716       0560        RCAL VERZ
102D D736       0570        RCAL RESET;Punkt aus
102F 2C         0580        INC  L;eins nach rechts
1030 D74D       0590        RCAL POINT;am Rand (dort ist Grafik an)?
1032 30F5       0600        JR   NC,PP1
1034 2D         0610        DEC  L;also wieder zurück
1035  D718       0620 PP2     RCAL SET
1037 D70A       0630        RCAL VERZ
1039 D72A       0640        RCAL RESET
103B 2D         0650        DEC  L;eins nach links
103C D741       0660        RCAL POINT;am Rand?
103E 30F5       0670        JR   NC,PP2
1040 2C         0680        INC  L
1041 18E6       0690        JR   PP1
                0700 ;
                0710 ;Verzögerung & Abbruch über Tastatur
1043 0605       0720 VERZ   LD   B,5;Geschwindigkeit
1045 FF         0730 VZ1    RST  RDEL
1046 10FD       0740        DJNZ VZ1
1048 E5         0750        PUSH HL
1049 DF61       0760        SCAL KBD
104B E1         0770        POP  HL
104C D0         0780        RET  NC
104D DF5B       0790        SCAL MRET
                0800 ;
                0810 ;
                0820 ;  ********************
                0830 ;*** GRAFIK-ROUTINEN *** 06.01.82
                0840 ;  ********************
                0850 ;
                0860 ;SET: Setzen eines Grafik-Punktes
                0870 ;-------------------------------
                0880 ;X(0-95) in L; Y(0-47) in H
104F F5         0890 SET    PUSH AF
1050 C5         0900        PUSH BC
1051 D5         0910        PUSH DE
1052 E5         0920        PUSH HL
1053 D739       0930        RCAL GRAFM
1055 47         0940        LD   B,A
1056 7E         0950        LD   A,(HL)
1057 E6C0       0960        AND  0C0H;iss schon Grafik?
1059 FEC0       0970        CP   0C0H
105B 78         0980        LD   A,B
105C 2001       0990        JR   NZ,ST5
105E B6         1000        OR   (HL)
105F 77         1010 ST5    LD   (HL),A
1060 E1         1020        POP  HL
1061 D1         1030        POP  DE
1062 C1         1040        POP  BC
1063 F1         1050        POP  AF
1064 C9         1060        RET
                1070 ;
                1080 ;RESET: Löschen eines Grafik-Punktes
                1090 ;---------------------------------
1065 F5         1100 RESET  PUSH AF
1066 C5         1110        PUSH BC
1067 D5         1120        PUSH DE
1068 E5         1130        PUSH HL
1069 D723       1140        RCAL GRAFM
106B 47         1150        LD   B,A
106C 7E         1160        LD   A,(HL)
106D 4F         1170        LD   C,A
106E E6C0       1180        AND  0C0H;iss schon Grafik?
1070 FEC0       1190        CP   0C0H
1072 78         1200        LD   A,B
1073 2005       1210        JR   NZ,RT5
1075 2F         1220        CPL
1076 F6C0       1230        OR   0C0H
1078 A1         1240        AND  C
1079 77         1250        LD   (HL),A
107A E1         1260 RT5    POP  HL
107B D1         1270        POP  DE
107C C1         1280        POP  BC
107D F1         1290        POP  AF
107E C9         1300        RET
```

```
                1320 ;POINT: Testen eines Grafik-Punktes
                1330 ;-----------------------------------
                1340 ;X(Ø-95) in L; Y(Ø-47) in H
                1350 ;Carry gesetzt wenn dort Grafik-Punkt
107F C5         1360 POINT  PUSH BC
1080 D5         1370        PUSH DE
1081 E5         1380        PUSH HL
1082 D70A       1390        RCAL GRAFM
1084 47         1400        LD   B,A
1085 7E         1410        LD   A,(HL)
1086 2F         1420        CPL
1087 A0         1430        AND  B
1088 E1         1440        POP  HL
