

nascom journal

Zeitschrift für Anwender des NASCOM 1 oder NASCOM 2

3. Jahrgang · Juni 1982 · Ausgabe 6

Herausgeber:

MK-SYSTEMTECHNIK Michael Klein · Pater-Mayer-Straße 6 · 6728 Germersheim/Rhein
Telefon (0 72 74) 20 93 · Telex 453500 mks d

MK-SYSTEMTECHNIK Thomas Gräfenecker · Kriegsstraße 164 · 7500 Karlsruhe · Telefon (07 21) 2 92 43
MK-SYSTEMTECHNIK Michael von Keltz · Pfaffenberg 4 · 5650 Solingen 1 · Telefon (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

2	NASCOM Journal INTERN	
3	Leserbriefe	
4	SNAKE	Tom D.Rüdebusch
5	Postscheck	Klaus Mombaur
6	Schreibmaschinentreiber	Wolfgang von Jan
7	Seite für Kinder	Günter Böhm
8	BLS-Pascal Teil2	Michael Bach
9	Robot Intelligence	Günter Böhm
12	Software f. Video-Karte	A.Zippel/D.Oberle
20	Seite(n) für Einsteiger	Günter Böhm
21	Spielautomat	Erich Mehnert
22	N2 Hardware-Tip	Peter Brendel
23	NASCOMPL/Impressum	
24	MKS-Angebote	

nascocom journal

intern

Liebe Leser,
nun ist es genau 1 Jahr her, daß das Redaktionsteam die Arbeit für das Journal begann. Mit Hilfe der vielen Mitarbeiter aus dem Leserkreis ist es uns gelungen, das Journal ein ganzes Stück weiterzubringen, zumindest muß man das den vielen positiven Leserzuschriften entnehmen. Unseren herzlichen Dank an all die aktiven Leser, die zum Gelingen beigetragen haben.

Dies wäre wohl ein Anlaß, eine Jubiläumsausgabe herauszubringen. Leider erreichten uns einige Zuschriften zu spät, um sie noch in dieser Ausgabe unterzubringen; deshalb diesmal ein etwas dünneres Heft. Betrachten Sie es als die "Ruhe vor dem Sturm": die nächste Nummer (ein Doppelheft, das Ende August erscheint), soll dafür besonders umfangreich werden, wie es einer Jubiläumsnummer zusteht. Auch wird sie einige sehr interessante Beiträge enthalten. Auf einen sei schon hingewiesen: ein Programm zur Sprachausgabe ohne zusätzlichen Hardwareaufwand!

Trotz bereits vorliegenden Materials will ich den obligatorischen Leseraufruf nicht vergessen: Schicken Sie uns Beiträge, schicken Sie uns Beiträge, schicken Sie uns..... Es wurde uns bereits wieder ein Vorschlag für einen Wettbewerb gemacht: NASCOM und Umwelt. Unter diesem Thema sollten Hardware- bzw. mechanische Konstruktionen in Wettbewerb treten. Was halten Sie von dem Thema? Falls genügendes Interesse besteht (sprich Teilnehmer vorhanden wären), könnten wir das Preisausschreiben starten.

Im Listing zum Schreibmaschinen-Interface von Günter Kreidl (Heft 5) müssen zwei Zellen eingefügt werden.

1305 PUSH BC
1335 POP BC

Eigentlich sollte schon in diesem Heft ein vereinfachtes Interface vorgestellt werden, das nicht mehr als die vorhandenen PIO Leitungen benötigt und mit Interrupts arbeiten kann, sodaß die CPU beim Ansteuern der Typenradmaschine entlastet wird. Der Prototyp hatte aber einige Geburtsschwierigkeiten und wird im nächsten Heft nachgeliefert einschließlich eines vollständigen Terminalprogramms zur Verwendung der Schreibmaschinentastatur über PIO.

Was die Bestellung der Folien zum Video-Interface von Zippel/Oberle angeht, so ist da einiges schiefgelaufen. Leider habe ich übersehen, daß es sich um eine beidseitig beschichtete Platine großen Formats handelt. Europa-Format-Folien werden mir mit DM 5.- berechnet, so scheint mir der Preis von DM 10.- für zwei Folien dieser Größe nicht übertrieben. Leider habe ich im letzten Heft nur DM 5.- angegeben. Ich hoffe die Besteller, die schon einen Fünfmarkschein eingesandt haben, gleichen das irgendwie aus. Bis jetzt mußte ich den Fehlbetrag selbst drauflegen.

Mit dem Rückumschlag klappt es auch nicht so richtig. Die Folien sind so groß (ca 25x25 cm), daß sie bestimmt nicht in einen kleinen Standardumschlag passen. Schicken Sie also mit der Bestellung bitte einen KARTONIERTEN Umschlag DIN A 4 (wegen des Knickens) oder legen Sie DM 2.- in Briefmarken für Porto und Verpackung bei. (Hoffentlich habe ich mich hier nicht wieder verkalkuliert; ich bin kein guter Geschäftsmann). Das Interesse an der Video-Karte ist mit Recht sehr groß. Helfen Sie, meine Arbeit (und evt. Unkosten) so gering wie möglich zu halten.

Verschiedentlich wurde schon angefragt, welche ASCII-Codes denn eigentlich für die Umlaute verwendet würden. Deshalb hier nun endlich eine Tabelle der Codes und der Tastenbelegung auf dem NASCOM (mit Tastaturerweiterung).

Ä	5B	⌈
Ö	5C	⋄
Ü	5D	⌋
ä	7B	CTRL+;
ö	7C	SHIFT,CTRL+<
ü	7D	SHIFT,CTRL+=
ß	7E	SHIFT,CTRL+>

So viel für diesmal. Viel Spaß beim Schmökern.

Ihr Günter Böhm

LESERBRIEFE

Nun noch ein anderes Thema: Anglizismen in der Rechner-Literatur. Vieles bei Rechnern liest man zunächst im Englischen, einen deutschen Begriff dafür weiß man nicht oder es gibt ihn (noch) nicht. Hinzu kommen einfache Übersetzungsfehler. Ein paar Beispiele: "control" ist oft nicht mit "Kontroll-" zu übersetzen; insbesondere heißt es "Steuerzeichen" und nicht "Kontroll-Charakter", womit gleichzeitig auch darauf hingewiesen sei, daß ein "Charakter-Generator" ein psychologisches Phänomen wäre. "File" und "Datei" sind gleichbedeutend, "Datenfiles" ist doppelt gemoppelt. Also: Leute, versucht die Verhunzung der Sprache wenn möglich nicht mitzumachen! Übrigens ist das Journal auf dem Gebiet ja noch verhältnismäßig gut, bei manchen anderen Blättern hat man den Eindruck, daß "Fachenglisch" Verständnis ersetzt.

Noch eine zusätzliche Information zum Acht Damen-Problem aus dem letzten Journal: die Lösung benötigt mit dem BLS-Pascal 24sec bei 2MHz Takt.

Michael Bach, Stegen

Damit AUTO LINE NUMBER des Toolkit zusammen mit NASSYS3 wieder richtig funktioniert, biete ich eine Lösung an, die nur minimalen Aufwand erfordert.

Dazu wird in der UOUT-Routine des Toolkit lediglich die Reihenfolge von 6 Bytes vertauscht:

```
ALT: # B151 CD AD F9 EF 20 00
```

```
NEU: # B151 EF 20 00 CD AD F9
```

Dies hat zur Folge, daß nun der Blank vor der Zeilennummer ausgegeben wird und damit das von NASSYS3 verursachte Löschen des 1. Zeichens der Zeile wirkungslos bleibt.

Wolfgang Haslbeck, 85000 Nürnberg

Heinz Heim, Schwieberdingen und Andreas Dieckmann, Heidelberg, schlagen vor, die Adresse B15B im Toolkit mit 0C (statt 08) zu laden, um es mit NASSYS3 lauffähig zu machen.

Wer sein NASCOM ROM BASIC V.4.7 im RAM laufen läßt oder entsprechend abändert, kann mit dem BASIC Befehl INP(0) das Keyboard abfragen. Der Befehl liefert den Wert 0, wenn keine Taste gedrückt war. Folgende Speicherstellen sind zu ändern:

```
E0E0 0D 28 63 29 20 31 39 37
E0E8 38 20 62 79 20 4D 69 63
E0F0 72 6F 73 6E 66 74 0D 00
E0F8 B7 C2 10 3E DF 62 D8 AF
E100 C9 00 00 4D 65 6D 6F 72
E108 79 20 73 69 7A 65 00 22
F448 F8 E0
```

zB : Warten auf ein Zeichen:

```
10 A=INP(0):IF A=0 THEN 10
20 PRINT CHR$(A)
```

Vorschläge zu Leserbeiträgen:

1. Leserinitiative zum Erstellen eines kommentierten Disassemblerlistings z.B. von ZEAP und 8K-BASIC oder NASPEN. (Schwach kommentiert liegt bereits alles vor).

2. Gemeinsam ein Programm erstellen, das es gestattet, CP/M Software (geladen über z.B. V24) auf einem NASCOM mit 48 kByte und Recorder laufen zu lassen.

Constantin Olbrich, Berlin

In Heft 10-81 veröffentlichten wir eine Kontaktadresse für Interessenten an lokalen Computerbekanntschaften. Die Resonanz war nicht gerade überwältigend. Es folgt eine Liste der "Kandidaten" mit Angabe des Systems.

Klaus Bott, [REDACTED]

[REDACTED] Rastatt Tel. [REDACTED]

(Steuer- und Regeltechnik) NASCOM 2, NASSYS 1

ZEAP 2.0, BASIC. Inter. Cassettentausch

Stefan Bürger, Dipl. Ing., [REDACTED]

[REDACTED] Tamm-Hohenstange Tel. [REDACTED]

(gesch.) [REDACTED]

NASCOM 1, NASSYS 1, 8K-Tape-BASIC, ECB-Bus

Anselm Fabig, Berlin

Tel. [REDACTED] DD6ES

Karlheinz Körner, [REDACTED]

[REDACTED] Germersheim Tel. [REDACTED]

NASCOM 2, NASSYS

Tom D. Rudebusch, Kropbacher Weg 40
6300 Gießen Tel.
NASCOM 2, 'NASSYS' mit Fernschreiber

Werner Öhring, Lothringer Str. 22
8000 München Tel. 089/4480671
(gesch. 089/722-52481)
Personal Computing Club München
NASCOM 1, 2*32K RAM + 8K EPROM
(FDC)/Prommer, Diablo-Drucker/ECB-Rechner
mit 5" und 8" FD

Hermann Schmidt, Zaunbusch 28
5600 Wuppertal Tel. 0202/732323
NASCOM 1, T2 Interesse: Speichererw.
NASSYS 3, BASIC-SW

SNAKE

von Tom D. Rudebusch

```

10 REM "SNAKE"
20 REM
30 REM TOM D. RUEDEBUSCH
40 REM
50 REM GIESSEN
60 REM TEL.
70 REM
80 CLS
90 FOR A=3264 TO 3272 STEP 2
100 READQ:DOKE A,Q:NEXT
110 POKE 3274,233:DOKE 4100,3264
120 PRINT "*****Tastenbelegung*****":PR
INT
130 PRINT " W oben P"
140 PRINT " Z unten ."
150 PRINT " A links L"
160 PRINT " S rechts ;"
170 PRINT:PRINT:PRINT
180 PRINT"Zwei Initialien des ersten Spielers:
";
190 INPUTS1$
200 PRINT"Zwei Initialien des zweiten Spielers:
";
210 INPUTS2$
220 CLS:L=0:F1=0:RESTORE650
230 FOR Q=1 TO 4:READPX,PY:GOSUB670:NEXT
240 FOR Q=1 TO 4:READPX,PY:GOSUB680:NEXT
250 SCREEN12,7:PRINTLEFT$(S1$,2)
260 SCREEN37,7:PRINTLEFT$(S2$,2)
270 FOR X=0 TO 95:SET(X,0):SET(X,44):NEXT
280 FOR Y=0 TO 44:SET(0,Y):SET(95,Y):NEXT
290 X1=27:Y1=19:X2=70:Y2=Y1
300 X3=1:Y3=0:X4=-1:Y4=Y3:GOTO390
310 E=USR(0)
320 L=L+1:L$=STR$(L)+" "
330 POKE3040,ASC(MID$(L$,2,1))
340 POKE3041,ASC(MID$(L$,3,1))
350 POKE3042,ASC(MID$(L$,4,1))

```

```

360 IF E=0 THEN390
370 IFE=87ORE=65ORE=83ORE=90 THENE1=E:GOSUB690
380 IFE=80ORE=76ORE=59ORE=46 THENE2=E:GOSUB730
390 X1=X1+X3:Y1=Y1+Y3:X2=X2+X4:Y2=Y2+Y4
400 IF POINT(X1,Y1)=1 THEN F1=1
410 IF POINT(X2,Y2)=0 THEN440
420 IF F1=1 THEN480
430 GOTO460
440 IF F1=1 THEN470
450 F1=0:SET(X1,Y1):SET(X2,Y2):GOTO310
460 G$=S1$+" gewinnt":S1=S1+L:GOTO490
470 G$=S2$+" gewinnt":S2=S2+L:GOTO490
480 G$="Unentschieden"
490 FOR X=1 TO 2000:NEXT:CLS
500 SCREEN10,1:PRINT"***** Spielstand *****
*****"
510 SCREEN14,3:PRINTG$:L;" Punkte."
520 SCREEN13,6:PRINTS1$;" hat Jetzt ";S1;" Punk
te."
530 SCREEN13,8:PRINTS2$;" hat Jetzt ";S2;" Punk
te."
540 SCREEN14,11:PRINT"Sieg bei 1000 Punkten"
550 IF S1<1000 THEN570
560 SCREEN18,13:PRINT"SIEG fuer ";S1$:GOTO600
570 IF S2<1000 THEN590
580 SCREEN18,13:PRINT"SIEG fuer ";S2$:GOTO600
590 FOR X=1 TO 3000:NEXT:GOTO220
600 FOR X=1 TO 1500:NEXT
610 SCREEN16,15:INPUT"Nach ein Spiel ";A$
620 IF LEFT$(A$,1)="J"THENS1=0:S2=0:GOTO220
630 END
640 DATA 25311,312,18351,10927,-8179
650 DATA 21,16,71,16,21,21,71,21,21,17,71,17
660 DATA 26,17,76,17
670 Y=PY:FOR X=PX TO PX+5:SET(X,Y):NEXT:RETURN
680 X=PX:FOR Y=PY TO PY+3:SET(X,Y):NEXT:RETURN
690 IF E1=87 THEN Y3=-1:X3=0:RETURN
700 IF E1=65 THEN X3=-1:Y3=0:RETURN
710 IF E1=83 THEN X3=1:Y3=0:RETURN
720 IF E1=90 THEN Y3=1:X3=0:RETURN
730 IF E2=80 THEN Y4=-1:X4=0:RETURN
740 IF E2=76 THEN X4=-1:Y4=0:RETURN
750 IF E2=59 THEN X4=1:Y4=0:RETURN
760 IF E2=46 THEN Y4=1:X4=0:RETURN

```

Das Spiel SNAKE von Tom D. Rudebusch ent-
spricht im Ablauf exakt dem Doppelwurm von
Michael Bach.(Unser Gewinner im letzten
Preisausschreiben).

