nascom journal

Zeitschrift für Anwender des NASCOM 1 oder NASCOM 2

3. Jahrgang · Januar 1982 · Ausgabe 1

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 Keitz · 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

nascom journal

INTERN

Liebe Leser,
endlich erhalten Sie die Januar-Ausgabe, auf die Sie möglicherweise schon längere Zeit gewartet haben. Dies schreibe ich in der Annahme, daß sich auch diesmal wieder Verzögerungen bis zum Versand einschleichen. So war das auch mit unserer Weihnachtsausgabe, die Sie noch vor Weihnachten erhalten sollten; aber nach Abschluß der Redaktionsarbeit muß das Heft noch einige Stationen durchmachen, für die das Redaktionsteam nicht mehr verantwortlich ist. So nützen auch keine Telephonanrufe bei mir, denn die Druckunterlagen sind in der Regel ab dem 20. des jeweiligen Monats unterwegs. Wir sollten die Verzögerung einfach einplanen und das Heft vierzehn Tage später erwarten.
Dieses Heft trägt ein neues Gesicht, denn auch wir sind "modebewußt". Es läßt sich so besser heften. Sicher findet mancher Leser aber einen Nachteil darin, während andere die Änderung begrüßen. Unstimmigkeiten zwischen den Ansichten der Leser regen uns aber nicht mehr auf, (Sie können sie zum Teil in den Leserbriefen verfolgen). Wir bemühen uns, gute Vorschläge aufzugreifen und der Mehrzahl der Leser etwas zu bieten. Hier sollten auch Sie etwas tolerant sein; die Leserschaft ist (glücklicherweise) eben eine sehr vielfältige (und vielseitige) Gruppe.
Der Bestellservice der Folien brachte nicht das erwartete Echo. Die Bestellung der Ätzfolien war so gering, daß eine Serienfertigung nicht möglich war. Ich habe die Folien nun zunächst "ausgeliehen) und bitte Sie, sich bei Interesse an einer Folie zu einem "Rundlauf" anzumelden. So entstehen Ihnen keine Kosten; vielleicht war das der Grund für die Zurückhaltung.
Weitere "Rundläufe" zu einzelner Software werden von den Autoren in ihren Artikeln angeboten.
Die Floppy-Tauschaktion läuft noch. Wolfgang Mayer-Gürr hat mir eine Liste der Programme geschickt, die schon zum Tausch zur Verfügung stehen. Ich habe 130 Positionen gezählt. (In Kürze werden wir die Liste veröffentlichen). Wenn Sie am Tausch interessiert sind, schicken Sie eine Floppy (mit möglichst vielen Programmen) an die Tauschzentrale.
Noch eine Bitte an alle Leser: die Redaktionsarbeit nimmt ganz schön Zeit in Anspruch. Erleichtern Sie uns die Arbeit, indem Sie nicht nur Programme, sondern auch Artikel und Leserbriefe auf Cassette schikken. (Diese bekommen Sie zurück, wenn's vielleicht auch eine Weile dauert). Ladeformate wären NASCOM 1 Format und NASCOM 2 - 300 Baud (in Ausnahmefällen auch N2 -1200 B, aber dafür muß ich immer mein Interface neu abstimmen). Das N1 Format ist mir für diesen Zweck allerdings am liebsten. Viele Leser haben das schon so gehandhabt, und ich danke ihnen für die Mühe.
Für die INFO-Ecke liegen keine neuen Informationen vor. Was die Arbeit an einem programmierbaren Zeichengenerator angeht, so hat sich da schon einiges entwickelt. Allerdings fehlt uns noch jemand, der anhand eines ausführlichen Blockschaltbildes einen Prototypen frei verdrahten würde, um die Schaltung einmal grundsätzlich zu testen. (Dies ist nur eine Zeitfrage!) Wer hätte dafür Interesse? Wenn die Schaltung funktioniert, ist das Layout für eine Platine geplant, die sich ohne Kabelverbindungen auf das Grundsystem aufstecken läßt (nach dem "Oberle-Prinzip").
Mit der üblichen Bitte um weitere rege Mitarbeit möchte ich mich für diesmal zurückziehen. Viel Spaß und Ernst bei der Lektüre
Ihr Günter Böhm

P.S. Was halten Sie von Fotos im Journal (wie in der Dezemberausgabe 81)? Falls Sie irgendwelches Material zur Veröffentlichung haben: wir sind dankbare Abnehmer!

Leserbriefe

An die Redaktion des NASCOM Journals
Verbesserungsvorschlaege:
-Fuer jedes Maschinenprogramm eine Pruefsumme nach dem Muster von MC 2/1981 Seite 28 (Z8Ø-Texted.) bilden.
-Die Festlegung auf NASSYS 3 halte ich fuer schlecht; wenn dann alle umgestellt haben, kommt sicher NASSYS 4 usw. Fuer besser halte ich das Anlegen einer Sprungtabelle mit Uebernahme der Daten vom Stack oder etwas Aehnliches.
-In dem Listing von Quest sind wieder Fehler; gebt doch dem Autor eine Druckfahne zur Korrektur, bevor das Journal in Druck geht.
-Basic Listings enthalten immer wieder Sonderzeichen, die nicht ASCII entsprechen (Siehe Quest: O=O=Ø ? mue=^ ? Grad=^ ?). Ich halte das fuer aergerlich.(Wir auch! Die Legende fürdie Sonderzeichen war aber auf Seite 22 abgedruckt. Es existiert aber ein Umsetzprogramm, das diese Zeichen in Zukunft vermeidet. Gut gedruckte und formatierte BASIC Listings veröffentlichen wir ohnehin im Original. Der einzige Fehler in QUEST ist ein Programmierfehler, der von Herrn Peter in diesem Heft erläutert wird. Das Spiel hat nun mal einige frustrierende Fallen, man kann aber durchaus den Schatz aus der Höhle bringen, wenn man nicht vorher aus Verzweiflung aufgibt. Red.)
-Der Strichcode geistert immer wieder durch die Zeilen, und kein Ende ist zu sehen. Wie waer's mit einer Abstimmung? Ich stimme mit NEIN!
Bemerkungen zu Leserbriefen:
-Herr Mombaur ist begeistert, aber warum, interessiert mich nicht. Wenn er aber einen Schwank aus meinem Leben hoeren will, kann er mich ja anrufen.
-Herr Buerger plaediert fuer die Versendung von Assembler Listings. Im Prinzip finde ich das gut, aber was, wenn die Moeglichkeiten fehlen? Nicht jeder hat einen Assembler und einen Printer. Bei dem Vorschlag, einen Taschenrechner als Arithmetikprozessor zu verwenden, dreht sich mir der Magen um. Warum keine richtige APU oder Programme, die es ja zu genuege gibt.(Wie wär's mit einer Einsendung davon? Red.)
Anfragen:
-Ich starte den Versuch, das ROM Basic V4.7 zu disassemblieren und die einzelnen Routinen zu suchen. Wer hat das schon gemacht oder versucht und kann mir dazu Hinweise geben?
-Hat oder plant die Firma Lukas ein 16 Bit Systhem mit Z8ØØØ? (Hier in Deutschland weiß man noch nichts. Es ist aber geplant, mit Lukas direkt Kontakt aufzunehmen, um in Zukunft über solche Dinge früher Bescheid zu bekommen. Red.)lerfrei.
-Falls bei den Lesern Interesse besteht zu erfahren, wie ich einen Magnetband-Digital Speicher (MDS) an meinen NASCOM angebunden habe, stehe ich zur Verfuegung. Das kommt aber nur fuer Leser mit Hobbyraum und Drehstromanschluss in Frage, denn der MDS hat die Ausmaße eines kleinen Kleiderschranks, ist aber als Oldtimer am Markt sehr preiswert zu bekommen.
-Den Lesern, die versuchen, ihren NASCOM mit 64kB dyn. Speichern auszuruesten, kann ich ein paar bescheidene Tips geben (Ich habe selbst noch ein kleines Problem, das aber weiter nicht stoert).
Alle Angaben beziehen sich auf einen NASCOM 2 mit NASSYS 1 und 64kB.
-Alles, was nicht bemosert wurde, koennen Sie auf die Guthabenseite verbuchen.(Da bleibt doch hoffentlich noch etwas? Red.)
So nun wuensche ich froehliches Editieren und bitte Sie noch, vor einem Abdruck im Journal meine Grammatik zu verbessern.(Das ist hoffentlich zur Zufriedenheit geschehen! Red.)
Alles Gute im neuen Jahr wuenscht
Peter Urban, Niefern