1089 D1         1450        POP  DE
108A C1         1460        POP  BC
108B C0         1470        RET  NZ
108C 37         1480        SCF
108D C9         1490        RET
                1500 ;
                1510 ;Berechnung von Position & Bit-Maske
108E 7C         1520 GRAFM  LD   A,H;D:=H/3
108F 0E03       1530        LD   C,3
1091 16FF       1540        LD   D,-1
1093 14         1550 DIV    INC  D
1094 91         1560        SUB  C
1095 30FC       1570        JR   NC,DIV
1097 81         1580        ADD  A,C
1098 47         1590        LD   B,A;Rest=Y-Maske
1099 7D         1600        LD   A,L
109A 0E00       1610        LD   C,0
109C CB3F       1620        SRL  A
109E 3001       1630        JR   NC,ST2
10A0 0C         1640        INC  C;X-Maske
10A1 5F         1650 ST2    LD   E,A;E:=L/2
10A2 D713       1660        RCAL POS
10A4 3E01       1670        LD   A,1
10A6 04         1680        INC  B
10A7 87         1690 ST4    ADD  A,A
10A8 10FD       1700        DJNZ ST4
10AA CB41       1710        BIT  0,C
10AC 2804       1720        JR   Z,ST3
10AE 87         1730        ADD  A,A
10AF 87         1740        ADD  A,A
10B0 1802       1750        JR   ST6
10B2 CB3F       1760 ST3    SRL  A
10B4 F6C0       1770 ST6    OR   0C0H
10B6 C9         1780        RET  ;Bit-Maske in A, Adresse in HL
                1790 ;
                1800 ;UP POS: Berechnung der Bildschirmpos.
                1810 ;        aus E(=X,0-47) und D(=Y,0-15)
                1820 ;        Bereichsüberschreitung gesichert,
                1830 ;        Ergebnis in HL.
                1840 ;   Diese Routine ist ähnlich wie SCREEN
                1850 ;   in Basic, setzt aber nicht den Cursor.
10B7 2600       1860 POS    LD   H,0
10B9 7A         1870        LD   A,D
10BA E60F       1880        AND  0FH
10BC 6F         1890        LD   L,A
10BD C5         1900        PUSH BC
10BE 0606       1910        LD   B,6
10C0 29         1920 PØØ    ADD  HL,HL
10C1 10FD       1930        DJNZ PØØ
10C3 C1         1940        POP  BC
10C4 1600       1950        LD   D,0
10C6 7B         1960        LD   A,E
10C7 FE30       1970        CP   48
10C9 3802       1980        JR   C,PO1
10CB 1E2F       1990        LD   E,47
10CD 19         2000 PO1    ADD  HL,DE
10CE 110A08     2010        LD   DE,VL1
10D1 19         2020        ADD  HL,DE
10D2 C9         2030        RET
                2040 ;
                2050 ; ENDE
```

```
1000 3E 0C F7 21 00 00 06 2C|A4
1008 CD 4F 10 24 10 FA 06 5F|D7
1010 CD 4F 10 2C 10 FA 06 2C|B4
1018 CD 4F 10 25 10 FA 06 5F|E8
1020 CD 4F 10 2D 10 FA 21 01|B5
1028 18 D7 24 D7 16 D7 36 2C|71
1030 D7 4D 30 F5 2D D7 18 D7|7C
1038 0A D7 2A 2D D7 41 30 F5|BD
1040 2C 18 E6 06 05 FF 10 FD|91
1048 E5 DF 61 E1 D0 DF 5B F5|5D
1050 C5 D5 E5 D7 39 47 7E E6|9A
1058 C0 FE C0 78 20 01 B6 77|AC
1060 E1 D1 C1 F1 C9 F5 C5 D5|2C
1068 E5 D7 23 47 7E 4F E6 C0|11
1070 FE C0 78 20 05 2F F6 C0|C0
1078 A1 77 E1 D1 C1 F1 C9 C5|92
1080 D5 E5 D7 0A 47 7E 2F A0|BF
1088 E1 D1 C1 C0 37 C9 7C 0E|55
1090 03 16 FF 14 91 30 FC 81|0A
1098 47 7D 0E 00 CB 3F 30 01|B5
10A0 0C 5F D7 13 3E 01 04 87|CF
10A8 10 FD CB 41 28 04 87 87|0B
10B0 18 02 CB 3F F6 C0 C9 26|89
10B8 00 7A E6 0F 6F C5 06 06|77
10C0 29 10 FD C1 16 00 7B FE|56
10C8 30 38 02 1E 2F 19 11 0A|C3
10D0 08 19 C9 00 00 00 00 00|CA
```

nascompl



Hallo liebe Leser,
die Verzögerung im Versand des Journals hat mich leider darum betrogen, Ihnen einige philosophische Gedanken zu Fasching zu erörtern. Aber glücklicherweise naht schon ein neuer Höhepunkt im Jahresablauf eines Mikros: das Osterfest.