Für manche Leser kann es interessant sein,
einmal die BASIC-Version mit der Assembler-
ausführung zu vergleichen.

In Zeile 90/100 wird ein Maschinenprogramm
zur Tastaturabfrage (INKEY\$) geladen. Die
Daten dazu in Zeile 640 ergeben wieder unser
altes Problem mit den beiden BASIC-Versio-
nen. Wenn Sie die Zeile folgendermaßen än-
dern, läßt sie sich auf beiden Versionen
anwenden. Ein Tip für zukünftiges Program-
mieren. Red.

```
640 DATA 25311,312,18351,-15441,-3854
```

Postscheck von Klaus Mombaur

Beispiel eines ausgefüllten Postschecks:

```

**siebenundsiebzigtausendsiebenhundert*
**siebenundsiebzig*****
Herrn Günter Bohm Karlsruhe

Nürnberg 7.7.77

10 REM blaues Postscheckformular ausfüllen
15 REM
20 REM C by K.Mombaur, Nuernberg,
25 REM
30 REM Drucker initialisieren
40 OUT6,15:OUT7,207:OUT7,245
50 OUT5,255:OUT5,247:OUT5,255
60 REM Betr.Art Kl
70 DOKE3840,289:DOKE3842,-8448:DOKE3844,-14065
80 DOKE4100,3840:U=USR(1)
100 CLEAR200:CLS
110 PRINT"Ich fueelle einen Postscheck aus!"
120 PRINT:PRINT"Spannen Sie dazu das blaue Form
ular mit
130 PRINT"Mitte - '8500' auf Oberkante Blech ei
n."
140 PRINT
160 INPUT"An wen:";A$
165 IF A$="" THEN A$=" "
170 IF LEN(A$)>42 THEN PRINT"Zu lang!":GOTO160
180 PRINT:INPUT"DM - Betrag in Ziffern:";B$
185 IF LEN(B$)>5 THEN PRINT"Zu hoch!":GOTO180
190 IF LEN(B$)<1 THEN B$="0"
200 PRINT:INPUT"Pfg - Betrag in Ziffern:";P$
210 IF LEN(P$)>3 THEN PRINT"Nur 2 Ziffern!":GOTO200
214 IF LEN(P$)<1 THEN P$="00"
216 IF LEN(P$)<2 THEN P$="0"+P$
220 PRINT:INPUT"Ausstellungsart (wenn < > Nbg):"
;0$
222 IF 0$="" THEN 0$="Nürnberg"
225 IF LEN(0$)>10 THEN PRINT"Zu lang!":GOTO220
230 PRINT:INPUT"Datum:";D$
232 IF D$="" THEN D$=" "
235 IF LEN(D$)>9 THEN PRINT"Zu lang!":GOTO230
240 PRINT:INPUT"Verwendungszweck:";V$
245 IF V$="" THEN V$=" "
250 IF LEN(V$)>52 THEN PRINT"Zu lang!":GOTO240
260 PRINT:W$="D":PRINT"Drucken ?"
270 INPUT"Wiederh.= W";W$
280 IF W$="W" THEN 100
290 PRINT:PRINT:PRINT"Vergessen Sie die Untersc
hrift nicht!"
300 REM Druckart vorbereiten
310 FOR J=1 TO 12:READ X:GOSUB2200:NEXT
315 X=42:GOSUB2200:GOSUB2200
320 L=LEN(B$):ZL=0
325 REM DM-Betrag in Buchstaben wandeln
330 M1=VAL(MID$(B$,1,1)):M2=VAL(MID$(B$,2,1))
340 IFL<>5 THEN 550
345 REM Zehntausender *****
350 A=1
355 REM Zt+T.Stelle=11or12?
370 IF W1=1 THEN IF W2=1 THEN S$="e1f":GOSUB2000:GOTO
570
380 IF W1=1 THEN IF W2=2 THEN S$="zw61f":GOSUB2000:GO
TO570
385 REM T.Stelle=0?
390 IF W2<>0 THEN 430
400 RESTORE
410 READS$:IFS$(">")+ " THEN 410
420 GOSUB2130:GOSUB2000:GOTO570
425 REM Zt.Stelle=1?
430 IF W1<>1 THEN 430
440 A=2:GOSUB2100:GOSUB2000

```

```

450 S$="zehn"
460 GOSUB2000
470 GOTO 570
475 REM Zt.Stelle, "und", T.Stelle x10
480 A=2:GOSUB2100:GOSUB2000
490 S$="und"
500 GOSUB2000
510 A=1:RESTORE
520 READS$:IFS$(">")+ " THEN 520
530 GOSUB2130:GOSUB2000
540 GOTO570
545 REM Tausender*****
550 IFL<>4 THEN 600
560 A=1:GOSUB2100:GOSUB2000
570 S$="tausend":GOSUB2000
575 REM hundert nur wenn (>0)
580 IFL=4 THEN IF W2=0 THEN A=3:GOSUB2300:GOTO820
590 IFL<>5 THEN 600
592 W3=VAL(MID$(B$,3,1))
593 IF W3=0 THEN A=4:GOSUB2300:GOTO820
595 REM Hunderter *****
600 REM
610 IFL<4 THEN 650
615 A=3:IFL=4 THEN A=2
620 GOSUB2100:GOSUB2000
630 S$="hundert":GOSUB2000
650 IF ZL<30 THEN GOSUB2300:X=13:GOSUB2200
695 REM H.Stelle wenn <4 Stellen *****
700 IFL<>3 THEN 750
705 X=42:GOSUB2200:GOSUB2200
710 A=1:GOSUB2100:GOSUB2000
720 S$="hundert":GOSUB2000
730 A=2:GOTO823
750 IFL=5 THEN A=4:GOTO822
760 IFL=4 THEN A=3:GOTO822
795 REM Zehner *****
800 IFL<2 THEN 990
810 A=1
812 GOTO 822
815 REM 2+E.Stelle=11or12?
820 X=13:GOSUB2200
822 X=42:GOSUB2200:GOSUB2200
823 W1=VAL(MID$(B$,A,1)):W2=VAL(MID$(B$,A+1,1))
824 IF W1=0 AND W2=0 THEN 1100
826 IF W1=0 THEN 990
828 IF W1=1 THEN IF W2=1 THEN S$="e1f":GOSUB2000:GOTO
1100
830 IF W1=1 THEN IF W2=2 THEN S$="zw61f":GOSUB2000:GO
TO1100
835 REM E.Stelle=0?
840 IF W2<>0 THEN 880
850 RESTORE
860 READS$:IFS$(">")+ " THEN 860
870 GOSUB2130:GOSUB2000:GOTO1100
875 REM Zt.Stelle=1?
880 IF W1<>1 THEN 930
890 A=A+1:GOSUB2100:GOSUB2000
900 S$="zehn"
910 GOSUB2000
920 GOTO1100
925 REM E.Stelle, "und", Z.Stelle x10
930 A=A+1:GOSUB2100:GOSUB2000
940 S$="und"
950 GOSUB2000
960 RESTORE
970 READS$:IFS$(">")+ " THEN 970
980 A=A+1:GOSUB2130:GOSUB2000:GOTO1100
985 REM Einer *****
990 A=A+1:IFL<>1 THEN 1000
995 X=42:GOSUB2200:GOSUB2200
1000 IF W1=0 AND L=1 THEN S$="null":GOTO1020
1010 GOSUB2100
1020 IFS$="ein" THEN S$="eine"
1030 GOSUB2000
1100 GOSUB2300
1110 FOR J=1 TO 10:X=32:GOSUB2200:NEXT
1120 FOR J=1 TO 3:X=ASC("*"):GOSUB2200:NEXT
1130 S$=B$:GOSUB2000
1140 X=ASC(","):GOSUB2200
1150 S$=P$:GOSUB2000
1160 FOR J=1 TO 3:X=ASC("x"):GOSUB2200:NEXT
1170 X=13:GOSUB2200
1195 REM An wen
1200 X=32:FOR J=1 TO 5:GOSUB2200:NEXT
1210 S$=A$:GOSUB2000
1220 X=13:GOSUB2200
1230 X=27:GOSUB2200:X=84:GOSUB2200
1240 X=54:GOSUB2200:X=48:GOSUB2200

```

```

1250 X=13:GOSUB2200
1295 REM Ausst.Ort, Datum
1300 S#=0$:GOSUB2000
1305 S#=" ":GOSUB2000
1310 S#=D$:GOSUB2000
1317 RESTORE
1318 FOR J=1 TO11:READ X :GOSUB2200: NEXT
1320 X=13:GOSUB2200
1395 REM Verw.zweck
1400 S#=""
1410 GOSUB2000
1420 S#=U$
1430 GOSUB2000
1900 X=13:GOSUB2200
1910 PRINT:INPUT"Selben Druck wdh = w":W$
1920 IFW$="w"THEN RESTORE:W$="N":GOTO300
1930 END
2000 REM U-Ausgabe *****
2010 FORA2=1TO LEN(S$)
2020 X=ASC(MID$(S$,A2,1)):ZL=ZL+1
2030 GOSUB 2200
2040 NEXT A2
2050 RETURN
2100 REM U-DATA SETZEN
2110 RESTORE
2120 READS$:IFS$(1)="*"THEN2120
2130 FORA1=1TO VAL(MID$(S$,A1,1)):READS$:NEXT
2140 RETURN
2200 REM U-Drucken
2210 OUT4,X:OUT5,253:OUT5,255:DATA15,16,18
2220 RETURN
2300 REM U-Auffuellen
2310 FOR A2=1TO37-ZL
2320 X=ASC("*"):GOSUB2200
2330 NEXT:ZL=0
2340 RETURN
3000 DATA27,76,48,48,51,27,84,52,51,27,88,13,+
3010 DATAein,zwei,drei,vier,fünf,sechs,sieben
3020 DATAacht,neun,+,zehn,zwanzig,dreißig
3030 DATAvierzig,fünfzig,sechzig,siebzig
3040 DATAachtzig,neunzig

```

Schreibmaschinen-Treiber von Wolfgang von Jan

Ein Treiberprogramm für ein Schreibmaschineninterface wurde schon vor langer Zeit im NASCOM Journal abgedruckt, allerdings für das T4 Betriebssystem. Hier nun ein Ansteuerprogramm für den Kanis-Drucker mit dem NASSYS Monitor.

Das eigentliche Treiberprogramm finden Sie im Bereich E00 - EFF. Ab 1000 liegt ein Eingabeprogramm, das es gestattet, gleich in NASPEN zu starten, mit einer Bildschirmausgabe, die es jedem ermöglicht, sofort mit der Texteingabe zu beginnen. Das Eingabeprogramm sollte im Generate Format abgespeichert werden (G-Kommando E 000 1120 B806 bei NASSYS 1). Vor dem Laden muß NASPEN kaltgestartet werden.

Das Treiberprogramm ist eine verbesserte Version. Einmal wird bei erneut gleichem Buchstaben eine Verzögerung vorgesehen (die Typenhebel verklemmen sonst manchmal bzw. der Typenhebel war noch nicht wieder so weit zurück, daß er erneut angesteuert werden konnte). Andererseits wurde die Shift-

Setzen/Shift-Löschen-Verzögerung so geändert, daß sie bei jedem Wechsel wirksam wird.