Beim Lesen des September-Journals habe ich mich einigermaßen über den Leserbrief von Ulrich Wallis geärgert. (Tschuldigung). Habe ich mich mit meinem leistungsfähigen Entwicklungssystem in einen elitären Verein eingekauft, oder sehe ich das falsch?
Hier meine ganz persönliche Ansicht:
Sie beklagen sich über mangelndes Interesse am Cassettentausch und bitten um rege Mitarbeit der Leser. Es würden vielleicht mehr Leser mitarbeiten, solche wie ich, wenn das hohe Niveau des NASCOM Journals nicht durch

und durch so hoch wäre. Nein, keine Demontage! Das N.J. soll um Himmelswillen so bleiben wie es ist!
Nur - eine kleine, aber wie ich meine, wichtige Bereicherung: Die Seite für den Anfänger.("Ich rede nicht für mich", sagte der Fuchs,"aber man sollte die Hühner in den Wald treiben, damit sie immer etwas zu picken haben").
Ich bin nur ein einfacher Elektromechaniker, der gezwungen ist, den Computer zu packen, bevor er ihn packt. Unter der Leserschaft gibt es bestimmt noch mehr arme Schweine wie mich, die mit ihrem NASCOM allein auf weiter Flur stehen und für jeden Tip dankbar sind, über den die Spezialisten nur noch müde lächeln können. Doch bei dem für meine Bedürfnisse zu hohen Niveau (Herr Wallis ist da bestimmt ganz anderer Meinung) halten sich viele Leser (die schweigende Mehrheit?) mit ihren Beiträgen und Programmen zurück, nur um sich nicht zu blamieren. So beobachtet bei mir selbst. Weiß ich, ob meine Programme mit ihrem möglicherweise wahnwitzigen Aufbau in den Augen der Fachleute bestehen werden? Ein Blick in das NASCOM JOURNAL bestätigt meine Zweifel: Lauter spezielle Artikel von Spezialisten für Spezialisten. Ich schäme mich nicht zuzugeben, daß ich das meiste nicht kapiere.
Nochmals die Bitte um eine Rubrik "von Anfängern für Anfänger" oder so ähnlich. Vielleicht in der Mitte zum Heraustrennen, damit das Auge einiger alter Hasen nicht beleidigt werde. Vielleicht kommt angesichts solcher Anfängerbeiträge hie und da ein guter Tip aus berufenem Munde, wie dieses oder jenes besser zu machen wäre. Der Lerneffekt wäre ein ungeheurer.
Ich finde, das NASCOM JOURNAL sollte für jeden etwas bieten, auch für (noch) Nichtkönner, die es zweifellos gibt. Oder bin ich der einzige? Das wäre wohl ein Grund, die ganze Computerei den Profis zu überlassen.
P.S. Mein NASCOM ist kein diesjähriges Weihnachtsgeschenk, vielmehr raubt er mir schon das 2.Jahr die Zeit und die Nerven. Anscheinend habe ich die höheren Weihen noch nicht erhalten. Noch 'ne Frage: was ist aus dem Kurs "Programmieren in Assembler" geworden?
Gruß an Nascompl.
Peter Brendel, Mannheim.

Vorerst möchte ich der gesamten Redaktion des NASCOM JOURNAL zu der neuen Erscheinungsform gratulieren. Seit das Journal in der neuen Aufmachung erscheint, hat es auch sehr an Qualität gewonnen. Zu Ihrem Artikel in der Ausgabe 10/1981 Punkt 14 ist anzumerken, daß alle bisherigen Betriebssysteme incl. NASSYS 1 den Befehl "BØ" als "Breakpoint an der Stelle ØØØØ ausgeführt haben.
Der Monitor versuchte also, auf die Stelle "ØØØØ" im Speicher "E7" zu schreiben. Nachdem das Betriebssystem normalerweise im ROM steht, wirkt sich das nicht weiter aus. Versucht man aber, das Betriebssystem als eine Art Unterprogramm zu verwenden, das ab Hex "ØØØØ" im RAM (!) steht (was natürlich eine Hardware-Änderung erfordert), so bewirkt "BØ", daß der erste OP-Code mit "E7" überschrieben wird, und daher der Monitor auch nicht mehr das tut, was er soll.
Das Betriebssystem NASSYS 3 geht da einen anderen Weg. Soweit ich das bisher feststellen konnte, (ich habe leider kein Assembler-Listing) wird der Code "E7" erst während der Ausführung des "Execute"-Befehls in die entsprechende Speicherstelle eingeschrieben und bei jeder Rückkehr in NASSYS 3 wieder durch den Original Op-Code ersetzt. Möglicherweise interpretiert der Monitor auch, wie in einer deutschen NASSYS 3 Beschreibung zu lesen ist, den Befehl "BØ" auch wirklich als "Breakpoint abschalten".

Zum Programm "Yatzi" aus dem NASCOM-JOURNAL vom Juli 1981: die Routine zur Auswertung einer "Straße" funktioniert (wahrscheinlich aufgrund eines Denkfehlers) nicht richtig. Es werden z.B. auch folgende Würfe als "Straße" anerkannt: 2,3,3,4,2. Das hat folgenden Grund: es wird nicht ausgewertet, ob jede der zur betreffenden "Straße" gehörenden Zahl einmal vorkommt, sondern, ob jeder der Würfe auf eine der Zahlen paßt. Dadurch werden Doppelwürfe nicht erkannt. Diesem Mißstand kann leicht abgeholfen werden. Es muß ja - definitionsgemäß - nur überprüft werden, ob auch wirklich jede Zahl der "Straße" in dem Wurf vorkommt:

```
2111 for ol=a to o:for ok=1 to 5
2112 if wl(ok)=ol then wl(ok)=Ø:cc=cc+1:ok=5

2113 next ok,ol
```

Ich habe das Programm geändert, so daß

mehrere Spieler gleichzeitig am Computer spielen können, und auch der Spielstand notiert wird. Anbei das komplette Listing. (Haben wir aus Platzgründen nicht abgedruckt. Das Programm kann in einem "Rundlauf" auf Cassette angefordert werden.-Red.)

Ich möchte gleich die Gelegenheit nutzen zur Dezember-Ausgabe zu gratulieren. Wirklich gut gemacht. Im Programm "Quest" habe ich einen Fehler entdeckt, der wahrscheinlich keinem auffallen wird: wenn man nach der ersten Schatzsuche noch nicht genug hat (so wie ich) und auf die Frage hin noch eine Schatzsuche fordert (Zeile 1180), hängt sich das Programm mit einem "RG-ERROR" auf. Die Zeile 1190 springt nämlich mit "GOTO" in eine Subroutine. Am besten statt der Zeilen 1170 bis 1190 ein Delay einbauen.
Christian Peter, Wien.

Sortieren in Basic

Teil 6 von Wolfgang Mayer-Gürr

Die bisher vorgestellten Sortieralgorithmen eignen sich nur für kleinere Felder, die Rechenzeit ist sonst nicht mehr akzeptabel. Schneller geht es mit dem Heap-Sort, das von Williams und Floyd entwickelt wurde.
Zum Verständnis ist die Kenntnis eines binären Baumes sehr hilfreich. Üblich ist die Darstellung einer Folge aus Buchstaben in einer von links nach rechts verlaufenden Reihe. Andere Formen wären von rechts nach links (arabisch) oder von oben nach unten (chinesisch). Bei einem binären Baum ordnet man die einzelnen Bestandteile in mehreren Ebenen. Jede Ebene enthält die doppelte Anzahl an Buchstaben wie die über ihr liegende. Lediglich in der untersten Ebene kann dies meist nicht aufgehen.

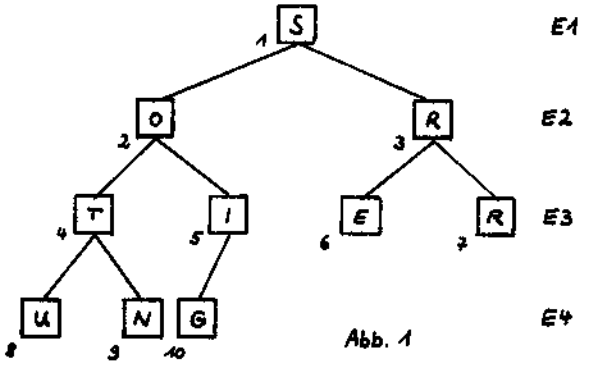


Abb. 1

Jedem Element werden nun 2 (daher binär) unter ihm liegende Elemente zugeordnet. Dreht man Abb.1 auf den Kopf, erhält man das Bild eines Baumes. Behält man die laufende Numerierung des Feldes bei, gehören zu einem Feldinhalt i ("Vater") die Felder 2i und 2i + 1 ("Söhne").

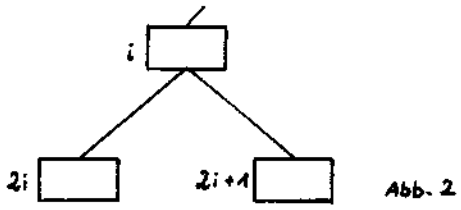


Abb. 2

Ist der Inhalt von i größer oder gleich dem Inhalt von 2i und auch 2i + 1, spricht man von einem Heap. Auch ein "kinderloser Vater" ist dann ein Heap.
Im ersten Teil des Programms (Zeile 280 - 320 mit Unterprogramm 410 - 520) wird nun der Heap angelegt. Element N/2 ist von rückwärts gesehen der erste "Vater" mit einem "Sohn". Im Unterprogramm wird dem "Vater" der größte Wert zugewiesen. Nach Abschluß dieses Programmteils befindet sich der größte Wert in der Ebene E1 (siehe Abb. 3).