Zu Weihnachten haben wir versucht, Ihrem Computer die Tage zu verschönen; diesmal soll an den Programmierer gedacht werden.
Erinnern Sie sich noch an die Zeit, da Sie als Kind in der freien Natur nach Ostereiern suchten? War es nicht ein wichtiges Ereignis, das auch Ihr weiteres Leben bestimmte, wenn Sie erfolgreich triumphierend die süße oder hartgekochte Köstlichkeit gefunden hatten und mit dem Erfolgserlebnis tirilierend auf Vater und Mutter zusprangen? Wo sind heute diese Erfolgserlebnisse, wo ist heute die freie Natur? Ihr NASCOM soll Ihnen dabei helfen, die Zeit für einen erfrischenden Augenblick zurückzudrehen. Hier das "Ostereiersuchprogramm", das Ihnen und Ihrer Familie das diesjährige Osterfest unvergeßlich machen wird.
Geben Sie zunächst den Befehl C0C80 0C81 FFFF ein und schaffen Sie sich dadurch in Ihrem Speicherbereich ein Abbild der unberührten Natur. Dann wählen Sie vier beliebige Adressen in Ihrem RAM und laden Sie durch MXXXX mit den ASCII Codes 0E, 0F, 1B und (oder) 4F. Dadurch haben Sie vier symbolische Ostereier versteckt. Nun müssen Sie den NASCOM nur noch zwei bis drei Wochen eingeschaltet lassen, bis Sie die entsprechenden Adressen vergessen haben; dadurch wird ein richtiges "Verstecken" simuliert.
Am Morgen des Ostersonntags sollten Sie nun versuchen, durch willkürliches Kontrollieren verschiedener Speicherzellen die "Ostereier" zu finden. Wirklichen Spaß macht das natürlich erst bei 64 K Speicherplatz. Welche Freude für die ganze Familie, wenn die vier Eier gefunden sind.
In diesem Sinne wünscht Ihnen allen ein wunderschönes Osterfest
Ihr NASCOMPL

IMPRESSUM

REDAKTION: Günter Böhm, Günter Kreidl
Wolfgang Mayer-Gürr, Josef Zeller
RESSORTS:
Maschinenprogramme:
Günter Böhm, [redacted]
[redacted] Karlsruhe [redacted], Tel.[redacted]
Günter Kreidl
[redacted], [redacted] Straelen
Tel.[redacted]
BASIC und FLOPPY:
Wolfgang Mayer-Gürr
[redacted]
[redacted], [redacted] Recklinghausen
Tel.[redacted]
HARDWARE:
Josef Zeller,[redacted],[redacted] Neu-Ulm
Verlag NASCOM Journal, c/o MK-Systemtechnik
Pater-Mayer-Str.6, 6728 Germersheim,
Tel.07274/2756, Telex 453 500 mks d.
Vertrieb: Direktvertrieb durch den Verlag
Erscheinungsweise: monatlich
Bezugspreis: Im In- und Ausland 48,- für ein Jahresabonnement.

Bei Bestellung nach dem 1.1. werden die fehlenden Hefte mit der ersten Lieferung bis zum Bestellzeitpunkt automatisch mitgeliefert.

Bezugsmöglichkeiten: Durch Bestellung bei MK - Systemtechnik,(beigefügte Bestellkarte)
Bankverbindungen: Alle Zahlungen für das NASCOM-JOURNAL unter Angabe der Rechnungsnummer. bindungen: Alle Zahlungen für das NASCOM-JOURNAL unter Angabe der Rechnungsnummer. Fa.Michael Klein
Zahlungen: Nach Eingang Ihrer Bestellung erhalten Sie von uns die ausstehenden Hefte bis zur aktuellen Ausgabe sowie eine Rechnung. Bitte, zahlen Sie dann den Rechnungsbetrag auf unser Sonderkonto (s.o.) ein. Bitte keine Vorauszahlungen!!