Im Vergleich zur Quelle/Olivetti-Schreibmaschine lt. MC 3/82 schafft unsere alte Typenhebelmaschine mit 8 Zeichen netto pro Sekunde deutlich mehr als die Typenradmaschine mit 5 Z/sec, die man lt. MC sicherheitshalber einstellen soll. Da bleibt nur noch der Vorteil der Wechselmöglichkeit des Typenrades... Meine Wahl damals war rückwirkend offensichtlich nicht falsch, sie würde heute noch genauso ausfallen. Ein sauber schreibender Matrixdrucker kommt für mich immer noch einfach zu teuer.

```

0E00 B7 C8 F8 C5 CD EF 0E 22 36
0E08 B6 0E 00 00 D7 60 FF 05 15
0E10 F5 00 FF 40 FF FF C6 F4 0A
0E18 CF 0E FF F5 C5 02 FF FF BC
0E20 FF FF FF FF FF FF 4E FF 75
0E28 FF FF FF C1 FF F5 F4 03 DF
0E30 30 0D 29 20 2B 04 18 2A 35
0E38 E4 C8 CC C4 C3 08 89 F1 C7
0E40 F0 EC 0E EB D8 EA D0 E9 70
0E48 28 31 18 2C 2A 0C E8 2E 3F
0E50 13 1B 25 21 1A 1D 12 16 31
0E58 15 0A 0D 0B 14 09 0E 32 FA
0E60 26 22 19 11 1C 2D 24 23 70
0E68 1E 05 02 01 04 10 C1 F5 66
0E70 FE 60 26 00 6F 3E 3F D3 C1
0E78 07 3C 38 03 AF CB AD 09 34
0E80 86 21 B8 0E 4F FE 35 20 9D
0E88 0E 06 FF 79 D3 05 FF 10 09
0E90 FD F1 C1 2A B6 0E C9 06 0A
0E98 30 6E 0F 4B 71 20 0F 30 6E
0EA0 EA CB 86 3E 36 06 58 D3 8E
0EA8 05 FF 10 FD 18 E9 38 DB DB
0EB0 CB C6 3E 33 18 EF FE 0E D3
0EB8 5E A5 C5 D5 E5 F5 DD E5 FF
0EC0 FD E5 F7 CD 00 0E 37 3F F8
0EC8 DF 61 38 0B 00 0E 11 DD 14
0ED0 E1 F1 E1 01 C1 00 C9 FE EA
0ED8 48 20 F2 FD E1 DD E1 F1 CD
0EE0 E1 D1 C1 00 00 C3 06 B8 E2
0EE8 DF 7B CD BA 0E 18 F9 21 17
0EF0 FE 0E BE 77 00 F5 06 15 0F
0EF8 FF 10 FD 00 F1 C9 02 21 EF

```

```

1000 00 04 3C 00 4E 00 01 00 9F
1008 01 06 7A 6F 7C DE 82 07 8B
1010 07 3C 01 3F 20 10 0A 08 E5
1018 20 10 16 11 40 C3 BA 0E 4A
1020 44 69 65 20 4B 61 73 73 F4
1028 65 74 74 65 20 69 73 74 5A
1030 20 6A 65 74 7A 74 20 72 23
1038 69 63 68 74 69 67 20 65 45
1040 69 6E 67 65 6C 65 73 65 9C
1048 6E 2E 0D 0D 44 69 65 73 93
1050 65 72 20 54 65 78 74 20 1C
1058 68 69 65 72 20 6D 75 73 85
1060 73 20 6E 75 6E 20 67 65 40
1068 6C 6F 65 73 63 68 74 20 8A
1070 77 65 72 64 65 6E 0D 75 87
1078 6E 64 20 64 61 6E 6E 20 3B
1080 6B 61 6E 6E 20 6E 65 75 A0
1088 65 72 20 54 65 78 74 20 54
1090 65 69 6E 67 65 67 65 62 D6
1098 65 6E 20 77 65 72 64 65 B2
10A0 6E 2E 0D 0D 5A 75 6D 20 C2
10A8 4C 6F 65 73 63 68 65 6E E9
10B0 20 27 4B 27 20 75 6E 64 E0

```

```

1088 20 64 61 6E 6E 20 27 59 29
10C0 27 20 65 69 6E 74 61 73 9B
10C8 74 65 6E 2E 0D 0D 41 6E 16
10D0 73 63 68 6C 69 65 73 73 3E
10D8 65 6E 64 20 6D 69 74 20 A9
10E0 27 41 27 20 6F 64 65 72 49
10E8 20 27 49 27 20 64 65 6E 06
10F0 20 6E 65 75 65 6E 0D 54 9C
10F8 65 73 74 20 65 69 6E 67 1C
1100 65 62 65 6E 2E 0D 0D 53 46
1108 74 61 6E 64 20 32 37 2E 77
1110 32 2E 38 32 0D 20 FF 20 37
1118 FF 00 FF 00 FF 00 FF 00 25

```

Tausche SPACE-INVADERS mit Ton,
 fuer alle Monitore geeignet,
 gegen BASIC-TOOLKIT auf cc.
 Cassette senden an

Constantin Olbrich

Seite für Kinder von Günter Böhm

Die Zeitschrift ELCOMP hat vor einiger Zeit einmal eine Serie für Kinder begonnen, die allerdings über eine Fortsetzung nicht hinauskam. Vielleicht können Sie als Leser dazu beitragen, daß unsere Serie sich über längere Zeit hält; denn mir scheint es wichtig, die Kinder sinnvoll an den Computer heranzuführen, das Interesse sollte nicht nur durch Abschlußspiele und ähnliches geweckt werden. Daß der Computer nicht negativ sein muß, haben wir, die wir uns laufend damit beschäftigen, wohl schon lange mitbekommen.

Auf folgendes Programm kam ich, als mich mein zehnjähriger Sohn bat, ihn das kleine Einmaleins abzuhören. Schnell war klar, daß ein Programm nicht nur das eingegebene Ergebnis testen mußte, sondern auch die benötigte Zeit berücksichtigen sollte. In Zusammenarbeit mit meinem Sohn entstand dann das folgende Programm, das er gerne als Spielzeug benutzt, und das gar keinen "Lern-Beigeschmack" für ihn besitzt. Hoffentlich folgen von Ihnen noch weitere Programme, die Kindern beim Lernen helfen und zudem den Umgang mit Rechnern fördern.

```

1 S=0:N=0:D=0:REM SPIELE,NASCOM,DU
2 GOSUB1100
3 PRINT "Zum Spielbeginn ENTER drücken.":INPUTX
4 Z=0:Y=0
5 PRINT:B=INT(RND(1)*10)
6 IFB=0THEN5
7 IFB=1THEN5
8 A=INT(RND (1)*10)
9 IFA=0THEN8
10 IFA=1THEN8
20 Y=Y+1:IFY=11THEN100
30 PRINT "WAS IST ";A;" MAL ";B;" ?"
40 GOSUB 1000:IFA$=CHR$(0)THEN10
50 IF A*B=CTHENPRINT"RICHTIG !!!!":Z=Z+1:GOTO5
55 PRINT"Das ist leider falsch!
56 PRINT"Merke dir:":PRINT
60 PRINTA;" MAL ";B;" IST ";A*B
70 PRINT:PRINT:PRINT: GOSUB1003:GOTO10
100 IFZ=10THEN CLS:PRINT"ALLE RICHTIG!!!!!!!!!"
101 IFZ=10THEND=D+1:GOTO105
103 CLS:PRINT"DU HAST NUR ";Z;" RICHTIGE VON 10
!
104 N=N+1
105 S=S+1:PRINT:PRINTS;". Spiel"
106 PRINT"NASCOM hat ";N;" Punkte
107 PRINT"Du hast ";D;" Punkte
108 IFS->10THENPRINT"ENTER":INPUTE$:GOTO2
109 PRINT"Sag' deinem Vater, er soll jetzt
110 PRINT"endlich den Kasten ausschalten!
111 PRINT"Du hast für heute genug gespielt.
112 INPUTE$
113 IFE$="N"THENEND
114 GOTO1
1000 FORI=1TO400 :A$=CHR$(USR(0)):REM *ZEIT-
1001 IFA$->CHR$(0)THEN1004 :REM SCHLEIFE*
1002 NEXT:PRINT:PRINT"Du bist noch zu langsam !
1003 PRINT"Versuche es weiter";:PRINT:GOTO1010
1004 PRINTA$;
1005 B$=CHR$(USR(0)):IFB$=CHR$(0)THEN1005
1006 PRINTB$:C=VAL(A$)*10+VAL(B$)
1007 IFB$=CHR$(13)THENC=VAL(A$):RETURN
1008 X=USR(0):IFX=0THEN1008
1009 RETURN
1010 PRINT" mit ENTER !
1011 X=USR(0):IFX=0THEN1011
1012 IFX->13THEN1011
1013 RETURN
1100 CLS
1101 SCREEN10,16:PRINT"Kleines Einmaleins-Spiel
1102 DOKE4100,3200: REM Tastaturabfrage
1103 FORI=3200TO3208 STEP2
1104 READ J:DOKE I,J:NEXT:RESTORE
1105 DATA 25311, 312, 18351, -15441, -3854
1106 RETURN

```

BLS-PASCAL

von Michael Bach

Auf meinen Artikel im letzten Journal hin haben eine Reihe von Lesern mit mir Kontakt aufgenommen und ihr Interesse an diesem Pascal-System für den Nascom bekundet. Deshalb hier ein weiteres Programm. Ich hoffe, daß der Text diesmal die Redaktion unbeschadet übersteht (d.h. das Einrücken sollte bleiben). (Deshalb habe ich extra mein Formatierprogramm kaputtgemacht. Ich dachte mir gleich, daß Sie wegen des Formats sauer sind, dabei hat es so schön ausgesehen! Ihr Nascomp!). Viele haben mich auch gefragt, ob ich das Pascal verkaufe oder tausche. Das habe ich abgelehnt, weil ich finde, daß die Entwickler dieses Systems gute Arbeit geleistet haben und auch dafür was erhalten sollten (Kopieren käuflicher Software ist ja sowieso ein heikles Thema). Für diejenigen, die es bestellen wollen, gebe ich hier die Adresse an, das hatte ich beim letzten Mal vergessen:

Electrovalue Ltd.,
700 Burnage Lane,
Burnage,
Manchester M19 1NA
England.

```
PROGRAM MALEDRACHEN; (* 05.06.82
  Demonstration eines rekursiven Algorithmus, malt
  eine hübsche unerwartete Figur. Nach "Problem
  Solving Using Pascal" von K.Bowles.
  Von Michael Bach, [REDACTED], [REDACTED] Stegen,
  [REDACTED].*)
```

```
CONST KARO=3;
VAR XTURTLE, YTURTLE, WTURTLE, GROESSE: INTEGER;
```

```
(*Zunächst 3 vereinfachte "TURTLE"-Prozeduren,
  kann nur rechte Winkel*)
```

```
PROCEDURE TURN(WINKEL: INTEGER);
BEGIN WTURTLE:=(WTURTLE+WINKEL) MOD 360 END;
```

```
PROCEDURE MOVE(L: INTEGER);
VAR A, I: INTEGER;
BEGIN
  CASE WTURTLE OF
    0: BEGIN
      A:=XTURTLE; XTURTLE:=A+L;
      FOR I:=A TO XTURTLE DO PLOT(I, YTURTLE, 1);
    END;
```

```
90: BEGIN
  A:=YTURTLE; YTURTLE:=A-L;
  FOR I:=A DOWNTO YTURTLE DO PLOT(XTURTLE, I, 1);
END;
180: BEGIN
  A:=XTURTLE; XTURTLE:=A-L;
  FOR I:=A DOWNTO XTURTLE DO PLOT(I, YTURTLE, 1);
END;
270: BEGIN
  A:=YTURTLE; YTURTLE:=A+L;
  FOR I:=A TO YTURTLE DO PLOT(XTURTLE, I, 1);
END; (*CASE*)
END; (*MOVE*)

PROCEDURE CLEARSCREEN;
BEGIN
  WRITE(CHR(12)); XTURTLE:=47; YTURTLE:=24; WTURTLE:=0;
END;

PROCEDURE DRACHEN(L: INTEGER);
BEGIN
  IF L=0 THEN MOVE(KARO)
  ELSE
    IF L<0 THEN BEGIN
      DRACHEN(L-1); TURN(90); DRACHEN(-L+1);
    END ELSE BEGIN
      DRACHEN(-L-1); TURN(270); DRACHEN(L+1);
    END;
  END; (*DRACHEN*)
BEGIN (*HP*)
  CLEARSCREEN;
  REPEAT
    WRITE(' Drachengröße (0-8)? '); READLN(GROESSE);
    IF GROESSE=0 THEN BEGIN
      CLEARSCREEN; XTURTLE:=28; YTURTLE:=20;
      DRACHEN(GROESSE);
    END;
  UNTIL GROESSE<1;
  WRITE(' Auf Wiedersehen!');
END.
```

EPROM-Karte (2716) 2X16K umschaltbar,
30K bestückt mit 8K-BASIC, 5K-Text-
editor, ZEAP 2-Assembler, Disassembler,
Debug, Eprommer, Relocator, Speicher-
vergleich, Menue und anderen Programmen.
DM 400.-
NASSYS 3 DM 40.-
Hans Schneider Tel. [REDACTED]
[REDACTED]
[REDACTED]

ROBOT-Intelligence

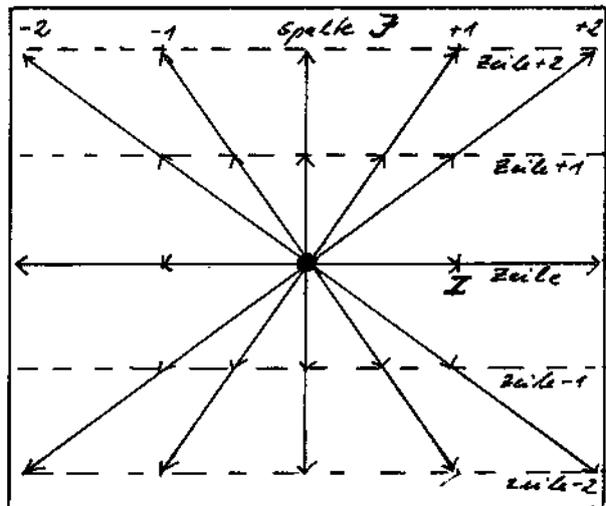
von Günter Böhm

Da ich mir jede Veröffentlichung über Roboter anschaffe, kam mir nun auch das Buch "ROBOT INTELLIGENCE with experiments" von David L. Heiserman unter die Finger (TAB BOOKS INC vertrieben von Hofacker Verlag). Das Buch unterscheidet sich von anderen Werken des Authors dadurch, daß es sich nicht mit dem Bau und der Programmierung von beweglichen Maschinen beschäftigt, sondern mit der Simulation von "Maschinenintelligenz" am Bildschirm eines Rechners. Dies macht für einen größeren Leserkreis interessant. Die vielen Beispielprogramme sind für den TRS 80 geschrieben, lassen sich aber, wie die abgedruckten Beispiele zeigen, relativ leicht an den NASCOM anpassen. Heiserman sieht die Haupteigenschaft eines Roboters in der Fähigkeit, sich neuen Situationen anzupassen. So ist für ihn ein sogenannter Industrie-Roboter kein wirklicher Roboter, da er nur vorprogrammierte Schritte ausführt und nicht umlernen kann, wenn neue Situationen auftauchen. Sein Ziel ist die Konstruktion einer Maschine, die sich selbst programmiert.

Dieses Ziel will er durch die Entwicklung eines "Programmes" erreichen, das er EAMI (Evolutionary Adaptive Machine Intelligence) nennt, also ein Programm, das sich entwickelt und anpaßt. Die Entwicklung des Programmes (die noch lange nicht abgeschlossen ist, und zu deren Weiterführung der Leser aufgefordert wird - Sie kennen so etwas ja aus dem Journal), wird sehr anschaulich beschrieben (leider alles in Englisch) und an Umengen von Programmen verdeutlicht.