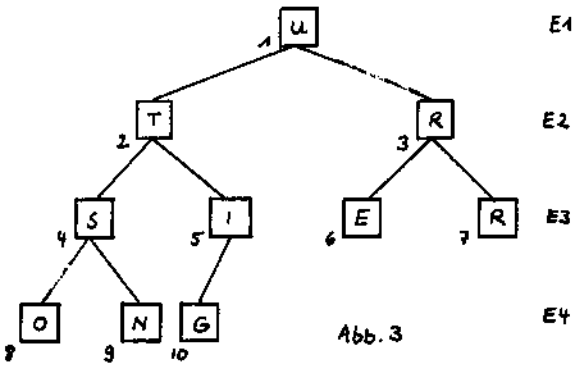


Abb. 3

Im 2.Teil wird nun der 1.Wert an das Ende gelegt und dann nicht mehr beachtet. Der Heap wird nun für die restlichen Felder wieder neu angelegt, der größte Wert an die vorletzte Stelle gebracht und so fort.

```
100  REM  ****************************
110  REM  *  HEAP SORTIEREN          *
120  REM  ****************************
130 N = 10
140  REM  * N = ANZAHL DER ELEMENTE
150  DIM N$(N)
160  FOR I = 1 TO N
170  PRINT "NR. ";I; TAB( 8);
180  INPUT N$(I)
190  NEXT I
200  GOSUB 260
210  REM  * ZUM UNTERPROGRAMM SORTIEREN
220  FOR I = 1 TO N
```

```
230  PRINT N$(I)
240  NEXT I
250  END
260  REM  * UNTERPROGRAMM HEAPSORTIEREN
270  REM  * -- TEIL 1 --
280 M = N
290  FOR L =  INT (N / 2) TO 1 STEP  - 1
300 H$ = N$(L)
310  GOSUB 410
320  NEXT L
330  REM  * -- TEIL 2 --
340 L = 1
350  FOR M = N - 1 TO 1 STEP  - 1
360 H$ = N$(M + 1)
370 N$(M + 1) = N$(1)
380  GOSUB 410
390  NEXT M
400  RETURN
410  REM  * -- HEAP ANLEGEN --
420 I = L
430 J = I + I
440  IF J > M GOTO 510
450  IF J = M GOTO 470
460  IF N$(J + 1) > N$(J) THEN J = J + 1
470  IF H$ >  = N$(J) THEN 510
480 N$(I) = N$(J)
490 I = J
500  GOTO 430
510 N$(I) = H$
520  RETURN
```

NASPEN-Zusatz

von Günter Böhm

Das folgende kleine Programm soll es ermöglichen, unformatierte Texte in den Textspeicher von NASPEN VS.1 einzulesen. Unter unformatiert verstehe ich einfach eine Reihe von ASCII Zeichen ohne Angabe von Speicherbereich oder besondere Kontrollzeichen. Solche Texte können z.B. durch das Formatierprogramm (Heft 8-81) oder das Lesen vom Bildschirm (Heft11/12-81) generiert werden. Zunächst macht man einen Kaltstart von NASPEN bei B8Ø0. Dann wird das Einleseprogramm bei E0Ø gestartet. Die grüne Drive-LED leuchtet auf als Zeichen, daß der Text eingelesen werden kann. (Wer den Cassettenrecorder automatisch ansteuert, muß eben rechtzeitig die Textcassette einlegen. Der Text wird während des Lesens auf dem Bildschirm ausgegeben. Nach Textende (@ muß als einziges Kontrollzeichen auf derCassette sein.) verlischt die LED, und das Programm macht automatisch einen Warmstart von NASPEN. Falls sich am Textbeginn vom Start des Einlesens noch Zufallszeichen befinden sollten oder der Schirm gar dunkel bleibt, kann man durch mehrmaliges Drücken der Space-Taste bis zum richtigen Textanfang vorrücken und die überflüssigen Zeichen mit DELETE "d" entfernen. Der Text steht nun für weitere Verarbeitung mit NASPEN zur Verfügung. Die Zeilen 25Ø bis 27Ø laden die Adresse der Druckerroutine in die entsprechenden Speicherzellen. (Bei mir beginnt sie in #C8Ø) Sie müssen bei Bedarf verändert oder weggelassen werden.

```
                    READ-NASPEN
              0010 ; EINLESEPROGRAMM FUER
              0020 ; UNFORMATIERTE TEXTE
              0030 ; IN NASPEN-TEXTPUFFER
              0040 ; 4.1.82/G.BOEHM,K'HE
              0050 ; START E00, DANN EIN-
              0060 ; LESEN DER CASSETTE
              0070 ;
0E00          0080        ORG  #E00
0E00 212010   0090 START  LD   HL,#1020;PUFFERSTART
0E03 DF5F     0100        DEFW #5FDF ; MFLP
0E05 CF       0110 READ   RST  8 ;RIN
0E06 FE40     0120        CP   #40; ENDZEICHEN?
0E08 2805     0130        JR   Z END
0E0A 77       0140        LD   (HL),A
0E0B 23       0150        INC  HL
0E0C F7       0151        RST  #30
0E0D 18F6     0160        JR   READ
0E0F DF5F     0170 END    DEFW #5FDF; MFLP
0E11 221A10   0180        LD   (#101A),HL;TEXTPOINTER
0E14 3E20     0190        LD   A,#20;ENDZ.F.NASPEN
0E16 77       0200        LD   (HL),A
0E17 23       0210        INC  HL
0E18 3EFF     0220        LD   A,#FF
0E1A 77       0230        LD   (HL),A
0E1B 3EC3     0240        LD   A,#C3
0E1D 321D10   0250        LD   (#101D),A ;PRINTER REFL.
0E20 11800C   0260        LD   DE,#C80
0E23 ED531E10 0270        LD   (#101E),DE
0E27 C306B8   0280        JP   #B806 ; WARMST.NASPEN
```

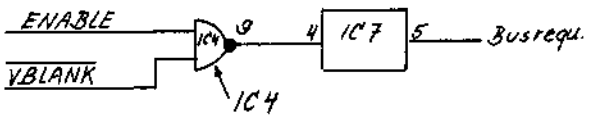
Hochauflösende Grafik

1. Ist es möglich, neben dem 62,5 Hz Signal auch ein 31,25 Hz Signal an IC 1 zu legen, sodaß es 384*256 Bildpunkte gibt?
2. Von welchem IC stammt der zweite Port links unten (der mit Pin 9 am Pin 4/IC 7 verbunden ist)?

W.v.d.Vaart, Waddinxveen/Holland

zu 1. Man kann auch 384*256 Bildpunkte erzielen. Dazu muß man alle Taktfrequenzen vom Videoteil der Reihe nach an die Puffer IC 1 und 2 anschließen, also nicht wie im Schaltplan eine Frequenz auslassen.

zu 2. Es handelt sich um IC 4. Da ich nach meinen Unterlagen nicht sicher bin, ob dies das gefragte IC ist, folgt noch einmal der anscheinend betroffene Schaltungsteil:



Mit freundlichen Grüßen
Ihr H.Martin Pohl

Lottoprogramm
von Eberhard Horch

Folgendes Programm vergleicht die Tippzahlen mit den Gewinnzahlen im Lotto. Nach Eingabe der Gewinnzahlen wird angezeigt, ob man und wie man gewonnen hat. Zu sagen ist nicht viel, das kurze Programm erklärt sich von selbst. Ich habe es mal gemacht, weil ich zu bequem war, die Zahlen immer zu vergleichen. Die Daten in Zeile 40 - 80 enthalten die getippten Zahlen vom Lottoschein und in Zeile 130 ist in der FOR/NEXT Schleife die Anzahl der getipoten Kästchen einzusetzen.

```
10 CLS:PRINTTAB(10)"HABEN WIR HEUTE GEWONNEN?"
20 PRINTTAB(10)"                              "
30 DIM Z(10),D(100)
40 DATA
50 DATA
60 DATA > HIER WERDEN DIE LOTTOZAHLEN
70 DATA  EINGETRAGEN !
80 DATA
90 FOR I=1 TO 7
100 INPUT"GEWINNZAHL=";Z(I)
110 NEXT I
120 CLS:PRINTTAB(10)"*** GEWINNZAHLEN ***":PRINT
130 FOR M=1 TO 10:PRINT"KASTEN";M;"=";:N=0
140 FOR J=1 TO 6
150 READ D(J)
160 FOR K=1 TO 6
170 IF Z(K)=D(J) THEN PRINT Z(K)
180 IF Z(K)=D(J) THEN N=N+1
190 NEXT K
200 NEXT J
210 IF N=6 THEN 320
220 REM  * ABFRAGE ZUSATZZAHL *
230 IF N<5OR N>5 THEN 270
240 FOR J=1 TO 6
250 IF Z(7)=D(J) THEN PRINT"  Z-ZAHL=";Z(7);
260 NEXT J
270 PRINT
280 NEXT M
290 PRINTTAB(12)"GEWINNZAHLEN"
300 PRINTZ(1);Z(2);Z(3);Z(4);Z(5);Z(6);"[";Z(7);"]"
310 END
320 PRINT: REM * SCHLEIFE BLINKEN *
330 FOR V=1 TO 10
340 SCREEN10,M+3:PRINT"                    "
350 FOR W=1 TO 50:NEXT
360 SCREEN 10,M+3:PRINT"HURRA 6 RICHTIGE"
370 FOR W=1 TO 50:NEXT
380 NEXT V
390 GOTO 280
```