Bitte, Anfragen wegen Abonnements oder Lieferung nicht an die Redaktion sondern nur an den Verlag. Die Autoren tragen die Verantwortung für ihre Beiträge selbst. Für Fehler in Text, Bildern und sonstigen Angaben kann keine Haftung übernommen werden.

KLEINANZEIGEN

Jeder Abonnent kann Kleinanzeigen bis 40 Wörter aufgeben!

VERKAUFE NASCOM-CLD-FLOPPY:
Controller-Karte, 2 Laufwerke, 30 Disketten, CLDDOS, CLD-BASIC, CLD-Assembler, Editor, Disassembler, Microsoft-Double Precision-BASIC, Microsoft-FORTRAN, Microsoft-Macroassembler, incl.viele weitere Software
Tel. [redacted] (18-23°°h)

SUCHE Drucker oder Fernschreiber für NASCOM 2, ab 1.4.82
K.Körner, [redacted]
[redacted] Tel. [redacted]

BIETE CLD-BASIC, NASPEN, NAS-DIS, NAS-DEB. TOOLKIT oder ZEAP 2.0 zum Tausch gegen NASSYS 3, Tiny PASCAL, NAS-CHESS oder andere Programme
Johannes C. Lotter, [redacted]
Tel. [redacted]

Suche BASIC-Toolkit
Listing oder auf EPROM 2708/2716
H.Broja Tel. [redacted]

SUCHE EPROM-Karte für ELZET 80/KONTRON (2508 oder 2516)
Jochen Münster , [redacted]

VERKAUFE:
NAS-SYS Assembler ZEAP 2.0 4K DM 140.-
NAS-SYS Disass. NASDIS 3K DM 100.-
NAS-SYS DEBUG 1K DM 40.-
SUCHE: 8K BASIC (Tape Version) für N 1 Betriebssystem NAS-SYS 1
B.Schäfer Tel. [redacted] (nach 17°°h)

VERKAUFE S100 Dyn. RAM Karte 16 K best. 220.-DM. Floppy FDD 100-8D 300.-DM
SUCHE NASCOM 2-Schaltplan, Mini DCR
R.Wieskämper Tel. [redacted]

VERKAUFE NASCOM 1
mit NASSYS und Gehäuse 600.-
zusätzlich 8 K stat. RAM 200.-
Tel. [redacted]

Ich biete ein PASCAL-System für CLD-DOS an, das aus einem Compiler, der P-Code erzeugt, einem Interpreter, 14 Programmbeispielen und einem Handbuch besteht. Dieses PASCAL verarbeitet Zahlen zwischen +/-1.7E+/-38 mit einer Genauigkeit von 9 Digits. Random-Files
Wolfgang Mayer-Gürr
Treptower Str.2
4350 [redacted] Tel. [redacted]

VERKAUFE: Hardwareuhr mit entsprechender Software. Die Uhr ist Akku-gepuffert, wird wie ein RAM angesprochen. Anzeige: Wochentag, Datum, Monat, Stunde, Minute, Sekunde, Zehntelsek. Schaltjahre werden berücksichtigt.
M.Reimer, [redacted]
[redacted] Tel. [redacted]

VERKAUFE komplette 16 K EPROM Platine zus. mit 1 K DEBUG, 3 K Disassembler, 4 K ZEAP 2.0 und 8 K BASIC; alles in 2716 zusammen ca. DM 450.-
Otto Fößel, [redacted]
Tel. [redacted]

VERKAUFE 64 K dyn.RAM-Karte für 65xx, 68xx, 8080, 8085, Z80 geeignet. ECB-Bus-kompatibel; mit NASC.2 getestet.