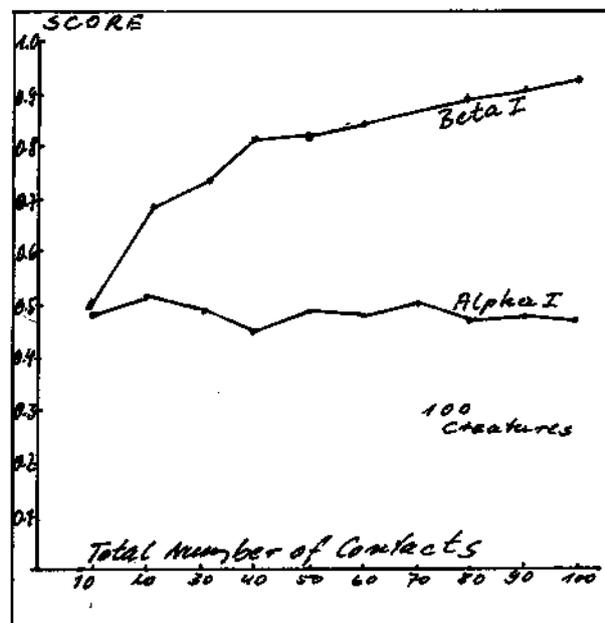
Hier ein kurzer Abriss des Aufbaus:

Heiserman teilt seine "Intelligenzen" in Alpha-, Beta- und Gamma-Wesen ein, jeweils Level (Niveau) I oder II. Die Alphawesen existieren nur in der Gegenwart und reagieren rein zufällig. Auf dem Bildschirm wird das simuliert, indem ein "Wesen" (in unserem Beispiel das Bell-Zeichen) sich innerhalb eines rechteckigen "Spielfeldes" bewegt. Würde bei der nächsten Bewegung der Feldrand berührt (Contact), so muß eine Bewegungsrichtung gefunden werden, die dies vermeidet. Aus 24 möglichen Richtungen wird per



Zufall eine ausgewählt. Über dem Spielfeld werden die Kontakte, die erfolgreichen Bewegungen (vom Rand weg) und deren Verhältnis (Score) angezeigt. Wieoft man auch die Simulation durch Drücken der D-Taste neu startet (D=disturb=stören), das Alpha-Wesen wird immer eine erfolgsquote von 0,5 und knapp darunter erreichen.

Das Beta-Wesen merkt sich die erfolgreichen Bewegungen und setzt sie in gleichen Situationen wieder ein. Es "lebt" also in Gegenwart und Vergangenheit. Es beginnt seine Bewegung auch erst rein zufällig, nach gewisser Zeit bemerkt man aber, daß es sich ein bestimmtes "Verhaltensmuster" aneignet; seine Bewegungen verlaufen nach dem gleichen



Schema, und die Erfolgspunkte bei "Score" steigen. Wird der Ablauf durch D gestört, so bleibt das "Gedächtnis" erhalten, und sehr rasch wird das Verhaltensmuster wieder aufgenommen. Die Erfolgsquote kann so leicht 1,0 erreichen. Interessant ist, daß jedes Beta-Wesen (d.h. nach Reset und Neustart des Programms) sich andere Muster angewöhnt. Das Gamma-Wesen besitzt als zusätzliche Eigenschaft die Fähigkeit, auch "in die Zukunft zu blicken", d.h. es merkt sich ebenfalls seine erfolgreichen Züge, verallgemeinert sie aber und versucht, sie auf zukünftige neue Situationen anzuwenden. Dazu wird ein gesonderter Speicher angelegt, der den Inhalt einer entsprechenden Adresse um 1 erhöht, wenn ein erfolgreicher Zug gemacht wurde. Dadurch steigt das "Erfolgserlebnis" bis es bei der Zahl 4 dazu führt, den erfolgreichen Zug zu speichern und auf ähnliche Kontakte anzuwenden. Bei Mißerfolgen wird der "Erfolgsspeicher" erniedrigt, und der entsprechende Zug vergessen. Die einzelnen Routinen sind im Buch sehr genau erklärt und würden den Rahmen dieses Heftes sprengen. Die Aktionen des Gamma-Wesens waren für mich aber lang nicht so überzeugend wie die Lernfähigkeit der Beta-Ausgabe. Vielleicht ist mir da beim Umsetzen des TRS 80 Programms ein Fehler unterlaufen. Man kann die Programme aber nicht nur zum Experimentieren einsetzen (es sind sogar Feldversuche über 24 Stunden abgedruckt), sondern als recht amüsante Spiele für mehrere Mitspieler anwenden. So kann ein Spieler beim Spiel "Irrgarten" die Zeit stoppen, die "sein" Wesen benötigt, um vom Ausgangspunkt das Ziel ZZ zu erreichen. (Wohl gemerkt: es ist nicht auf das Erreichen dieses Ziels programmiert und läuft auch gerne rückwärts!) Beim Spiel "Einschließen" (Es sind nur jeweils die entsprechenden Zeilen zu ändern oder hinzuzufügen) kann man stoppen, wie lange das Wesen, das nun eine Spur hinterläßt, braucht, um sich selbst den Weg abzuschneiden. Vielleicht finden Sie noch andere Möglichkeiten, die "Wesen" im Spiel einzusetzen. Die abgedruckten Programme erkennen nur Space (Weg frei) und Nicht-Space (Kontakt). Dies wird vom Author als Level I bezeichnet. Die Level II - Wesen können verschiedene Hindernisse unterscheiden. Dadurch werden

Ihre Bewegungen natürlich interessanter, besonders der Versuch, nach Störungen das alte Muster wieder aufzunehmen. Wenn Sie daran oder an ausführlicherer Information über das gesamte System interessiert sind, empfehle ich, das Buch anzuschaffen. Wer hat sich schon mit ähnlichen Programmen befaßt? Vielleicht können Sie weiteren Stoff liefern zum Thema "Maschinenintelligenz". Viele Leser würden sich sicher freuen.

```

1 REM** ALPHA-IG MASTER **
2 DOKE4100,3200:FORI=3200TO3208STEP2
3 READJ:DOKEI,J:NEXT:REM *INKEY$=USR(0)**
4 DATA25311,312,18351,-15441,-3854
10 REM ** ALPHA-IG MASTER **
15 CLS:SCREEN1,16:PRINT"CONTACTS: ";
16 PRINT"GOOD MOVES: SCORE: "
17 GOSUB550:GOSUB600
20 GOSUB725:IFC0=32THEN35
25 C=C+1:GOSUB650:I=RI-2:J=RJ-2:
26 GOSUB725:IFC0=32THEN25
30 D=D+1:GOSUB800
35 GOSUB400:IFUSR(0)=68THEN10
36 GOTO20
400 REM ** MOVE II-1 **
405 POKENP,32:NP=NP+I+64*J:POKENP,07:RETURN
550 REM ** FIELD II-1 **
555 FORF=0TO95:SET(F,1):NEXT
560 FORF=0TO95:SET(F,42):NEXT
565 FORF=1TO42:SET(0,F):NEXT
570 FORF=1TO42:SET(95,F):NEXT
575 RETURN
600 REM ** INITIAL IIG-1 **
605 NP=2192:GOSUB650:I=RI-2:J=RJ-2:C=0:D=0
606 RETURN
650 REM ** FETCH NEW IIG-1 **
655 RI=INT(RND(1)*10):IFRI=4THEN655
656 RJ=INT(RND(1)*10):IFRJ=4THEN656
657 IFRI=2ANDRJ=2THEN655
660 RETURN
725 REM ** CON SENSE II-1 **
730 FORSI=ABS(I)TO1STEP-1
731 FORSJ=ABS(J)TO1STEP-1
735 CO=PEEK(NP+SI*SGN(I)+64*SJ*SGN(J))
740 IFC0=32THENRETURN
745 NEXTSJ,SI:RETURN
800 REM***UD SCORE IIG-1***
801 AA=3018 : REM TOPLINE
802 A=AA+9:X=C:GOSUB1000
803 A=AA+25:X=D:GOSUB1000
804 A=AA+36:X=D/C:GOSUB1000
805 RETURN

```

```

1000 X$=STR$(X)
1010 FORY=1TOLEN(X$):POKEA,ASC(MID$(X$,Y,1))
1020 A=A+1:NEXT
1030 RETURN

```

ALPHA I " Irrgarten "

```

26 GOSUB725:IFCO=90THEN30
27 IFCO<=32THEN25

```

```

405 POKENP,32:NP=NP+I+64*J:
406 IFPEEK(NP)=90THEN408
407 POKENP,007:RETURN
408 PRINT"GAME OVER":END

```

550 REM ** FIELD II-1 MAZE **

```

555 FORF=0T095:SET(F,1):NEXT
560 FORF=0T095:SET(F,42):NEXT
565 FORF=1T042:SET(0,F):NEXT
570 FORF=1T042:SET(95,F):NEXT
571 FORF=1T070:SET(F,10):NEXT
572 FORF=25T095:SET(F,20):NEXT
573 FORF=1T070:SET(F,30):NEXT
574 POKE2891,ASC("Z"):POKE2892,ASC("Z")
575 RETURN

```

ALPHA I " Einschließen "

```

405 POKENP,42:NP=NP+I+64*J:
-----

```

10 REM ** BETA-IG MASTER **

```

11 DIMM(4,4,2)
15 CLS:SCREEN1,16:PRINT"CONTACTS: ";
16 PRINT"GOOD MOVES: SCORE: "
20 GOSUB550:GOSUB600
25 GOSUB725:IFCO=32THEN45
30 C=C+1:MI=I+2:MJ=J+2:RI=M(MI,MJ,1)
31 RJ=M(MI,MJ,2):KL=M(MI,MJ,0):IFKL=0THEN50
35 I=RI-2:J=RJ-2:GOSUB725:IFCO<=32THEN55
40 D=D+1:GOSUB800
45 GOSUB400:IFUSR(0)=68THEN15
46 GOT025
50 GOSUB650:M(MI,MJ,1)=RI:M(MI,MJ,2)=RJ
51 M(MI,MJ,0)=1:GOTO35
55 C=C+1:GOTO50

```

10 REM ** GAMMA-IG MASTER **

```

11 DIMM(4,4,2):DIMG(7,1):DIMK(7)
15 CLS:SCREEN1,16:PRINT"CONTACTS:
16 PRINT"GOOD MOVES: SCORE: "
20 GOSUB550:GOSUB600
25 GOSUB725:IFCO=32THEN35
30 C=C+1:GOSUB200:GOSUB800:IFGF=1THEN40
35 GOSUB400:IFUSR(0)=68THEN15
36 GOT025
40 GOSUB100:GOTO35

```

100 REM ** GAMMA-GEN G1 **

```

105 POKE2123,71:FORK=0T07:K(K)=0:FORL=0T01
106 G(K,L)=0:NEXTL,K:FORMI=0T04:FORMJ=0T04
107 GOSUB175:IFM(MI,MJ,0)=4THEN115
115 NEXTMJ,MI

```

```

120 FORGP=0T07:FORGQ=0T01
125 IFK(GP)=0THEN145
130 G(GP,GQ)=G(GP,GQ)/K(GP)
135 IFG(GP,GQ)<2G(GP,GQ)=1
140 IFG(GP,GQ)>2G(GP,GQ)=3
145 NEXTGQ,GP

```

```

150 FORMI=0T04:FORMJ=0T04:GOSUB175
155 IFM(MI,MH,0)=1THEN165
160 M(MI,MJ,1)=G(GP,0):M(MI,MJ,2)=G(GP,1)
161 M(MI,MJ,0)=1:POKE2125,42:POKE2125,32
165 NEXTMJ,MI
170 POKE2123,32:GF=0:RETURN
175 REM ** GAMMA PHASE-2 **
180 IFMI<2ANDMJ<2THENG=0
182 IFMI<2ANDMJ=2THENG=1
184 IFMI<2ANDMJ>2THENG=2
186 IFMI=2ANDMJ<2THENG=3
188 IFMI=2ANDMJ>2THENG=4
190 IFMI>2ANDMJ<2THENG=5
192 IFMI>2ANDMJ=2THENG=6
194 IFMI>2ANDMJ>2THENG=7
196 RETURN
200 REM ** GAMMA CONL G-1 **
205 MI=I+2:MJ=J+2:RI=M(MI,MJ,1):RJ=M(MI,MJ,2)
206 KL=M(MI,MJ,0):IFKL=0THEN235
210 I=RI-2:J=RJ-2:GOSUB725:IFCO=32THEN240
215 C=C+1:IFKL=0THEN235
220 KL=KL-1:IFKL=0THEN235
225 IFKL=3THEN230
226 GOT0210
230 GF=1:GOTO210
235 GOSUB650:KL=1:GOTO210
240 IFKL<4THEN245
241 GOT0255
245 KL=KL+1:IFKL=4THEN250
246 GOT0255
250 GF=1
255 M(MI,MJ,1)=RI:M(MI,MJ,2)=RJ
256 M(MI,MJ,0)=KL:D=D+1:RETURN

```

Software für VIDEO-Interface

von A. Zippel u. D. Oberle

Wie bereits in Heft 5/82 angekündigt hier die Treibersoftware zur dort vorgestellten Video-Interfacekarte. Doch zunächst ein paar Hinweise zur Programm Benutzung:

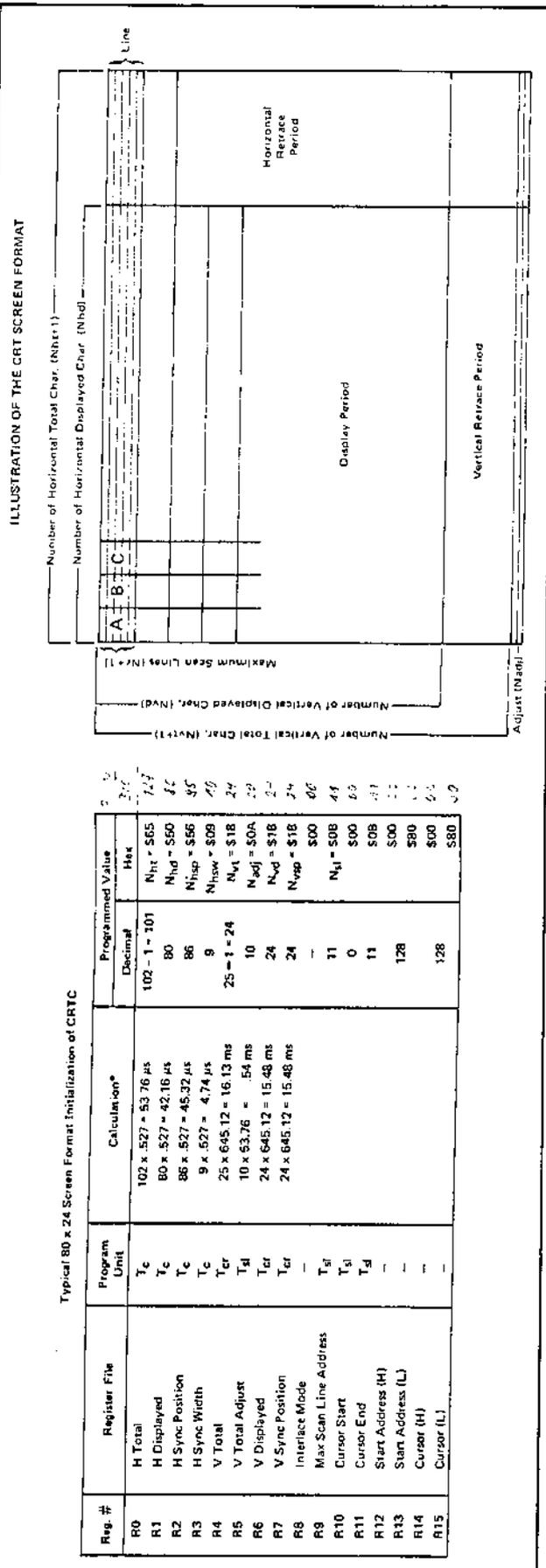
Da das Programmlisting recht gut mit Kommentaren ausgestattet ist spare ich mir hier an dieser Stelle eine detailliertere Programm Beschreibung. Die Kommentare sind leider (oder "Gott sei Dank") in Englisch. Ich hoffe dennoch, daß die meisten Leser damit zurechtkommen. Es macht sehr viel Arbeit alles nochmal deutsch zu kommentieren.

Das Hauptprogramm wird durch die Routine CONOUT (Console Output) dargestellt. Der auszugebende Charakter muß dieser Routine in Register C übergeben werden, d.h. das darzustellende Zeichen muß vor dem Aufruf von CONOUT immer in Register C gespeichert sein. (die NASSYS Routine verwendet dazu Register A). Um irgendwelche Fehlfunktionen des Programmes zu vermeiden sollten außerdem vor dem Aufruf von CONOUT alle Register gesichert werden. Für die Benutzung mit NASSYS I kann eine solche Ausgaberroutine z.B. so aussehen:

```

USEROU:  PUSH AF
         PUSH HL
         PUSH DE
         PUSH BC
         LD C,A
         CALL CONOUT
         POP BC
         POP DE
         POP HL
         POP AF
         RET
    
```

dabei ist vorher in der Sprungtabelle von NASSYS I als USER-Output-Routine die Adresse von USEROU einzutragen, sodaß mit dem "U" Befehl jederzeit die Ausgabe auch auf den 80-Zeichen-Bildschirm umgesteuert werden kann, natürlich auch nur im 48 Charakter/line Format wenn keine Änderungen im Monitorprogramm vorgenommen werden.




```

0001 ;-----
0002 ; THE CRT-CONTROLLER-CARD WILL BE INITIALISED AT THIS POINT
0003 ; THIS CODE EXECUTES ONLY ONCE, BEFORE "CONOUT" IS USED
0004 ;-----
0005 ;
0006 INIT$CRT:
0000 212400 0007 LD HL,MODES ; POINT TO MODE-LIST
0003 0610 0008 LD B,10H ; 16 BYTES TO SEND
0005 0E21 0009 LD C,CRTR
0007 1600 0010 LD D,0 ; START WITH REGISTER 0
0009 7A 0011 INITL: LD A,D
000A D320 0012 OUT (CRTR),A ; TELL CRT THE ADDRESS OF REGISTER
000C 14 0013 INC D
000D EDA3 0014 OUTI ; SEND BYTE INTO REGISTER
000F 20F8 0015 JR NZ,INITL ; AND LOOP UNTIL READY
0011 0E11 0016 LD C,11H ; TO HAVE AN INITIAL STATE
0013 CD5700 0017 CALL CONOUT ; PERFORM A CLEAR-SCREEN AND RETURN
0016 2100F8 0018 LD HL,CRTMEM
0019 1101F8 0019 LD DE,CRTMEM+1
001C 018007 0020 LD BC,24*80
001F 3620 0021 LD (HL),' '
0021 EDB0 0022 LDIR
0023 C9 0023 RET
0024 ;
0024 7F505F0A 0025 MODES: DEFB 127,80,95,10,24,20,24,24,0,11,00,11,0,0,0,0,0,0
18141818
C000000B
00000000
00000000

0026 ;
0027 ;
0028 ;-----
0029 ;
0030 ; CONSOLE DISPLAY ROUTINE
0031 ;
0032 ;-----
0033 ;
0034 CRTF: EQU 020H ; ADDRESS OF FILE-REGISTER
0035 CRTR: EQU 021H ; ADDRESS OF DATA-PORT
0036 CRTMEM: EQU 0F800H ; BASE-ADDRESS FOR CRT-MEM
0037 CRTEND: EQU CRTMEM+(24*80); END OF ADDRESSABLE VIDEO-RAM
0038 CRTBAS: EQU CRTMEM SHR 8 ; STARTING PAGE OF MEM
0039 CRTTOP: EQU (CRTMEM+2048) SHR 8; TOP-PAGE OF VIDEO-RAM
0040 ;
0037 00000000 0041 DEFB 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
00000000
00000000
00000000
0047 00000000 0042 DEFB 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0; AREA FOR STACK
00000000
00000000
00000000
(0057)
0043 TEMSTK: EQU $
0044 ;
0045 ; THIS IS THE ENTRY TO THE CRT-DRIVER ROUTINE.
0046 ; THE CHARACTER PASSED IN REGISTER C WILL BE PRINTED
0047 ; ON THE CRT-CARD CONECTED VIDEO-MONITOR.

0048 ;
0057 F3 0049 CONOUT: DI ; INTERRUPTS ARE NOW DANGEROUS
0058 ED730902 0050 LD (SPSAVE),SP ; SAVE STACK OVER CRT-PROCESSING
005C 315700 0051 LD SP,TEMSTK ; GET NEW STACK AT SAVE AREA
005F E5 0052 PUSH HL ; SAVE REGISTER OVER OPERATION
0060 D5 0053 PUSH DE
0061 C5 0054 PUSH BC
0062 79 0055 LD A,C
0063 E67F 0056 AND 7FH
0065 4F 0057 LD C,A
0058 CRT1:
0066 2A0602 0059 LD HL,(CURSOR) ;GET CURSOR
0069 7C 0060 LD A,H
006A E60F 0061 AND 0001111B ;BE SURE THAT THE CURSOR IS IN RANGE
006C F6F8 0062 OR CRTBAS
006E 67 0063 LD H,A
0064 ;
0065 ; PROCESS THE CHARACTER PASSED IN ACCU AND C-REGISTER
0066 ;
006F CDA300 0067 CALL OUTCH
0068 ;
0069 ; ENSURE THAT CURSOR IS IN FRAME OF VIDEO RAM
0070 ;

```

```

0072 E5          0071 CRLOP:  PUSH   HL
0073 1180FF     0072          LD     DE,CRTEND
0076 A7         0073          AND    A
0077 ED52       0074          SBC   HL,DE
0079 E1         0075          POP   HL
007A 3807       0076          JR    C,CRTUP
007C 115000     0077          LD     DE,80
007F ED52       0078          SBC   HL,DE
0081 18EF       0079          JR    CRLOP
                0080 ;
                0081 ; NOW SAVE THE CURSOR-POINTER IN HL AND SET THE CURSOR
                0082 ; REGISTER IN 6845
                0083 ;
0083 220602     0084 CRTUP:  LD     (CURSOR),HL
0086 1100F8     0085          LD     DE,CRTMEM
0089 A7         0086          AND    A
008A ED52       0087          SBC   HL,DE
008C 3E0E       0088          LD     A,14D
008E D320       0089          OUT   (CRTF),A ;ADDRESS THE REG 14 IN CRT
0090 7C         0090          LD     A,H
0091 D321       0091          OUT   (CRTR),A
0093 3E0F       0092          LD     A,15D
0095 D320       0093          OUT   (CRTF),A
0097 7D         0094          LD     A,L
0098 D321       0095          OUT   (CRTR),A
009A C1         0096          POP   BC ; RESTORE REGISTER
009B D1         0097          POP   DE
009C E1         0098          POP   HL
009D ED7B0902   0099          LD     SP,(SPSAVE) ; GET ORIGINAL CURSOR BACK
00A1 FB         0100          EI ; INTERRUPTS ARE SAVE AGAIN
00A2 C9         0101          RET
                0102 ;
                0103 ; DO OUTPUT TO THE SCREEN
                0104 ;
00A3 110502     0105 OUTCH: LD     DE,LEADIN
00A6 1A         0106          LD     A,(DE) ;GET ESCAPE-SEQUENCE STATE
00A7 B7         0107          OR    A
00A8 C2BE01     0108          JP    NZ,MULTI ;JUMP IF IN ESCAPE-MODE
00AB 79         0109          LD     A,C ;PROCESS CHARACTER IN C
00AC FE20       0110          CP    20H ;TEST FOR CONTROL-CHAR
00AE 380D       0111          JR    C,CONTRL ;YES IT IS A CONTROL-CHARACTER
00B0 71         0112 DISPLA: LD     (HL),C ;DISPLAY THE CHARACTER ON SCREE
00B1 23         0113          INC   HL ;POINT TO NEXT
00B2 1180FF     0114          LD     DE,CRTEND
00B5 CD1701     0115          CALL  HILO ;TEST IF CURSOR OUT OF RAM
00B8 D0         0116          RET ;CURSOR IN RANGE
00B9 CD9801     0117          CALL  LFEED ;MOVE SCREEN ONE LINE UP
00BC C9         0118          RET
                0119 ;
                0120 ; PROCESS ALL CONTROL CHARACTER'S
                0121 ;
00BD E5         0122 CONTRL: PUSH  HL
00BE 21D700     0123          LD     HL,CTLTAB ;POINT TO TABLE OF CONTRL-CH.
00C1 010E00     0124          LD     BC,CTLSIZ/3 ;LENGTH OF TABLE
00C4 CDCB00     0125          CALL  SEARCH ;SEARCH A CHAR IN TABLE
00C7 E1         0126          POP   HL ;GET CURSOR BACK
00C8 C0         0127          RET   NZ ;NOT FOUND
00C9 C5         0128          PUSH  BC ;ADDRESS OF CORRESPNDENT ROUTINE ON STACK
00CA C9         0129          RET
                0130 ;
                0131 ; SEARCH A CONTROL CHARACTER IN THE LIST, IF FOUND THE ADDRESS OF
                0132 ; THE CORRESPONDING ROUTINE IS IN BC AND THE ZERO-FLAG IS SET
                0133 ; ELSE THE ZERO-FLAG IS RESET
                0134 ;
00CB EDB1       0135 SEARCH: CPIR ;SEARCH IN LIST
00CD C0         0136          RET   NZ ;NOT FOUND
00CE F5         0137          PUSH  AF ;SAVE FLAG
00CF 09         0138          ADD   HL,BC
00D0 09         0139          ADD   HL,BC
00D1 09         0140          ADD   HL,BC ;POINT TO ADDRESS OF ROUTINE
00D2 4E         0141          LD     C,(HL)
00D3 23         0142          INC   HL
00D4 46         0143          LD     B,(HL) ;BC POINT'S TO ROUTINE
00D5 F1         0144          POP   AF ;GET FLAG'S BACK
00D6 C9         0145          RET
                0146 ;
00D7 1F         0147 CTLTAB: DEFB 1FH
00D8 1E         0148          DEFB 1EH
00D9 1B         0149          DEFB 1BH
00DA 1A         0150          DEFB 'Z'-64