Prüfsumme für Nassys 3 von Christian Peter

Dieses Programm gibt einen definierten Speicherbereich mit Prüfsummen aus. Es ist vor allem für die Benutzer von NASSYS 3 gedacht, weil in diesem Betriebssystem vom "T"-Befehl keine Prüfsummen mehr ausgegeben werden. Es läuft aber auch unter NASSYS 1, dadurch kann man auch direkt vom Schirm die Prüfsummen ablesen. Das Programm ist voll verschiebbar.

```
start des programms:
e aaaa ssss eeee
            aaaa: startadresse des tab-programms
            ssss: beginn des
            eeee: ende des speicherbereiches

0000  2a 0e 0c      tab  ld hl,(arg2)
0003  ed 5b 10 0c        ld de,(arg3)
                         hl  kleiner als
0007  b7             tb1  or a            de?
0008  ed 52              sbc hl,de
000a  19                  add hl,de
000b  38 06               jr c tb2
                          wenn hl groesser
000d  ef                  rst prs als de: ende
000e  2e 0d 00            defb '.,cr,0
0011  df 5b               scal mret
                          init checksum
0013  0e 00          tb2 ld c,0
                          output adresse
0015  ef                  rst prs
0016  20 20 00            defb ' ,' ,0
0019  df 66               scal tbcd3
                          output 8 bytes
001b  06 08               ld b,8
001d  7e              tb3 ld a,(hl)
001e  df 67               scal tbcd2
0020  23                  inc hl
0021  df 69               scal space
0023  10 f8               djnz tb3
                          output checksum
0025  79                  ld a,c
0026  df 68               scal b2hex
0028  df 6a               scal crlf
002a  18 db               jr tb1
```

NASSYS 3

NASSYS 3 wird immer mehr entblättert. Neue Informationen zu BØ finden Sie im Leserbrief von Christian Peter. Das Verhalten von INPUT in BASIC wird näher in meinen Anmerkungen zu "Seeschlacht" beleuchtet.
Hier noch einige Bemerkungen von Günter Kreidl:

Inkompatibilitäten mit NAS-SYS 1:
a) Wer das Tabulate-Kommando in seinen Programmen verwendet hat, wie ich z.B. in meinem RELOCATOR (Heft 6/81), der muß vor dem Aufruf der Tabulate-Routine zusätzliche Parameter laden.
b) Wer in seinen Programmen unter NAS-SYS 1 über die Tabellenzeiger $OUT (C73) und $IN (C75) auf die vier NAS-SYS-internen Output- und Input-Tabellen zugegriffen hat, der wird sein blaues Wunder erleben, wenn er diese Programme unter NAS-SYS 3 fährt. Diese Tabellen sind dort genau um -6 Bytes verschoben! Aus diesem Grund ist auch der FORTH-Interpreter nicht ohne Anpassung unter NAS-SYS 3 lauffähig. Die MCODE-Routinen OUTC, OUTD und OUTN müssen angepaßt werden.

Stacküberlauf im Monitorstack
Als ich das ansonsten erheblich verbesserte Tabulate-Kommando mit "breiterem" Ausgabeformat auf meine Schreibmaschine ausgeben ließ (über den U-Befehl), blinkte anschliessend der Cursor nur noch im Minutenabstand. Eine Untersuchung des Workspace zeigte, daß der im Vergleich zu NAS-SYS 1 verkleinerte Monitorstack "übergelaufen" war und die Geschwindigkeitsregister des Cursors KSHORT und KBLINK überschrieben hatte. Das wird man wohl nur vermeiden können, wenn das Treiberprogramm für den Drucker auf einen anderen Stackbereich ausweicht.

Zufallszahlen

von Günter Kreidl

ZEAP Z80 Assembler - Source Listing

```
                        0010 ;PSEUDORANDOM GENERATOR
                        0020 ;NACH H.T.GORDON
                        0030 ;DR. DOBB'S NR. 40
                        0040 ;6502/Z-80-TRANSFORMATION
                        0050 ;VERS. 1.0 ALS UPRO
                        0060 ;G.K. 5.12.81
                        0070 ;DAS PROGRAMM ERZEUGT PSEUDO-
                        0080 ;ZUFALLSZAHLEN ZWISCHEN 0 UND
                        0090 ;255, WOBEI SICH ERST NACH ETWA
                        0100 ;50 MIO. AUFRUFEN DIE GLEICHE
                        0110 ;FOLGE VON 256 ZAHLEN ERGIBT.
                        0120 ;DIE ZAHL WIRD IM AKKU UEBERGEBEN,
                        0130 ;DIE REG. IX UND BC WERDEN VERAENDERT.
0C80                    0140        ORG   #C80
0C80 DD21C70C           0150 MIXSIM LD    IX,MEMEX
0C84 DDE5               0160        PUSH  IX
0C86 DD4E00             0170        LD    C,(IX)
0C89 0D                 0180        DEC   C
0C8A 79                 0190        LD    A,C
0C8B FEFF               0200        CP    #FF
0C8D 2815               0210        JR    Z,RESET
0C8F 0600               0220        LD    B,0
0C91 DD09               0230        ADD   IX,BC
0C93 DD7E02             0240        LD    A,(IX+2)
0C96 B7                 0250        OR    A
0C97 2011               0260        JR    NZ,SIMRND
0C99 DD7E03             0270        LD    A,(IX+3)
0C9C B7                 0280        OR    A
0C9D 280B               0290        JR    Z,SIMRND
0C9F DD7E01             0300        LD    A,(IX+1)
0CA2 181B               0310        JR    DEJUM
0CA4 0E02               0320 RESET  LD    C,2
0CA6 0600               0330        LD    B,0
0CA8 DD09               0340        ADD   IX,BC
0CAA DD7E01             0350 SIMRND LD    A,(IX+1)
0CAD CB27               0360        SLA   A
0CAF CB27               0370        SLA   A
0CB1 37                 0380        SCF
0CB2 DD8E01             0390        ADC   A,(IX+1)
0CB5 DD7701             0400        LD    (IX+1),A
0CB8 DD8604             0410        ADD   A,(IX+4)
0CBB FE80               0420        CP    128
0CBD 3802               0430        JR    C,STOREX
0CBF EE7F               0440 DEJUM  XOR   #7F
0CC1 DDE1               0450 STOREX POP   IX
0CC3 DD7100             0460        LD    (IX),C
0CC6 C9                 0470        RET
0CC7 00                 0480 MEMEX  DEFB  0
0CC8 01                 0490 RND1   DEFB  1
0CC9 63                 0500 RND2   DEFB  99
0CCA C8                 0510 RND3   DEFB  200
0CCB FF                 0520 ADD1   DEFB  255
0CCC 0F                 0530 ADD2   DEFB  15
0CCD F0                 0540 ADD3   DEFB  #F0
                        0550 ;RND1-3 UND ADD1-3 KOENNEN MIT
                        0560 ;BELIEBIGEN STARTWERTEN GELADEN
                        0570 ;WERDEN, WENN DIESE DEN FOLGENDEN
                        0580 ;BEDINGUNGEN GENUEGEN:
                        0590 ;ADD1 µ° 0, ADD2 µ° ADD1
                        0600 ;ADD3 µ° ADD2, ADD3 µ° ADD1
                        0610 ;
                        0620 ;TEST MIXSIM
                        0630 ;ZEIGT 144 PSEUDO-ZUFALLSZAHLEN
                        0640 ;AUF DEM BILDSCHIRM AN  (HEX)
0CCE 0690               0650 TESTM  LD    B,144
0CD0 C5                 0660 LOOPM  PUSH  BC
0CD1 CD800C             0670        CALL  MIXSIM
0CD4 DF                 0680        RST   #18
0CD5 68                 0690        DEFB  #68
0CD6 EF                 0700        RST   #28
0CD7 202000             0710        DEFB  32,32,0
0CDA C1                 0720        POP   BC
0CDB 10F3               0730        DJNZ  LOOPM
0CDD DF                 0740        RST   #18
0CDE 5B                 0750        DEFB  #5B
```

```
0C80 DD 21 C7 0C DD E5 DD 4E|4A
0C88 00 0D 79 FE FF 28 15 06|5A
0C90 00 DD 09 DD 7E 02 B7 20|B6
0C98 11 DD 7E 03 B7 28 0B DD|DA
0CA0 7E 01 18 1B 0E 02 06 00|74
0CA8 DD 09 DD 7E 01 CB 27 CB|B3
0CB0 27 37 DD 8E 01 DD 77 01|DB
0CB8 DD 86 04 FE 80 38 02 EE|D1
0CC0 7F DD E1 DD 71 00 C9 00|20
0CC8 01 63 C8 FF 0F F0 06 90|94
0CD0 C5 CD 80 0C DF 68 EF 20|50
0CD8 20 00 C1 10 F3 DF 5B 0F|11
```

MDCR Interface
Teil3 von J. C. Lotter

UNICON stellt fuer allgemeinen Gebrauch eine Anzahl von nuetzlichen Unterprogrammen zur Verfuegung, die im folgenden beschrieben werden sollen. Die angegebenen UNICON-Adressen beziehen sich auf die Standardversion (Startadresse A000h) und sind fuer andere Startadressen entsprechend umzurechnen.