E.Obermaier [redacted]
[redacted]

VERKAUFE: ZEAP 2.0 4X 2708
NASDIS 3X 2708
DEBUG 1X 2708
NASPEN 2X 2708
TOOLKIT 2X 2708
für NASSYS Preis VB DM 350.-
Krefting, [redacted]
Tel. [redacted]

VERKAUFE Bufferboard f. NASC.1 DM 100.-
32 K RAM B. voll bestückt DM 400.-
Floppy Controllerkarte für NASCOM 1 mit Kabel DM 600.-
bei Komplettabnahme DM1000.-
Hans Andriessen, [redacted]
[redacted]

Best.-Nr.	Produktbeschreibung	Preis incl. 13 % MWSt
N-101	NASCOM 1 Bausatz	DM 935,00
N-111	NASCOM 1 aufgebaut, getestet	DM 1 085,00
N-202	NASCOM 2 Bausatz 8k Stat. RAM	DM 1 950,00
N-203	NASCOM 2 Bausatz 16k Dyn. RAM	DM 2 065,00
N-204	NASCOM 2 Bausatz 32k Dyn. RAM	DM 2 140,00
N-205	NASCOM 2 Bausatz 48k Dyn. RAM	DM 2 195,00
N-212	NASCOM 2 aufgebaut, 8k Stat. RAM	DM 1 995,00
N-213	NASCOM 2 aufgebaut, 16k Dyn. RAM	DM 2 398,00
N-214	NASCOM 2 aufgebaut, 32k Dyn. RAM	DM 2 458,00
N-215	NASCOM 2 aufgebaut, 48k Dyn. RAM	DM 2 495,00
N-220	Mini-Motherboard für 4 Karten	DM 29,58
N-221	Motherboard für 14 Zusatzboards	DM 67,80
N-230	Buffer-Bus-Board Jedes RAM »B«-Board läßt sich auf 48K Dyn. RAM ausbauen. Eine Erweiterung auf 192K RAM durch den sog. Page-Mode-Kit ist möglich.	DM 230,00
N-260	RAM »B«-Karte mit 16K Dyn. RAM	DM 525,00
N-261	RAM »B«-Karte mit 32K Dyn. RAM	DM 584,00
N-262	RAM »B«-Karte mit 48K Dyn. RAM	DM 644,00
N-265	Aufpreis für aufgebaute und getestete Karte	DM 195,00
N-270	Ein/Aufgabekarte für drei PIO, ein CTC und einen UART. Bausatz ohne die Optionen.	DM 295,00
N-271	Ein/Ausgabekarte wie N-270, aufgebaut	DM 349,00
N-272	PIO-Option für I/O-Board	DM 78,00
N-273	CTC-Option für I/O-Board	DM 92,00
N-274	UART-Option für I/O-Board	DM 99,00
N-280	EPROM-Programmer, KIT	DM 185,00
N-281	EPROM-Programmer, Fertiggerät	DM 229,00
N-301	Netzteil 5V/3A,-5V/1A + 12V/1A, -12V/1A	DM 215,00
N-302	Netzteil wie N-301, aufgebaut, getestet	DM 247,50
N-310	Veroframe 19-Zoll Rahmen 5 HE, Bausatz	DM 246,00
N-313	Tastaturgehäuse, paßt für NC1/2 mit fertigem Ausschnitt	DM 32,00
N-314	NASCOM - Gehäuse aus schlagfestem Kunststoff, Lederstruktur mit Tastaturausschnitt, für NC + 3 Boards	DM 236,00
N-315	Einschubrahmen für N-314 zur Montage von drei Zusatzkarten	DM 148,00
N-330	NASBUG T2 Monitor	DM 49,00
N-331	NASBUG T4 Monitor	DM 99,00
N-332	NAS-SYS 1 Monitor	DM 125,00
N-333	Umschaltkarte für zwei Betriebssysteme	DM 75,00
N-335	NAS-SYS 3 Monitor	DM 135,00
N-340	2K Tiny Basic Interpreter	DM 99,00
N-341	Erweiterung auf Super Tiny Basic	DM 55,00
N-342	3K Super Tiny Basic	DM 149,00
N-343	8K Microsoft Basic auf Kassette	DM 129,00
N-344	8K Microsoft Basic auf 8K Rom	DM 155,00
N-345	8K Microsoft Basic auf 8 Stk 2708	DM 379,00