```

```

00DB 18      0151      DEFB      'X'-64
00DC 11      0152      DEFB      'O'-64
00DD 0D      0153      DEFB      'M'-64
00DE 0C      0154      DEFB      'L'-64
00DF 0B      0155      DEFB      'K'-64
00E0 0A      0156      DEFB      'J'-64
00E1 09      0157      DEFB      'I'-64
00E2 08      0158      DEFB      'H'-64
00E3 07      0159      DEFB      'G'-64
00E4 17      0160      DEFB      'W'-64
00E5 6601    0161      DEFW      CLREOS      ;CNTL-W IS CLEAR TO END OF SCREEN
00E7 1601    0162      DEFW      BELL        ;CNTL-G IS THE BELL
00E9 0901    0163      DEFW      BAKSPC     ;CNTL-H IS CURSOR LEFT
00EB 0D01    0164      DEFW      TAB        ;CNTL-I IS TAB
00ED 9801    0165      DEFW      LFEEED     ;CNTL-J IS CURSOR DOWN
00EF 7601    0166      DEFW      UPSCR      ;CNTL-K IS CURSOR UP
00F1 0B01    0167      DEFW      FORSPC     ;CNTL-L IS CURSOR RIGHT
00F3 1E01    0168      DEFW      RETURN     ;CNTL-M IS CR WITH CLRS
00F5 2E01    0169      DEFW      CLRSCN     ;CNTL-Q IS CLEAR SCREEN
00F7 4301    0170      DEFW      CLREOL     ;CNTL-X IS CLEAR TO END-OF-LINE
00F9 AD01    0171      DEFW      CLRLIN     ;CNTL-Z IS CLEAR LINE
00FB 0101    0172      DEFW      ESCAPE     ;CNTL-[ IS ESCAPE
00FD B901    0173      DEFW      HOMEUP     ;CNTL- IS HOME UP
00FF 0501    0174      DEFW      STUFF      ;CNTL- IS DISPLAY CNTL-CHAR'S
      (002A)    0175      CTLSIZ: EQU      $-CTLTAB
      0176      ;
      0177      ; LET US ENTER IN THE ESCAPE MODE
      0178      ;
0101 3E01    0179      ESCAPE: LD      A,1      ;SET ESCPAE MODE
0103 12      0180      LD      (DE),A
0104 C9      0181      RET
      0182      ;
      0183      ; IF YOU WANT TO DISPLAY A CHAR DIRECT, YOU MUST USE CNTL-UNDERLINE
      0184      ;
0105 3E04    0185      STUFF: LD      A,4
0107 12      0186      LD      (DE),A
0108 C9      0187      RET
      0188      ;
      0189      ; CURSOR ONE POSITION TO THE LEFT
      0190      ;
0109 2B      0191      BAKSPC: DEC     HL
010A C9      0192      RET
      0193      ;
      0194      ; CURSOR ONE POSITION TO THE RIGHT
      0195      ;
010B 23      0196      FORSPC: INC     HL
010C C9      0197      RET
      0198      ;
      0199      ; SET TABULATOR AT EVERY MODULO 8 POSITION
      0200      ;
010D 7D      0201      TAB: LD      A,L
010E E6F8    0202      AND     11111000B
0110 6F      0203      LD      L,A      ;POINT AT LAST TAB-POSITON
0111 110800  0204      LD      DE,8
0114 19      0205      ADD     HL,DE    ;POINT TO NEXT
0115 C9      0206      RET
      0207      ;
      0208      ;
0116 C9      0209      BELL: RET      ;SILENT ROUTINE FOR NOW
      0210      ;
      0211      ;
      0212      ;
      0213      ; TEST IF HL >= DE FLAGGED WITH CARRY
      0214      ;
0117 A7      0215      HILO: AND     A      ;RESET CARRY
0118 E5      0216      PUSH    HL
0119 ED52    0217      SBC     HL,DE
011F E1      0218      POP     HL
      0219      ;
011C 3F      0219      CCF
011D C9      0220      RET      ;CARRY=1 IF HL>DE
      0221      ;
      0222      ; PERFORM A CARRIAGE-RETURN WITH DELETING THE REST OF LINE
      0223      ;
011E 00      0224      RETURN: NOP
      0225      ;
      0226      ; COMPUTE FROM (ROW)THE START-ADDRESS OF THE CURRENT LINE
      0227      ;
      0228      ;
011F 3A0802  0229      CPOINT: LD     A,(ROW) ;GET ROW ADDRESS
0122 47      0230      LD     B,A

```

```

0123 04          0231      INC      B
0124 21B0F7     0232      LD       HL,CRTMEM-80
0127 115000     0233      LD       DE,80
012A 19         0234 CPNTL:  ADD      HL,DE
012B 10FD      0235      DJNZ    CPNTL
012D C9         0236      RET
                0237 ;
                0238 ; CLEAR THE SCREEN AND HOME THE CURSOR
                0239 ;
012E 2100F8     0240 CLRSCN: LD      HL,CRTMEM
0131 E5         0241      PUSH    HL
0132 1101F8     0242      LD       DE,CRTMEM+1
0135 017F07     0243      LD       BC,24*80-1
0138 3620      0244      LD       (HL),' '
013A EDB0      0245      LDIR                    ;CLEAR SCRENN
013C E1         0246      POP     HL
013D 3E00      0247      LD       A,0              ;SET NEW ROW-ADDRESS
013F 320802     0248      LD       (ROW),A
0142 C9         0249      RET
                0250 ;
                0251 ; CLEAR LINE FROM CURSOR TO END OF LINE
                0252 ;
0143 E5         0253 CLREOL: PUSH   HL              ;SAVE HL
0144 CD1F01     0254      CALL   CPOINT           ;POINT TO START OF LINE
0147 EB         0255      EX      DE,HL
0148 E1         0256      POP     HL
0149 E5         0257      PUSH   HL
014A E5         0258      PUSH   HL
014B EB         0259      EX      DE,HL
014C 015000     0260      LD       BC,80
014F 09         0261      ADD     HL,BC              ; POINT TO NEXT LINE
0150 A7         0262      AND     A
0151 ED52      0263      SBC    HL,DE              ; COMPUTE THE # OF CHARS TO BE DELETED
0153 44         0264      LD       B,H
0154 4D         0265      LD       C,L
0155 0B         0266      DEC    BC              ;LENGTH OF REST OF LINE IN BC
0156 E1         0267      POP     HL
0157 CD5C01     0268      CALL   CLR              ;CLEAR REST OF LINE
015A E1         0269      POP     HL
015B C9         0270      RET
                0271 ;
                0272 ; CLR CLEARS FROM (HL) , BC-LOCATION'S OF VIDEO RAM
                0273 ;
015C E5         0274 CLR:   PUSH   HL
015D 54         0275      LD       D,H
                0276 ;
015E 5D         0276      LD       E,L
015F 13         0277      INC    DE              ;DE POINTS TO START OF CLEAR
0160 3620      0278      LD       (HL),' '
0162 EDB0      0279      LDIR                    ;CLEAR REST OF LINE
0164 E1         0280      POP     HL              ;RESTORE CURSOR
0165 C9         0281      RET
                0282 ;
                0283 ; CLEAR SCREEN FROM CURRENT CURSOR POSITION TO BOTTOM
                0284 ;
0166 E5         0285 CLREOS: PUSH   HL
0167 EB         0286      EX      DE,HL
0168 2180FF     0287      LD       HL,CRTEND
016B A7         0288      AND     A
016C ED52      0289      SBC    HL,DE
016E EB         0290      EX      DE,HL
016F 4B         0291      LD       C,E
0170 42         0292      LD       B,D
0171 CD5C01     0293      CALL   CLR
0174 E1         0294      POP     HL
0175 C9         0295      RET
                0296 ;
                0297 ; MOVE CURSOR ONE LINE UP
                0298 ;
0176 11B0FF     0299 UPSCR:  LD       DE,-80
0179 19         0300      ADD    HL,DE
017A 3A0802     0301      LD       A,(ROW)
017D 3D         0302      DEC    A              ;ROW=ROW-1
017E FE18      0303      CP     24              ;TEST IF UNDERFLOW
0180 3801      0304      JR    C,UPSCR1         ;NO UNDERFLOW
0182 AF         0305      XOR    A              ;WRAP TO START ROW
0183 320802     0306 UPSCR1: LD      (ROW),A
0186 C9         0307      RET
                0308 ;

```

```

0309 ; MOVE CURSOR DOWN, IF ALREADY AT BOTTOM, IGNORE
0310 ;
0187 115000 0311 DNSCR: LD DE,80
018A 19 0312 ADD HL,DE
018B 3A0802 0313 LD A,(ROW)
018E 3C 0314 INC A
018F FE18 0315 CP 24D
0191 3801 0316 JR C,DNSCR1
0193 3D 0317 DEC A
0194 320802 0318 DNSCR1: LD (ROW),A
0197 C9 0319 RET
0320 ;
0321 ; PERFORM LINEFEED, IF AT BOTTOM, MOVE HOLE SCREEN ONE LINE UP
0322 ; AND CREATE A NEW CLEAR LINE AT BOTTOM OF SCREEN
0323 ;
0198 CD8701 0324 LFEED: CALL DNSCR
019B D8 0325 RET C ;CARRY MEANS, THAT NO AT BOTTOM
019C E5 0326 PUSH HL ;SAVE CURSOR
019D 2150F8 0327 LD HL,CRTMEM+80
01A0 1100F8 0328 LD DE,CRTMEM
01A3 013007 0329 LD BC,23*80 ;MOVE UP REST OF SCREEN 1 LINE
01A6 EDB0 0330 LDIR
01A8 CDAD01 0331 CALL CLRLIN ;CLEAR THE NEW LINE
01AB E1 0332 POP HL ;GET CURSOR BACK
01AC C9 0333 RET
0334 ;
0335 ; CLEAR THE CURRENT LINE
0336 ;
01AD E5 0337 CLRLIN: PUSH HL
01AE CD1F01 0338 CALL CPOINT
01B1 015000 0339 LD BC,80
01B4 CD5C01 0340 CALL CLR
01B7 E1 0341 POP HL
01B8 C9 0342 RET
0343 ;
0344 ; HOME CURSOR WITHOUT CLEARING SCREEN
0345 ;
01B9 3E20 0346 HOMEUP: LD A,' '
01BB C3DA01 0347 JP SETROW
0348 ;
0349 ; PROCESS ALL ESCAPE-SEQUENCES
0350 ;
01BE EB 0351 MULTI: EX DE,HL ;RESET ESCAPE MODE
01BF 3600 0352 LD (HL),0
01C1 EB 0353 EX DE,HL
01C2 FE01 0354 CP 1
01C4 200D 0355 JR NZ,M2TST
01C6 79 0356 SETXY: LD A,C ;GET CHAR
01C7 FE2A 0357 CP '*' ;CLEAR-SCREEN COMMAND ?
01C9 CA2E01 0358 JP Z,CLRSCN
01CC FE3D 0359 CP '=' ;SET CURSOR COMMAND ?
01CE C0 0360 RET NZ ;NO,ABORT
01CF 3E02 0361 LD A,2 ;OK, SET ESCAPE-MODE TO 2
01D1 12 0362 LD (DE),A
01D2 C9 0363 RET
01D3 FE02 0364 M2TST: CP 2
01D5 2013 0365 JR NZ,M3TST
01D7 3E03 0366 LD A,3 ;SET ESCAPE MODE TO 3
01D9 12 0367 LD (DE),A
01DA 79 0368 SETROW: LD A,C ;ARRIVE HERE ON THIRD CHAR
01DB DE20 0369 SBC ' ' ; OF ESC, '=',ROW,COL SEQUENCE
01DD DE18 0370 SETR2: SBC 24
01DF 30FC 0371 JR NC,SETR2 ; MAKE SURE ROW# IS 0-23
01E1 C618 0372 ADD A,24
01E3 320802 0373 LD (ROW),A
01E6 CD1F01 0374 CALL CPOINT
01E9 C9 0375 RET
01EA FE03 0376 M3TST: CP 3
01EC 2013 0377 JR NZ,M4TST
01EE 79 0378 SETCOL: LD A,C ; ARRIVE HERE ON FOURTH CHAR
01EF DE20 0379 SBC ' ' ; OF ESC, '=',ROW,COL SEQUENCE
01F1 DE50 0380 SETC2: SBC 80
01F3 30FC 0381 JR NC,SETC2 ; MAKE SURE COL# 0-79
01F5 C650 0382 ADD A,80
01F7 F5 0383 PUSH AF
01F8 CD1F01 0384 CALL CPOINT ; GET START OF LINE
01FB F1 0385 POP AF
01FC 5F 0386 LD E,A
01FD 1600 0387 LD D,0
01FF 19 0388 ADD HL,DE ; POINT TO CORRECT POSITION ON SCREEN
0200 C9 0389 RET

```

```

0201 CDB000      0390 M4TST: CALL   DISPLA
0204 C9          0391          RET
                0392 ;
                0393 ;
                0394 ; VARIABLE RAM AREA, FILLED WITH INITIAL VALUES
                0395 ;
                0396 ;
0205 00          0397 LEADIN: DEFB   0           ; HOLDS THE STATE OF ESCAPE SEQUENCE
0206 00F8       0398 CURSOR: DEFW  CRTMEM      ; HOLDS THE ABSOLUTE ADDRESS OF CURSOR
0208 00         0399 ROW:   DEFB   0           ; HOLDS THE CURRENT ROW NUMBER
0209 0000       0400 SPSAVE: DEFW  0           ; HOLDS IN LATER APPLICATIONS THE SYSTEM-STACK
020B 0000       0401          DEFW  0,0
                0000
020F (0000)     0402          END

```

CROSS REFERENCE LISTING

```

BAKSPC 0191 0163
BELL    0209 0162
CLR     0274 0268 0293 0340
CLREOL 0253 0170
CLREQS 0285 0161
CLRLIN 0337 0171 0331
CLRSCN 0240 0169 0358
CONOUT 0049 0017
CONTRL 0122 0111
CPNTL  0234 0235
CPOINT 0228 0254 0338 0374 0384
CRLOP  0071 0079
CRT1    0058
CRTBAS 0038 0062
CRTEND 0037 0072 0114 0287
CRTF    0034 0012 0089 0093
CRTMEM 0036 0018 0019 0037 0038 0039 0085 0232 0240 0242 0327 0328 0398
CRTR    0035 0009 0091 0095
CRTTOP  0039
CRTUP   0084 0076
CTLSIZ 0175 0124
CTLTAB 0147 0123 0175
CURSOR 0398 0059 0084
DISPLA 0112 0390
DNSCR  0311 0324
DNSCR1 0318 0316
ESCAPE 0179 0172
FORSPC 0196 0167
HILO   0215 0115
HOMEUP 0346 0173
INITSC 0006
INITL  0011 0015
LEADIN 0397 0105
LFEEED 0324 0117 0165
M2TST  0364 0355
M3TST  0376 0365
M4TST  0390 0377
MODES  0025 0007
MULTI  0351 0108
OUTCH  0105 0067
RETURN 0224 0168
ROW     0399 0229 0248 0301 0306 0313 0318 0373
SEARCH 0135 0125
SETC2   0380 0381
SETCOL  0378
SETR2   0370 0371
SETROW  0368 0347
SETXY   0356
SPSAVE 0400 0050 0099
STUFF   0185 0174
TAB     0201 0164
TEMSTK 0043 0051
UPSCR   0299 0166
UPSCR1 0306 0304

```

Bei achtzig Zeichen pro Zeile bekommt man in so einer Sprechblase doch gewaltig mehr unter als mit unserem alten Format!