A004 VCOM
Command Interpreter. Erwartet in 0BFCh die Startadresse einer Befehlsliste.
A0A1 XCALL
Springt zu dem Unterprogramm, dessen Startadresse in der Speicherstelle steht, auf die HL zeigt. Entspricht dem Pseudo-opcode JP (HL).
A0C9 ERROR xx
Gibt die Meldung "*ERROR xx" und einen Piepton aus. Das Byte xx steht hinter dem Aufruf (CD C9 A0 xx ....). Springt anschliessend in die Interpreterschleife.
A124 LINE
Schreibt eine waagerechte Linie auf den Bildschirm. DE wird veraendert.
A138 TIME
Startet die Real time clock. DE zeigt auf die Bildschirmzeile, in der die Zeit (HHMM) steht.
A1BE CLTOP
Loescht die oberste Zeile. HL und A werden veraendert.
A1CB OUTS n1 b1 ....nn bn 00
Port-Ausgabe. Das Byte bx wird an den Port nx ausgegeben. Beendet wird die Serie durch das Byte 00. A wird veraendert.
A1FB BEEP t1 d1 ... tn dn 00
Erzeugt eine Tonfolge, tx bestimmt die Hoehe, dx die Dauer des x-ten Tones. Ist t groesser als 8Fh, entspricht dies einer Pause. Beendet wird die Folge durch 00. A wird veraendert.
A4D3 B2DEC
Dezimalausgabe. E enthaelt das auszugebende Byte. A,B und E werden veraendert.
A27A DEL xx
Verzoegerung um (5 * xx) msec. I wird veraendert.
A506 SAVE
Fuehrt den SAVE-Befehl aus. ARG1 (0C0E) zeigt auf den Filenamen. ARG2(0C10) enthaelt die Startadresse, ARG3 (0C12) die Endadresse. ARG4 (0C14) enthaelt die Execute-Adresse.
A64D UPDATE
Fuehrt den UPDATE-Befehl aus. DE zeigt auf den Filenamen.
A65C VERIFY
Fuehrt den VERIFY-Befehl aus. DE zeigt auf den Filenamen.
A776 LOAD
Fuehrt den LOAD-Befehl aus. DE zeigt auf den Filenamen.
A793 EXECUTE
Wie LOAD, nur wird das Programmfile sofort an der Execute-Adresse gestartet.
A7A6 DIREC
Fuehrt den Directory-Befehl aus.

Damit duerfte es moeglich sein, MDCR-Befehle direkt in andere Programme (Assembler, Editor) einzubauen und deren Cassettenroutinen zu ersetzen.
Noch ein paar Tips: Wenn Sie ein Programm auf MDCR abspeichern, dann geben Sie ruhig noch ein paar Bytes dazu. So koennen Sie das Programm spaeter vergroessern (mit UPDATE), ohne es erst loeschen zu muessen.
Zum Abspeichern eines BASIC-Programms nehmen Sie als Startadresse 1000, als Execute Adresse FFFD. Die Endadresse ermitteln Sie, indem Sie (in BASIC) eintippen:
?xxxxx-FRE(0)
xxxxx ist Ihre hoechste Speicheradresse. Die so gefundene Endadresse mussen Sie noch ins Hex-System umrechnen.

-Falls Sie der ungluckliche Besitzer eines NASCOM 2 sind, wird auf Ihrer Hardware im allgemeinen kein Interruptprogramm laufen. Der Grund: Auf dem Bus zur RAM-Erweiterung liegt die IEI-IEO-Leitung parallel und nicht, wie erforderlich, seriell. Abhilfe: Trennen Sie die Leitung durch oder - falls Sie noch Geld uebrig haben - werfen Sie den Bus weg und kaufen Sie sich ein Motherboard.

Lottozahlen-Generator
von Bernd im Brahm

Der Lottozahlen-Generator liefert auf Tastendruck 6 Zufallszahlen. Durch Betätigung der Tasten "Shift/New Line" (ESC) wird das Programm beendet (Jede andere Taste liefert

weitere 6 Zahlen, Red.). Der benötigte Speicherbereich ist C8Ø - EØE. Bei C8Ø wird das Programm gestartet. Da das Programm den Speicherbereich, in dem es läuft, selbst berechnet, ist es voll verschiebbar und außerdem EPROM-lauffähig. Als Monitor wird NAS-SYS 1 verwendet.

```
TC8Ø EØE
  0C80 EF 0C 00 CD 44 00 2A FE C0
  0C88 0F 11 19 01 19 11 D5 0B D8
  0C90 01 1B 00 ED B0 21 00 00 76
  0C98 DF 58 EF 44 72 75 65 63 BD
  0CA0 6B 65 6E 20 53 69 65 20 4B
  0CA8 65 69 6E 65 20 62 65 6C A8
  0CB0 69 65 62 69 67 65 20 54 95
  0CB8 61 73 74 65 2C 0D 75 6D 8C
  0CC0 20 64 69 65 20 42 65 72 57
  0CC8 65 63 68 6E 75 6E 67 20 DC
  0CD0 7A 75 20 73 74 61 72 74 19
  0CD8 65 6E 2E 0D 44 69 65 20 24
  0CE0 45 69 6E 67 61 62 65 20 B7
  0CE8 76 6F 6E 20 22 53 48 49 6D
  0CF0 46 54 2F 4E 45 57 20 4C 1B
  0CF8 49 4E 45 22 0D 62 65 65 3B
  0D00 6E 64 65 74 20 64 61 73 10
  0D08 20 50 72 6F 67 72 61 6D 0D
  0D10 6D 2E 0D 00 DF 7B FE 1B 38
  0D18 20 04 18 79 18 F6 D7 77 36
  0D20 0D 06 06 E5 ED 5F D9 6F BF
  0D28 EB 44 7D AB 67 78 AA BC D1
  0D30 11 1F CB 19 6F D9 D6 00 6F
  0D38 27 B7 28 E8 FE 50 30 E4 95
  0D40 21 2C 0C C5 01 06 00 ED 5F
  0D48 B1 C1 E1 28 D6 77 23 10 50
  0D50 D2 21 2C 0C 0E 06 22 32 F0
  0D58 0C CB 84 41 05 DD 2A 32 3F
  0D60 0C DD 7E 00 57 DD 5E 01 67
  0D68 93 30 08 DD 73 00 DD 72 DF
  0D70 01 CB C4 DD 23 10 EA CB D2
  0D78 44 20 DE 06 06 EF 20 20 02
  0D80 00 DD 7E 00 DF 68 DD 2B 37
  0D88 EF 20 20 20 20 20 00 10 34
  0D90 F0 DF 6A 18 87 18 30 18 D5
  0D98 21 00 00 00 00 00 00 2A F0
  0DA0 2A 4C 4F 54 54 4F 5A 41 04
  0DA8 48 4C 45 4E 20 2D 20 47 90
  0DB0 45 4E 45 52 41 54 4F 52 1D
  0DB8 2A 2A 21 2C 0C E5 AF 06 0C
  0DC0 06 77 10 FD E1 C9 00 DF E0
  0DC8 4E AF 11 10 00 2A 29 0C 52
  0DD0 19 22 29 0C D7 25 2A 29 9C
  0DD8 0C 19 22 29 0C EF 2A 20 9A
  0DE0 56 49 45 4C 20 45 52 46 1A
  0DE8 4F 4C 47 21 20 2A 0D 00 4F
  0DF0 2A 29 0C 19 22 29 0C D7 A3
  0DF8 02 DF 5B EF 2A 2A 2A 2A D8
  0E00 2A 2A 2A 2A 2A 2A 2A 2A 5E
  0E08 2A 2A 2A 2A 0D 00 C9 00 94
```

Irrgarten

von Clemens Ballarin (12 J.)