N-351	Assembler für NASBUG auf 3 Stk 2708	DM 269,00
N-352	»NAS-DIS«-Disassembler für NAS-SYS	DM 199,00
N-353	DEBUG Programmierhilfe, NAS-Dis erforderlich!	DM 98,00
N-357	ZEAP 2.0 Assembler für NAS-SYS auf 4 Eprom 2708	DM 245,00
N-358	ZEAP 2.0 auf Band	DM 165,00
N-361	NASPEN-Texteditor (für NAS-SYS)	DM 197,00
N-370	Schach auf Kassette	DM 98,00
N-400	Grafik — Rom für NASCOM 2	DM 98,00
N-401	Econografics - Kit, erweitert NC1 auf Grafik des NC2	DM 210,00
N-500	MKS-FLOPPY, mit einem BASF-Laufwerk Typ 6106, ca. 100K Speicherkapazität, aufgebaut, getestet, DOS Macroassembler, Basic, 1 Jahr Softwarepflege	DM 1 749,00
N-501	wie N-500, jedoch mit Gehäuse und Netzteil	DM 2 144,00
N-502	2. Laufwerk	DM 994,00
N-503	Microsoft Double-Precision Basic	DM 429,00
N-504	FORTRAN	DM 449,00
N-520	IMP-Normalpapierdrucker	DM 2 136,00
N-521	Farbband für IMP-Drucker	DM 46,00
N-530	43-Pol. Anschluss an NC1-Grundkarte mit Kabel	DM 32,00
N-531	77-Pol. Busstecker	DM 29,00
N-532	Zeichengenerator für Nascom 1	DM 65,72
N-533	Z 80-CPU 2 MHZ	DM 45,97
N-534	Z 80-PIO 2 MHZ	DM 45,97
N-535	Z 80-CTC 2 MHZ	DM 52,00
N-536	UART (im 6402 oder COM 8017)	DM 38,00
N-537	Speicherbaustein 4116, 16K x 1, 200 ns	DM 12,50
N-538	MK 4118 1K x 8 STAT. RAM	DM 36,70
N-539	1K x 8 EPROM 2708	DM 26,00
N-540	2112, 1K 1 STAT, RAM	DM 4,93
N-600	Tastaturerweiterung von NC1 auf NC2	DM 88,00
N-601	Taste mit Kopf einzeln, ungraviert	DM 7,89
N-611	12"-SANYO Videomonitor, gruen, 18 MHZ	DM 698,00
N-620	Datenkassetten, C10	
	10 Stück	DM 19,80
	20 Stück	DM 36,00
	50 Stück	DM 87,50
	100 Stück	DM 160,00
N-630	Tabellierpapier für IMP-Drucker ca. 2000 Blatt	DM 58,00
N-632	Adressaufkleber 80x34 mm, für IMP	DM 69,80
N-640	NASCOM 2 - Dokumentation, deutsch	DM 56,00
N-641	Ordner für NASCOM - Dokumentation	DM 18,00
N-642	FLOPPY-DISK-Dokumentation	DM 49,00
N-643	Handbuch 8K BASIC, deutsch	DM 19,80
N-644	Handbuch "Z 80 Assemblersprache"	DM 29,80
N-649	"Z 80-Applikationsbuch, führt mit Beispielen in die Z80 Programmierung ein. Geeignet als Lehr- und Übungsbuch für NASCOM-Maschinensprach-programmierung	DM 32,00
N-650	NASCOM-JOURNAL Jahresabonnement	DM 48,00
N-651	Hefte NASCOM-JOURNAL 1980	DM 39,00
Komplettsysteme		
N-661	NASCOM 2 in Gehäuse mit Stromversorgung und 8K RAM, aufgebaut und getestet	DM 2 735,00
N-662	NASCOM 2 in Gehäuse mit Stromversorgung und 16K RAM, aufgebaut und getestet	DM 2 995,00
N-663	Speichererweiterung auf 32K	DM 96,00
N-664	Speichererweiterung auf 48K	DM 96,00
N-670	FLOPPY-DISK, 1.Laufwerk, 350 KBYTE	DM 2 995,00
N-671	wie N-670, Doppellaufwerk	DM 3 895,00
N-672	CP/M Version 2.2	DM 595,00
N-673	NAS-DOS Betriebssystem für FLOPPY-Disk	DM 325,00