Seite(n) für Einsteiger

von Günter Böhm

Diesmal auf dieser Seite etwas über BASIC. Dennoch bleiben wir auf der Maschinenebene; denn auf Leserwunsch sollen wir einmal etwas über PEEK, POKE, DEEK und DOKE schreiben. Diese Befehle gestatten es, direkt in den Speicher "einzusteigen", und so könnte dieser Artikel sowohl für BASIC als auch für Assembler Interessierte nützlich sein.

Mit dem Befehl PEEK(x), der auch auf anderen Rechnern üblich ist, kann man eine Adresse auf ihren Inhalt untersuchen. Mit PRINTPEEK(x) können Sie den Inhalt der Adresse ausgeben lassen. Man muß nur darauf achten, daß die Adressen im Hex-System "geladen" sind, die Ausgabe aber dezimal erfolgt. Da kann man sich keine Umrechnerei ersparen.

Interessanter als PEEK ist der POKE-Befehl. Damit kann man den Inhalt von Adressen ändern und somit direkt in Maschinenprogramme eingreifen. Wir wollen zur Anschauung einmal die Blinkgeschwindigkeit des Cursors ändern, die ja vom Betriebssystem (in meinem Fall NASSYS3), also auf Maschinenebene gesteuert wird. Dazu schauen Sie zunächst einmal nach, welche Werte die verantwortlichen Adressen normalerweise enthalten. Mit PEEK(3122), das ist die Adresse des niederwertigen Bytes, und PEEK(3123), der Adresse des höherwertigen Bytes, erhalten Sie die entsprechenden Werte (PRINT nicht vergessen!). Setzen Sie das MSB (Most Significant Byte=höherwertiges B.) nun durch POKE3123,0 auf Null, und geben Sie in die Adresse des LSB (Least Significant Byte) verschiedene Werte. PEEK und POKE müssen nicht in ein Programm eingebunden werden, sie arbeiten auch als Kommandos direkt vom Keyboard (Entschuldigung, Herr Bach; von der Tastatur!). Die Befehle DEEK und DOKE sind eine Besonderheit des NASCOM. Sie bewirken das gleiche wie ihre kleineren Brüder, nur kann man hiermit zwei aneinanderhängende Adressen versorgen; d.h. mit DEEK werden zwei Bytes gelesen, mit DOKE zwei Bytes geladen. Man muß darauf achten, daß die "Doppelbytes" in umgekehrter Reihenfolge im Speicher liegen (LSB und dann MSB). Der Hex-Wert 0C80 liegt also als 80 0C im Speicher. Letzterer Wert muß zum Umrechnen in eine Dezimalzahl ver-

wendet werden. Ergibt sich dabei eine Zahl größer als 32767, so muß diese von 65536 subtrahiert und als negative Zahl verwendet werden. Unser Beispiel ergibt also 3200 dez. (sehr häufig verwendet als Start eines Maschinenprogramms, da es die erste verfügbare Adresse für den NASCOM-Benutzer ist). Die Adresse C8D7hex ergibt 51415 und muß als -14121 eingegeben werden.

Die DEEK und DOKE Befehle haben den Vorteil, in den BASIC Zeilen Platz zu sparen. Der Nachteil ist die schwierigere Umrechnung von dez in hex und umgekehrt (das DOKE-Programm von Reinhard Zickwolf, Journal 10-81 hilft dabei) und der Aufwand, Programme mit diesen Befehlen für andere Rechner umzuschreiben. Ich werde in Zukunft auf das Bißchen Platzersparnis verzichten und nur noch mit PEEK und POKE arbeiten. Viele Besitzer anderer Rechner als NASCOM werden Ihnen dankbar sein, wenn Sie auch "durchsichtiger" programmieren!!

Unser Beispiel mit der Blinkgeschwindigkeit ließe sich natürlich mit DOKE3122,X verkürzen, aber....

Falls Sie Maschinenprogramme des Betriebssystems aufrufen wollen, schaden die DEEKs und DOKEs allerdings nichts, denn zu diesen Routinen haben "Fremdbenutzer" sowieso keinen Zugang. Schreiben Sie aber dann in der Dokumentation oder in REM Anweisungen genau dazu, was die BASIC Zeilen bezwecken, denn eine Rechneigene Adresse ist für einen Besitzer eines anderen Systems selbstverständlich niemals zu identifizieren.

Hier nun einige Adressen, die zum POKEN interessant sein könnten:

Bildschirmbeginn links oben	2058
Bildschirmende rechts unten	3001
Beginn der Titelzeile	3018
Ende der Titelzeile	3065
(eine Position in gleicher Spalte eine Zeile tiefer wird jeweils durch +64 ermittelt)	
Cursorposition	3113
Tastendruck Verzögerung	3118
Repeatgeschwindigkeit	3120
Cursor Blinkgeschwindigkeit	3122
\$UOUT(Adresse der Ausgabe)	3191
SRLX (Out über Serial Port)	91
MTFLP (Drive LED ein/aus)	81
Get Input (BIN)	105
BLINK until Input	120
Get a Line (INLIN)	752

Die letzten 5 Adressen können als Maschinenprogramm aufgerufen werden, wenn man sie in USRLOC einträgt (das ist der Speicher für den Beginn des Maschinenprogramms in 4100).

Beispiel:

```
10 DOKE4100,81:REM MTFLLP als Masch.progr.
20 Z=USR(0)
30 FORI=1TO1000:NEXT:REM Verzögerung
40 GOTO20
```

Das Beispielprogramm läßt die Drive LED blinken.

Vielleicht könnte ein "fortgeschrittener" Leser noch einige Informationen zur Benutzung von Maschinenprogrammen zusammentragen (z.B. die Übergabe von Parametern oder den Aufruf verschiedener Programme durch versch. Argumente). Die Maschinenprogramme machen das BASIC einfach schneller, in manchen Fällen sind sie sogar unerlässlich, wenn bestimmte Funktionen wie z.B. INKEY\$ im Wortschatz fehlen und ersetzt werden müssen. Für jeden Beitrag, der unseren Arbeitsaufwand für's Journal vermindert, sind wir dankbar. Es ist so schon genug zu tun. Packen Sie's an!

SPIELAUTOMAT

von Erich Mehnert

Der Spielautomat wird bei E00 gestartet. Die Tasten A-S-D werden zum Stoppen der Walzen benutzt.

Bei Übereinstimmung von drei Zahlen ertönt eine Melodie, und der Ergebniszähler zählt vorwärts. Bei dreimaligem Erscheinen der eins, zwei oder drei werden zusätzlich noch Sternchen gesetzt. Erscheint die Sieben in der Mitte, wird ein Punkt gezählt (Mindestgewinn), und eine Melodie ertönt. Die Töne werden an der Tastatur Pin13 abgenommen.

0C80	21	50	08	22	29	0C	06	20	82
0C88	3E	7F	F7	3E	12	F7	3E	08	D5
0C90	F7	00	00	00	10	F2	C9	CD	2B
0C98	80	0C	3E	05	32	87	0C	3E	76
0CA0	14	32	8C	0C	CD	80	0C	21	04
0CA8	50	09	22	81	0C	3E	12	32	3E
0CB0	8C	0C	3E	20	32	87	0C	CD	44
0CB8	80	0C	21	70	09	22	81	0C	99
0CC0	3E	05	32	87	0C	3E	13	32	57
0CC8	8C	0C	CD	80	0C	21	60	08	4E
0CD0	22	81	0C	3E	0A	32	87	0C	98
0CD8	3E	14	32	8C	0C	CD	80	0C	59
0CE0	3E	50	32	81	0C	3E	20	32	C9
0CE8	87	0C	3E	12	32	8C	0C	C9	6A
0CF0	D9	EB	44	7D	AB	67	78	AA	B5
0CF8	CB	11	1F	CB	19	6F	D9	CB	F6
0D00	E7	CB	EF	CB	B7	CB	BF	CB	85
0D08	9F	32	93	08	00	D9	EB	45	

```
0D10 44 7D AB 67 78 AA CB 11 EE
0D18 1F CB 19 6F B9 CB E7 CB ED
0D20 EF CB B7 CB BF CB 9F CB C4
0D28 13 09 C9 00 00 CD AB OF A1
0D30 21 93 08 22 0A OD 21 13 66
0D38 09 22 28 OD CD 03 OE CD 50
0D40 24 OE CD 87 OE CD 54 OE 10
0D48 CD A8 OE CD A8 OE 3A D6 6B
0D50 08 FE 37 CA 59 OD C3 5C E9
0D58 OD CD 46 OF C3 7C OD 3A 1A
0D60 D6 08 32 6C OD 32 77 OD AC
0D68 3A 13 09 FE 2A CA 73 OD 3D
0D70 00 C9 00 3A 99 08 FE 2A 49
0D78 CA B3 OD C9 21 93 08 22 B6
0D80 69 OD 21 99 08 22 74 OD 68
0D88 CD 5F OD 21 19 09 22 74 A7
0D90 OD CD 5F OD 21 13 09 22 42
0D98 69 OD CD 5F OD 21 99 08 16
0DA0 22 74 OD CD 5F OD CD A8 FE
0DA8 OE CD A8 OE CD A8 OE 00 C9
0DB0 C3 DF OD CD 46 OF CD 46 A1
0DB8 OF 3E 2A 32 93 08 32 99 D4
0DC0 08 32 D6 08 32 13 09 32 65
0DC8 19 09 CD A8 OE CD 46 OF 9C
0DD0 CD A8 OE CD 46 OF C3 B9 FE
0DD8 OE 00 00 00 00 00 00 3E 31
0DE0 00 32 93 08 32 99 08 32 BF
0DE8 D6 08 32 13 09 32 19 09 75
0DF0 CD A8 OE C3 30 OD C7 C6 OD
0DF8 2A 38 7E 6D FB 89 4D CE 21
0E00 C3 2D OD 21 3F 00 3E 02 AB
0E08 CD FO OC F5 AF 3D C2 OD 8F
0E10 OE F1 3D C2 0E 0E DF 62 76
0E18 FE 41 CA 24 0E 2B 7D B4 BD
0E20 C2 06 OE C9 21 D6 08 22 EE
0E28 0A OD 22 28 OD 21 OF 00 D4
0E30 3E 02 CD FO OC F5 AF 3D 28
0E38 C2 37 OE F1 3D C2 35 OE 80
0E40 DF 62 FE 53 CA 53 OE 2B 36
0E48 7D B4 C2 30 OE 00 00 00 87
0E50 00 00 00 C9 21 99 08 22 0B
0E58 0A OD 21 19 09 22 28 OD 17
0E60 21 2F 00 3E 0F CD FO OC D4
0E68 F5 AF 3D C2 6A OE F1 3D BF
0E70 C2 68 OE DF 62 FE 44 CA 03
0E78 86 OE 2B 7D B4 C2 63 OE A9
0E80 00 00 00 00 00 00 C9 00 57
0E88 00 00 21 12 00 3E 02 DF E8
0E90 62 FE 53 CA 24 OE F5 AF F1
0E98 3D C2 98 OE F1 3D C2 96 D1
0EA0 OE 2B 7D B4 C2 8D OE C9 3E
0EA8 F5 C5 06 60 FF 10 FD C1 A3
0EB0 F1 C9 00 00 00 00 00 00 78
0EB8 00 3A 6C OD FE 30 CA D6 47
0EC0 OE FE 31 CA DC OE FE 32 EF
0EC8 CA E2 OE FE 33 CA E8 OE 81
0ED0 00 00 00 C9 00 00 C3 1D 87
0ED8 OF 00 00 00 3E 2A 32 E4 73
0EE0 08 C9 3E 2A 32 E5 08 C9 OF
0EE8 3E 2A 32 E6 08 C9 00 3A 81
0EF0 1B OF CE 01 27 32 1B OF 7A
0EF8 3A 1C OF CE 00 27 32 1C AE
0F00 OF 21 EB 08 22 29 0C 2A B3
0F08 1B OF DF 66 C9 00 00 00 4F
0F10 00 00 00 00 00 00 00 1F
0F18 00 00 00 26 00 3E 2A 32 E7
0F20 E3 08 3A E4 08 FE 2A CC 34
0F28 41 OF 3A E5 08 FE 2A CC A2
0F30 41 OF 3A E6 08 FE 2A CC AB
0F38 41 OF CD 41 OF CD 01 CF 91
0F40 C9 3E 00 C3 0A OF 3E 04 74
0F48 00 00 OE 00 21 6C OF DD DE
0F50 21 8B OF 06 20 DD 5E 00 7B
0F58 56 D3 00 15 20 FB ED 51 FE
0F60 1D C2 58 OF 23 DD 23 10 E8
0F68 EC C3 EE OE 70 72 74 78 FO
0F70 7A 7C 7E 80 82 84 86 88 87
0F78 8A 8C 8E 90 92 94 96 98 OF
0F80 9A 9C 9E AO A2 A4 A6 A8 97
```

```

GF88 AA AC AE 10 10 10 10 10 EB
OF90 10 10 10 10 10 10 10 10 1F
OF98 10 10 10 10 10 10 10 10 27
OFA0 10 10 10 10 10 10 10 10 2F
OFA8 10 10 10 3E 0C F7 21 30 79
OFB0 30 22 ED 08 22 EB 08 CD E8
OFB8 97 0C C9 C2 39 7D 07 C2 74
OFC0 DA DA 4D CD B7 B9 2D 4C 86
OFC8 13 5A 87 C5 32 FE CD 4F DC
OFD0 B3 98 46 44 32 30 94 85 2F
OFD8 08 B2 44 02 53 91 80 07 52
OFE0 20 20 65 07 20 32 FA 0B F2
OFE8 48 0C 79 02 B8 01 0C 0C 97
OFF0 6B 07 0C 0C 00 00 34 0C C9
OFF8 85 04 00 00 00 00 01 30 C1

```

ZEAP-ASM von Günter Böhm

Für manche Benutzer ist es notwendig, Assemblerlistings von ASM Assembler an ZEAP 2.0 anzupassen (z.B. Programmtausch auf Cassette). Die Anpassung bereitet keine großen Schwierigkeiten.

Der Textpuffer des ASM (#1000) beginnt mit der freien Adresse nach dem Textpuffer (z.B. #24CC). Er wird durch # 00 FF abgeschlossen.