Das "Nascom-2-Graphik-Männchen" sucht seinen Weg durch ein vom Spieler auf dem Bildschirm selbst gezeichnetes Labyrinth; der Kopf des Männchens zeigt dabei immer in die Laufrichtung. Startpunkt "S" und Zielpunkt "E" können beliebig festgelegt werden.

Das Programm wurde auf Nascom 1 (NAS-SYS 1) erstellt, der auf Nascom-2-Graphik erweitert wurde. Eine erweiterte Tastatur ist empfehlenswert.

Nach dem Start des Programms mit E DØ8 erscheint etwa in der Mitte des Bildschirms ein blinkender Cursor, dessen Blinkfrequenz durch die Inhalte der Speicherstellen D2EH (Cursor an) und D3CH (Cursor aus) festgelegt ist. Mit den Pfeiltasten läßt sich der Cursor verschieben, mit "/" ein Klötzchen, mit "S" der Startpunkt und mit "E" der Endpunkt (Ziel) des Labyrinths plazieren.

Nun kann man ein einfaches oder schwieriges Labyrinth konstruieren. Die Mauern werden aus den Klötzchen zusammengesetzt. Hat man ein Zeichen falsch gesetzt, kann man es mit "Space" wieder löschen. Ist der Irrgarten fertig, kann man mit "New Line" den Eingabemodus verlassen und den Computer zum Suchen des Endpunktes veranlassen. Wurde jedoch der Startpunkt nicht plaziert, so kommt der Computer in die Eingaberoutine zurück. Andernfalls erscheint jetzt am Startpunkt das Männchen, durchläuft die Gänge des Labyrinths und erreicht früher oder später den Endpunkt, wenn:

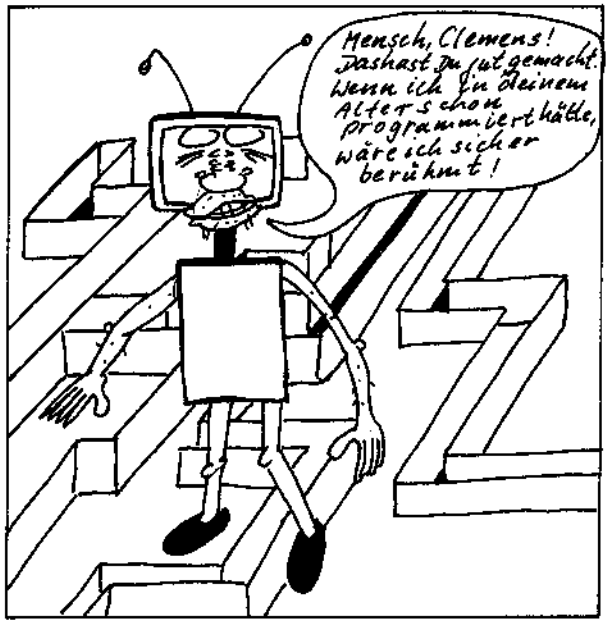
a) der Endpunkt plaziert ist,
b) es in keine unendliche Schleife gerät,
c) der Endpunkt mit dem Startpunkt durch einen Gang verbunden ist.

Die Geschwindigkeit, mit der das Männchen durch das Labyrinth geht, wird durch den Inhalt der Speicherzelle E1DH bestimmt.

Hat das Männchen den Endpunkt erreicht, kann durch "Backspace" das Programm verlassen, oder durch Drücken einer beliebigen anderen Taste der Eingabemodus aufgerufen werden.

```
  0D00 DF 0D 02 01 1F 0E 13 0E 4A
  0D08 31 08 0D 21 00 08 11 01 96
  0D10 08 01 FF 03 36 20 ED B0 1B
  0D18 21 24 0E 11 D7 0B 01 15 81
  0D20 00 ED B0 21 E5 09 01 40 1A
  0D28 00 C5 56 36 5F 0E 01 06 FA
  0D30 00 DF 62 38 13 10 FA 0D E0
  0D38 20 F7 72 0E 01 DF 62 38 56
  0D40 07 10 FA 0D 20 F7 18 E3 7D
  0D48 72 C1 FE 11 20 03 2B 18 FD
  0D50 D8 FE 12 20 03 23 18 D1 74
  0D58 FE 13 20 05 B7 ED 42 18 99
  0D60 C8 FE 14 20 03 09 18 C1 4C
  0D68 FE 2F 20 04 36 FF 18 B9 CC
  0D70 FE 20 20 02 18 04 FE 53 2A
  0D78 20 03 77 18 AC FE 45 20 46
  0D80 02 18 F7 FE 0D 20 A2 21 8C
  0D88 00 08 01 FF 03 3E 53 ED 1E
  0D90 B1 E2 33 0D 2B E5 DD E1 3E
```

```
0D98 36 B7 11 40 00 06 00 3E 27
0DA0 FF DD BE FF 28 40 DD 36 C1
0DA8 00 20 DD 2B CD FD 0D DD 91
0DB0 36 00 B5 CD 1C 0E DD BE 3A
0DB8 C0 28 E6 DD 36 00 20 11 D7
0DC0 C0 FF DD 19 11 40 00 CD A0
0DC8 FD 0D DD 36 00 B7 CD 1C 92
0DD0 0E DD BE 01 28 E0 DD 36 A2
0DD8 00 20 DD 23 CD FD 0D DD B9
0DE0 36 00 B6 CD 1C 0E DD BE 6B
0DE8 40 28 E6 DD 36 00 20 DD 53
0DF0 19 CD FD 0D DD 36 00 B7 B7
0DF8 CD 1C 0E 18 A4 DD 7E 00 13
0E00 FE 45 3E FF C0 D1 DD 36 32
0E08 00 B7 CF FE 08 20 04 CD 93
0E10 1C 0E C7 DD 36 00 45 36 9D
0E18 53 C3 23 0D 3E 20 FF 10 D9
0E20 FB 3E FF C9 2A 20 2A 20 C3
0E28 2A 20 49 52 52 47 41 52 47
0E30 54 45 4E 20 2A 20 2A 20 D9
0E38 2A 20 00 00 00 00 00 00 90
```



ROM BASIC V 4.7

von Günter Böhm

Wer das NASCOM ROM BASIC V4.7 besitzt und meint, die Programme, die dafür geschrieben sind, würden auch damit laufen, kann sein blaues Wunder erleben. Es gibt nämlich (mindestens) zwei Versionen!! Nach vielen vergeblichen Bemühungen meinerseits, BASIC Programme von anderen Lesern zu benutzen,und der Erkenntnis, daß das Problem nicht an meiner (Selbstbau-) Speichererweiterung liegen kann, kam mir der Verdacht, mit meinem BASIC sei "etwas faul". Und siehe da: beim Vergleich mit der Version eines Lesers stellte sich der Unterschied heraus.

Meine Version:

```
E000 F3 DD 21 00 00 C3 12 E0 8B E9
E00A F2 F0 CD DF E4
E32E E7 FF
E777 0C
FFFE 0C
```

Die "andere" Version:

```
E000 C3 03 E0 F3 DD 21 00 00 C3 12 E0 8B E9
E00D F2 F0
E32E 07 E6
E777 B1
FFFE B1
```

Ein genauer Vergleich zeigt, daß sich hier keine Bugs beim Schießen der Eproms eingeschlichen haben, sondern daß hier bewußt eine andere Version programmiert wurde. So etwas hätte man einem ja auch sagen können! Das Problem trat bei mir immer dann auf, wenn Tastatureingaben über ein Maschinenprogramm gemacht wurden (anstelle der fehlenden INKEY Funktion), das einen Parameter über die Routine in #F0F2 an BASIC übergibt. In meinem Manual war als Adresse, die wiederum die Adresse dieser Routine enthält, #E00A angegeben. In den oben erwähnten "Problemprogrammen"(zu denen übrigens alle Programme des INMC BASIC Buches gehören!) war aber immer ein Sprung zu (#E00D) vorgesehen. Vergleicht man nun die Inhalte von #E00A und #E00D in beiden Versionen, stellt man fest, daß beide einen Sprung zur gleichen Adresse bewirken.

So lassen sich die Programme leicht umändern; leichter wäre es allerdings gewesen, wenn die Hersteller vorher auf die Inkompatibilität (ich mag dieses Wort!) hingewiesen hätten. (Der Grund für die zweite Version ist mir überdies nicht einleuchtend.)

Falls Sie also auch unglücklicher Besitzer meines Minderheiten-BASIC sind, ändern Sie folgenden Wert bei der Keyboardroutine (Bisher wurde immer die selbe verwendet. Wer hat sie wohl als erster geschrieben?):

```
DATA .....,3370,-5664  LD HL,#E00D/JP (HL)
DATA .....,2602,-5664  LD HL,#E00A/JP (HL)
```

Ich wäre sehr daran interessiert, von Lesern zu hören, die ebenfalls letztere Version besitzen. Vielleicht sind auch schon andere Probleme mit den übrigen Adressenunterschieden aufgetaucht. Zumindest würde ich gerne wissen, wieviele dieser "Sonderausführungen" unter unseren Lesern herumgeistern.

Seeschlacht

von Klaus Mombaur

Sie haben für das Spiel 20 Schiffe zur Verfügung, nämlich: 3 Zerstörer über je 3 Felder, 7 Minensucher über je 2 Felder und 10 U-Boote auf je einem Feld. Der Computer verfügt über die gleiche Flottenzusammenstellung.

Zu Beginn verteilen Sie Ihre Schiffe über ein Spielfeld von 144 Feldern, natürlich für den Computer unsichtbar. Diese Arbeit nimmt Ihnen der Computer aber auf Wunsch ab! Er verteilt dann die Schiffe per Zufall, aber nach den Spielregeln, und ebenso seine Schiffe, für Sie unsichtbar.

In der folgenden Seeschlacht hat jeder abwechselnd einen Schuß auf das gegnerische Feld. Nach einem Treffer darf weitergeschossen werden, bis nichts mehr getroffen wird. Man braucht weder Papier noch Bleistift; alles wird vom Computer registriert und angezeigt. Das Programm erklärt sich weitgehend von selbst. Es benötigt 16 K RAM (C80 4FFF) und läuft mit NASSYS.

```
100 REM Seeschlacht
110 REM
120 REM  Copyright 2/81 by Klaus Mombaur
125 REM  [illegible], [illegible] Nuernberg
130 REM
131 CLS
132 A1=3360
134 FORA=1TO692:POKEA1,46:A1=A1+1:NEXT
135 CLS: SCREEN 3,4:
136 PRINT"    Wir spielen  -  Schiffe versenken
 -"
140 PRINT:PRINT
141 PRINT"          Geben Sie mir Ihren Namen:"
142 INPUT"          (maximal 8 Buchstaben)  ";N1
$
143 A1=LEN(N1$):IFA1>8THEN135
144 DOKE4044,DEEK(2541)
145 DOKE4046,DEEK(2543)
146 DOKE4048,DEEK(2545)
147 DOKE4050,DEEK(2547)
150 PRINT:PRINT
155 PRINT"     Wollen Sie Ihre Schiffe selbst l
egen?"
160 INPUT "                          (J/N)";J$
162 IFJ$="J"THEN1000
164 IFJ$<>"N"THEN135
165 CLS:PRINT:PRINT:PRINT
166 PRINT:PRINT"     Dann bitte ich nun um etwa
s Geduld!"
168 A!=3743
170 FORA=1TO144:POKEA1,153:A1=A1+1:NEXT
176 GOTO1006
180 A1=3888:S=3743:A=1
185 FORC=1TO144
190 POKES,PEEK(A1)
195 A1=A1+1:S=S+1:A=A+1
200 IFA=13THENA=1:A1=A1+1
210 NEXT
220 J$="F"
300 GOTO 1000
305 CLS: PRINT
308 PRINT"Jetzt verlegen Sie Ihre Schiffe!"
309 PRINT
310 PRINT:PRINT"Jedes Schiff muss ringsum frei
sein!"
311 PRINT"Es darf senkrecht oder waagerecht lie
gen."
312 PRINT"Die Felder eines Typs aneinanderlegen
."
313 PRINT"Jedes Feld wird einzeln eingegeben!"
314 PRINT"Immer erst den Buchstaben, dann die Z
ahl!"
315 PRINT
316 PRINT"Nach jeder Eingabe: Taste  ENTER  tip
gen!!!
317 PRINT
320 INPUT"                          weiter?  ";N0$
380 CLS
400 GOSUB500:GOTO590
500 PRINT"    A B C D E F G H I J K L"
502 PRINT" 1 . . . . . .  Verlegen
Sie bitte"
504 PRINT" 2 . . . . . ."
506 PRINT" 3 . . . . . ."
508 PRINT" 4 . . . . . .  3 Zerstoe
rer"
510 PRINT" 5 . . . . . .       = Z
Z Z"
512 PRINT" 6 . . . . . ."
514 PRINT" 7 . . . . . .  7 Minensu
cher"
516 PRINT" 8 . . . . . .       = M
M"
518 PRINT" 9 . . . . . ."
520 PRINT"10 . . . . . .  10 U - Bo
ote"
522 PRINT"11 . . . . . .       = U"
524 PRINT"12 . . . . . ."
530 RETURN
590 SCREEN29,1:PRINTN1$;":"
595 CLEAR
600 REM
605 B=0:D=2:F=0:S$="Zerstoerer"
610 REM
615 B=B+1:SCREEN 1,14
616 PRINT"Eingabe: ";B;". ";S$;":";
620 SCREEN 38,14:INPUT Z0$
625 GOSUB 5070
628 GOSUB5140
630 REM
632 X=90
635 POKE2125+PV+PH,X
638 GOSUB5050
641 G=2
642 GOSUB6000
648 IFB=3THEN680
650 GOTO610
680 GOSUB5500
700 REM
705 B=0:D=3:F=0:S$="Minensucher"
710 REM
715 B=B+1:SCREEN 1,14
716 PRINT"Eingabe: ";B;". ";S$;":";
720 SCREEN 38,14:INPUT M0$
725 GOSUB5070
728 GOSUB5140
730 REM
732 X=77
735 POKE2125+PV+PH,X
738 GOSUB5050
741 G=1
742 GOSUB6000
748 IFB=7THEN780
750 GOTO 710
780 GOSUB5500
800 REM
805 B=0:D=4:S$="U-Boot"
810 REM
815 B=B+1:SCREEN 1,14
816 PRINT"Eingabe: ";B;". ";S$;":";
820 SCREEN 38,14:INPUT U0$
825 GOSUB5070:GOSUB5140
830 REM
832 X=65
835 POKE 2125+PV+PH,X
838 GOSUB 5050
840 IFB=5THENGOSUB5500
841 IFB=8THENGOSUB5500
842 IFB=9THENGOSUB5500
845 IFB=10THEN880
850 GOTO 810
880 GOSUB 5500
890 SCREEN1,14
892 PRINT"Ich speichere nun Ihr Spielfeld!
 ";
900 REM Kopieren
910 A1=3743: S=2125
920 FORA=1TO144:POKEA1,PEEK(S):A1=A1+1:S=S+2
950 GOSUB 6200
960 NEXT
970 GOTO 1500
1000 REM
1001 CLS:SCREEN 1,6:PRINT
1002 PRINT"              Etwas Geduld"
1003 PRINT:PRINT
1004 PRINT"              Ich verlege meine Schiffe
1005 GOTO9000
```

```
1006 A=1:B=3888
1007 POKEB,153:A=A+1:B=B+1
1008 IFA=156THEN1010
1009 GOTO1007
1010 A=1:B=3887
1012 POKEB,170:A=A+1:B=B+13
1014 IFA=14THEN1130
1016 GOTO1012
1130 REM
1135 Z0=3:ZU=2
1140 GOSUB 6500
1150 H=PEEK(3888+T2)
1160 IF H<>153THEN 1140
1170 A=RND(1):A=INT(A+.5)
1190 IFA=0THENZA=1:GOTO1200
1195 ZA=13
1200 K=3888+T2:K1=K+ZA:K2=K+2*ZA
1202 GOSUB 6750
1205 GOSUB 6650
1210 POKE K,90:POKEK1,90:POKEK2,90
1215 Z0=Z0-1:IF Z0=0THEN1230
1220 GOTO1140
1230 REM
1235 M0=7:ZU=3
1240 GOSUB 6500
1250 H=PEEK(3888+T2)
1260 IF H<>153THEN 1240
1270 A=RND(1):A=INT(A+.5)
1290 IFA=0THENZA=1:GOTO1300
1295 ZA=13
1300 K=3888+T2:K1=K+ZA
1302 GOSUB 6750
1305 GOSUB 6700
1310 POKE K,77:POKEK1,77
1315 M0=M0-1:IF M0=0THEN1330
1320 GOTO1240
1330 REM
1335 U0=10:ZU=4
1340 GOSUB 6500
1350 H=PEEK(3888+T2)
1360 IFH<>153THEN1340
1400 K=3888+T2
1405 GOSUB6750
1410 POKEK,85
1415 U0=U0-1:IFU0=0THEN1430
1420 GOTO1340
1430 IFJ$="J"THEN305
1440 IFJ$="N"THENCLEAR:GOTO180