Der Puffer des ZEAP (normalerweise auch #1000) ist folgendermaßen aufgebaut:

Pufferlänge : LB HB (inklusive Endzeichen)

Zeilenzahl mal 2 LB MB HB

dann Nummer der ersten Zeile in Dez wie bei ASM

Das Endzeichen des Textpuffers ist 00 FF 00.

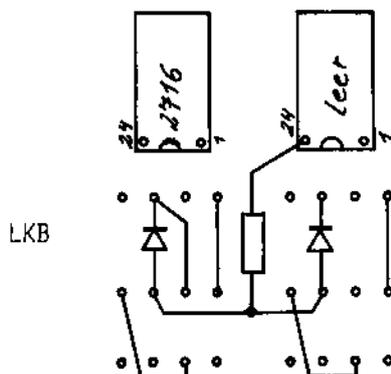
Bei der Anpassung ist also zunächst die Pufferlänge einzutragen (in unserem Beispiel CC 14) dann die Anzahl der Zeilen in Hex (deshalb empfiehlt es sich durch den Renumber Befehl immer den gleichen Zeilenabstand einzuhalten, damit man die Zahl genau errechnen kann). Nehmen wir an, der Sourcecode habe eine Länge von 50 Zeilen. Dann wäre ab #1002 einzutragen 64 00 00.

Nun muß man nur noch das Endzeichen des ASM in 00 FF 00 verändern und den ZEAP warmstarten. (Der Kaltstart vor der ganzen Prozedur ist selbstverständlich).

Die erste Zeile ist nun unvollständig, weil ihre Zeilennummer von der Zeilenanzahl überschrieben wurde; das kann durch editieren leicht korrigiert werden. Das Programm steht nun für ZEAP zur Verfügung.

N2 Hardware-Tip von Peter Brendel

Dies ist ein Hardware-Tip, wie man ein EPROM 2716 in den NASCOM2 einbaut, ohne irgendwelche Leiterbahnen zu durchtrennen. Ich habe einen Sonntagnachmittag damit verbracht, dies auszutüfteln. Warum soll sich ein anderer unbedingt dieselbe Arbeit machen?



Widerstand 10 K

Diode 1N 4448 o.ä.

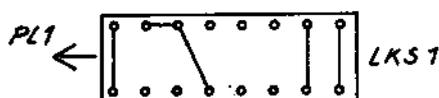
BESCHALTUNG des LKS 1

je nach Anfangsadresse des bestückten Sockels.

Beispiel für B5, B6 Sockel

Dyn. RAM- Erweiterung

Eprom auf B000

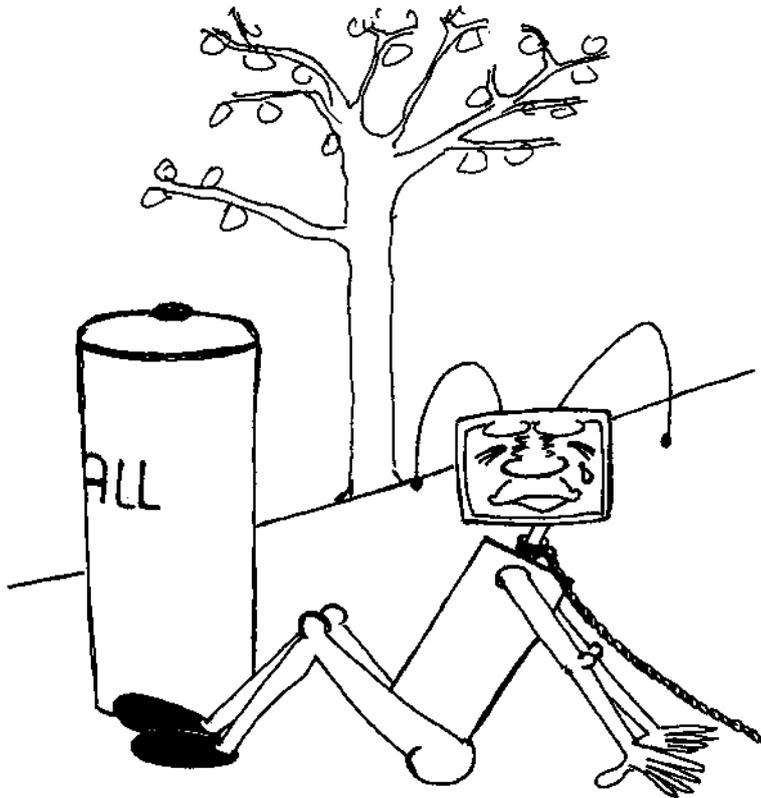


Um alle Sockel zu nutzen, müßte man wohl das N2MD PROM ändern.

SUCHE Integer PASCAL auf Cassette für
NASSYS 1
Hermann Lottes

NASCOMPL

IMPRESSUM



Hallo liebe Leser,
Es ist mal wieder soweit: die Reisezeit ist angebrochen. Immer größere Autokolonnen bilden sich auf dem Weg gen Süden. Jeder freut sich auf die "schönste Zeit des Jahres", wo man einmal abschalten kann und Abstand gewinnt zum tristen Alltag. Soweit ist das ja alles in Ordnung, und niemand könnte etwas einwenden, wenn die ganze Sache nicht einen großen Haken hätte: Computer halten nichts vom Abschalten, und so sind sie ihren Besitzern meist ein Klotz am Bein. Schlimm genug, daß viele Tischrechner zur Sommerzeit dann lieblos zu Bekannten oder Verwandten abgeschoben werden, die zudem häufig keine Fachleute sind und von Wartung und Pflege keine Ahnung haben, geschweige denn die für den Rechner so wichtige Zuwendung aufbringen. Der Gipfel an Herzlosigkeit ist aber das Aussetzen von Computern an Autobahn-Rastplätzen! Neben Abfällen und ausgelehrten Aschenbechern wartet dann so ein hilfloser Rechner auf sein weiteres Schicksal. Meist endet die Tragödie mit der Einlieferung in eine Altmaterialverwertung, weil Städte und Gemeinden nicht die Mittel aufbringen können (oder wollen!), Räume für herrenlose Rechner bereitzustellen.
Hier sollten wir unser bestes tun, die Not etwas zu mildern. Wenn Sie also an einem Rastplatz z.B. einen herrenlosen APPLE stehen sehen, so nehmen Sie ihn mit! Er wird es Ihnen danken. MK-Systemtechnik würde da auch gerne helfen, die haben aber leider schon den ganzen Keller voll solcher Findlinge. Ein großes Beispiel der Nächstenliebe!
In diesem Sinne Ihr NASCOMPL

REDAKTION: Günter Böhm, Günter Kreidl
Wolfgang Mayer-Gürr, Josef Zeller

RESSORTS:

MASCHINENPROGRAMME:

Günter Böhm,

■ Karlsruhe,

Tel. ■■■■■

Günter Kreidl,

■■■ Straelen

Tel. ■■■■■

BASIC und FLOPPY:

Wolfgang Mayer-Gürr,

■■■ Recklinghausen

Tel. ■■■■■

HARDWARE:

Josef Zeller,

■■■ Neu-Ulm

VERLAG: NASCOM JOURNAL, c/o MK-Systemtechnik

Pater-Mayer-Str.6,

6728 Germersheim

Tel. 077274/2756

Telex 453500 mkxd

VERTRIEB: Direktvertrieb durch den Verlag

Erscheinungsweise: monatlich

Bezugspreis: Im In- und Ausland 48,- für

ein Jahresabonnement. Abonnements können

aus technischen Gründen immer nur für die

Dauer eines Kalenderjahres, d.h. vom 1.1.

bis 31.12. laufen. Bei Bestellung nach dem

1.1. werden die fehlenden Hefte mit der

ersten Lieferung bis zum Bestellzeitpunkt

automatisch mitgeliefert. Bei nicht frist-

gerechter Kündigung verlängert sich das

Abonnement automatisch um ein Jahr. Die Kün-

digung für das Folgejahr muß bis spätestens

sechs Wochen vor Jahresende erfolgen.

Bezugsmöglichkeiten: Durch Bestellung bei MK

Systemtechnik.

Bankverbindungen: Alle Zahlungen für das

NASCOM JOURNAL unter Angabe der Rechnungs-

nummer an MK - Systemtechnik, Germersheim.

Zahlung: Nach Eingang Ihrer Bestellung er-

halten Sie von uns die ausstehenden Hefte

bis zur aktuellen Ausgabe sowie eine Rech-

nung. Bitte, zahlen Sie dann den Rechnungs-

betrag.

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 Anga-

ben kann keine Haftung übernommen werden.

nascocom

Die Alternative!

Kein »langweiliger Computer«

NASCOM 1 und NASCOM 2 sind Computer für Selbsterbauer, Tüftler, erfolgreiche Do-it-yourself-Freunde. NASCOM-Computer werden niemals langweilig! Die Systeme 1 und 2 sind keine fertigen »Kästen« ohne Erweiterungsmöglichkeit. Der hochwertige Platiniensatz Computer und Keyboard kann so aufgebaut, erweitert und »verpackt« werden, wie Sie es wünschen.

Für Vollpreis gibt die NASCOMs auch als Bausatz. Aber aufgepaßt: Das ist eine Sache nur für wirkliche Könnert! Und damit es auch nach dem Aufbauen nicht langweilig wird, gibt es das monatlich erscheinende NASCOM-JOURNAL. Eine Zeitschrift speziell für NASCOM-Freaks vollgestopft mit Hardware- und Software-Ideen, Kleinanzeigen, den neuesten Infos, und, und, und. . .

Die NASCOMs sind keine »Spielcomputer«. Mehr als 60% aller NASCOM-Systeme werden als sogenannte

»OEM-Baugruppen« von professionellen Anwendern in eigene Systeme eingebaut. Ingenieurbüros verwenden den NASCOM als Entwicklungssystem. Die Anwendungsmöglichkeiten sind mehr durch Ihre Phantasie begrenzt. Ein NASCOM-System kann fast alle gängigen Probleme lösen.

Mit NASCOM wachsen!

NASCOM-Systeme sind aufwärtskompatibel. Das kleinste, preisgünstigste NASCOM 1-System kann bis auf NASCOM 3-Level mit Floppy-Laufwerken und CP/M* ausgebaut werden. Bildschirm-Aussteuerung, Tastatur Betriebssystem und Systemsoftware sind durchweg kompatibel. Ohne faule Kompromisse! Mit NASCOM-Systemen gehen Sie kein Risiko ein. Ihr NASCOM wächst mit!



NASCOM 1

Spezifikationen:

- QWERTY-Tastatur, aufgebaut mit hochwertigen Magnetasten
- NAS-SYS Betriebssystem (2k Byte)
- 16 I/O-Leitungen
- Video (BAS) und TV-Ausgang
- 1k RAM, ausbaubar auf 192k RAM
- Display 48 Zeichen in 16 Zeilen

ab DM 935,-



NASCOM 2

Spezifikationen:

- Wie NASCOM 1, jedoch zusätzl.:
- 8k Mikrosah-BASIC u. 8k Stat. RAM
- Z80A-Mikroprozessor, 4 MHz
- Erweiterte Tastatur 57 Tasten
- Integrierte Bus-Pufferung
- Bis 192k Byte RAM
- Grafik-Möglichkeiten: 48 x 96 Punkte
- Serielle Schnittstelle; Baudrate wählbar, RS232C/20mA
- 16 parallele Ein/Ausgabeleitungen (Z80APIO)

ab DM 1950,-

nascocom 3 — der Profi



Spezifikationen: Wie NASCOM 2, jedoch zusätzl.:

- 0.35 Megabyte pro 5,25-Zoll Laufwerk
- Betriebssystem CP/M* 2.2 oder NAS-DOS
- Bildschirmausgabe erweiterbar auf 80 x 25 Zeichen

ab DM 2735,-

Die dritte NASCOM-Generation

NASCOM 1 und 2 haben OEM-Board, Schulungscomputer, Kompaktrechner etc. ca. 20 000 mal ihren Partner gefunden. Der NASCOM 3 möchte Ihr persönlicher Computer werden! Er möchte Ihnen helfen, sich selbst fortzubilden, im Beruf weiter zu kommen, auch mal in die Computertechnik »rein zu riechen«. Ingenieurbüros und Softwareingenieuren dient der NASCOM 3 als preisgünstiges Entwicklungssystem.

Universelle Betriebssoftware

Der NASCOM 3 kennt zwei Betriebssysteme: Das CP/M* (Version 2.2) — inzwischen Standard — und sein eigenes NAS-DOS. Die 5-Zoll Floppys bieten eine Speicherkapazität von 0.35 Megabyte pro Laufwerk (single sided, double density, double tracked). Damit wird das Spektrum universeller CP/M*-Software verfügbar!

**Wir informieren Sie unverbindlich:
Fordern Sie Ihr NASCOM-INFO-PAKET an! ****

Unsere Händler:

Heinz Vogel Verlag GmbH & Co.
Lehrmittellehrzentrum, Herr Seloff
Innsbrucker Straße 96
2800 Bremen-Findorff
☎ (04 21) 35 10 69

Christian Lampson
W.-Leuschner-Straße 4
6085 Nauheim
☎ (06 152) 5 67 30

MK-SYSTEMTECHNIK
Michael von Keitz
Plattenberg 4
5650 Solingen
☎ (0 21 22) 4 72 67

MK-SYSTEMTECHNIK
Kriegsstraße 104
7500 Karlsruhe
☎ (07 21) 2 92 43

Radio Zinburg
Herr Zinburg, Jr.
Röhstraße 10
5760 Arnsberg
☎ (0 29 32) 3 15 10

Graf Elektronik Systeme GmbH
Postfach 1610
8791 Kempten
☎ (08 31) 6 19 30

Autorisierter Distributor:



MK-SYSTEMTECHNIK
Pater-Mayer-Straße 6
6728 Germersheim
☎ (0 72 74) 20 93
Telex 453500 mks d

CP/M* ist ein eingetragenes Warenzeichen der DIGITAL RESEARCH
** NASCOM-INFO-PAKET gegen DM 2,- in Briefmarken (wird bei Kauf angerechnet)