1500 REM
1501 CLEAR
1502 S1=0:S2=0
1503 U1=10:M1=7:Z1=3
1504 U2=10:M2=7:Z2=3
1505 Z3=0:M3=0:Z4=0:M4=0:D=5
1506 N2$=CHR$(PEEK(4044))
1507 N3$=CHR$(PEEK(4045))
1508 N4$=CHR$(PEEK(4046))
1509 N5$=CHR$(PEEK(4047))
1510 N6$=CHR$(PEEK(4048))
1511 N7$=CHR$(PEEK(4049))
1512 N8$=CHR$(PEEK(4050))
1513 N9$=CHR$(PEEK(4051))
1514 N1$=N2$+N3$+N4$+N5$+N6$+N7$+N8$+N9$
1515 CLS:PRINT:PRINT:PRINT
1520 PRINT"          Nun beginnen wir den Kampf!
"
1522 PRINT:PRINT
1524 PRINT"   Nach jeder Eingabe: Taste ENTER t
ippen!
1530 PRINT:PRINT
1540 INPUT"          Wollen wir ?";J$
1550 GOTO2000
1570 CLS:GOSUB500
1575 DOKE3201,3504
1580 DOKE4100,3200:A1=USR(2)
1585 DOKE4100,3648:A=USR(3)
1590 GOSUB7000
1600 IFZ2+M2+U2=0THEN8000
1605 REM
1610 SCREEN 1,14
1620 PRINT"Wohin wollen Sie schiessen? "
1630 SCREEN 38,14: INPUT SF$
1635 A2=LEN(SF$):IFA2>3THENSCREEN38,14:PRINT"
    ";GOTO1630
1640 GOSUB 5070
1650 GOSUB 5149
1660 GOSUB 5050
1700 GOSUB 7100
1710 S1=S1+1:PA=PW*12
1720 VG=PEEK(3875+PA+PS+(PW-1))
1724 IF VG=90 THEN1850
1726 IF VG=77 THEN1900
1728 IF VG=85 THEN1950
1730 X=14
1750 POKE 2125+PV+PH,X
1760 REM
1770 POKE3491+PA+PS,X
1790 GOTO 2000
1850 REM
1860 X=90
1870 GOSUB7200
1874 Z4=Z4+1
1876 IF Z4=3 THENZ2=Z2-1:Z4=0
1880 POKE 2125+PV+PH,X
1885 POKE3491+PA+PS,X
1890 GOTO 1590
1900 REM
1910 X=77
1920 GOSUB7200
1924 M4=M4+1
1926 IF M4=2 THENM2=M2-1:M4=0
1930 POKE 2125+PV+PH,X
1935 POKE3491+PA+PS,X
1940 GOTO 1590
1950 REM
1960 X=85
1970 GOSUB7200
1974 U2=U2-1
1980 POKE 2125+PV+PH,X
1985 POKE3491+PA+PS,X
1990 GOTO 1590
2000 REM
2001 FORA=1TO1000:NEXT
2005 CLS
2008 GOSUB500
2010 REM
2012 DOKE3201,3360:DOKE4100,3200:A2=USR(3)
2020 GOSUB7300
2024 IFDG=1THENIFZ3=1THENONI1GOTO7520,7500
2026 IFDG=2THENIFZ3=1THENIFNW=1THENDG=7:ZB=ZB+1
:ZZ=ZZ-1
2027 IFDG=2THENIFZ3=1THENIFNS=1THENDG=7:ZZ=ZZ+1
:ZB=ZB-1
2030 IFZ1+M1+U1=0 THEN 8100
2040 SCREEN 1,14:PRINTCHR$(27);
2050 PRINT"      Ich schiesse auf:";
2056 IF Z3>0 THEN 2062
2057 IF M3>0 THEN 2062
2058 DG=0:TR=1:G2=0:G3=0:G4=0:G9=0:G0=0
2059 NW=0:NS=0
2060 GOSUB 7400
2062 GOTO 7900
2063 K0=PEEK(2125+PH+PV)
2064 IFK0<>46THEN 2247
2065 GOSUB5120
2066 ONI1GOTO7950,4950
2075 FORW=1TO1000:NEXT:DG=DG+1
2080 SCREEN38,14:PRINTCHR$(ZB);ZZ
2085 FORW=1TO1000:NEXT
2090 S2=S2+1:ZR=ZB-64
2095 SCREEN 30,13
2100 GOSUB 7100
2110 VG=PEEK(3730+ZZ*12+ZR)
2122 IFVG=90THENTR=TR+1:GOTO2250
2124 IFVG=77THENTR=TR+1:GOTO2300
2126 IFVG=85THEN2350
2150 POKE3347+ZZ*12+ZR,14
2160 POKE2125+PV+PH,14
2170 FORW=1TO1500:NEXT
2180 IFTR>1THENONI1GOSUB7600,4600
2190 GOTO1570
2247 IFTR=1THEN2060
2248 DG=DG+1:ONI1GOSUB7600,4600
2249 GOTO2020
2250 REM
2255 GOSUB7800
2260 X=90
2270 GOSUB7200
2275 POKE2125+PV+PH,X
2280 POKE3347+ZZ*12+ZR,X
2284 Z3=Z3+1
2286 IF Z3=3THENZ1=Z1-1:Z3=0:TR=TR-1: GOTO2020
2290 ONI1GOSUB7600,4600
2295 GOTO2020
2300 REM
2305 GOSUB7800
2310 X=77
2320 GOSUB7200
2325 POKE2125+PV+PH,X
2330 POKE3347+ZZ*12+ZR,X
2334 M3=M3+1
2336 IF M3=2THENM1=M1-1:M3=0:TR=TR-1:GOTO2020
2340 ONI1GOSUB7600,4600
2345 GOTO 2020
2350 REM
2360 X=85
2365 GOSUB7200
2370 POKE2125+PV+PH,X
2390 POKE3347+ZZ*12+ZR,X
2394 U1=U1-1
2395 GOTO 2020
4600 REM
4610 IFDG=1THENZZ=ZZ+1:RETURN
4620 IFDG=2THENZZ=ZZ-2:RETURN
```

```
4622 IFG2+G3=0THEN4624
4623 GOTO4626
4624 IFDG=3THENDG=7:ZZ=ZZ+1:GOTO4670
4625 IFDG>7THEN4680
4626 IFG3=0THEN4628
4627 GOTO4630
4628 IFDG=3THENDG=5:ZZ=ZZ-2
4629 IFDG=4THENDG=6:ZZ=ZZ+4
4630 IFDG=3THENZZ=ZZ-1:RETURN
4635 IFDG=4THENIFG4=0THENZZ=ZZ-1:DG=5
4640 IFDG=4THENZZ=ZZ-1:RETURN
4650 IFDG=5THENZZ=ZZ+5:RETURN
4660 IFDG=6THENZZ=ZZ+1:RETURN
4670 IFDG=7THENZB=ZB-1:RETURN
4680 IFDG=8THENZB=ZB+2:RETURN
4685 IFG9=0THEN4688
4686 GOTO4690
4688 IFDG=9THENDG=11:ZB=ZB+2
4689 IFDG=10THENDG=12:ZB=ZB-4
4690 IFDG=9THENZB=ZB+1:RETURN
4695 IFG0=0THEN4697
4696 GOTO4700
4697 IFDG=10THENZB=ZB+1:DG=11
4700 IFDG=10THENZB=ZB+1:RETURN
4710 IFDG=11THENZB=ZB-5:RETURN
4720 IFDG=12THENZB=ZB-1:RETURN
4722 IFDG>12THEN8400
4950 REM
4951 IF PH<64 THENK6=50:K2=50:K8=50
4952 IF PH>640THENK5=50:K1=50:K7=50
4953 IF PV<2 THEN K6=50:K3=50:K5=50
4954 IF PV>20THEN K8=50:K4=50:K7=50
4955 IF DG=0 THEN 4971
4956 IFDG=1THENK2=50:GOTO4971
4957 IFDG=2THENK1=50:GOTO4971
4958 IFDG=3THENK1=50:GOTO4971
4959 IFDG=4THENK1=50:GOTO4971
4960 IFDG=5THENK2=50:GOTO4971
4961 IFDG=6THENK2=50:GOTO4971
4962 IFDG=7THENK4=50:GOTO4971
4963 IFDG=8THENK3=50:GOTO4971
4964 IFDG=9THENK3=50:GOTO4971
4965 IFDG=10THENK3=50:GOTO4971
4966 IFDG=11THENK4=50:GOTO4971
4971 IFK1>76THEN2247
4972 IFK2>76THEN2247
4973 IFK3>76THEN2247
4974 IFK4>76THEN2247
4975 IFK5>76THEN2247
4976 IFK6>76THEN2247
4977 IFK7>76THEN2247
4978 IFK8>76THEN2247
4979 GOTO2075
5050 POKE2929,32:POKE2930,32:POKE2931,32
5060 RETURN
5070 PV=PEEK(2929)
5080 PH=DEEK(2930)
5090 IFPH<8250THENPH=PH-8240
5095 IFPH>8249THENPH=(PH-12081)/256+9
5100 IFPV<65THEN5108
5102 IFPV>76THEN5108
5104 IFPH<1THEN5108
5106 IFPH>12THEN5108
5107 GOTO5110
5108 GOSUB 5600
5109 B=B-1:ON D GOTO510,610,710,810,1610
5110 PS=(PV-64):PW=PH:PV=(PV-65)*2:PH=(PH-1)*64
5118 K0=PEEK(2125+PH+PV)
5120 K1=PEEK(2125+PH+PV+64)
5121 K2=PEEK(2125+PH+PV-64)
5122 K3=PEEK(2125+PH+PV-2)
5123 K4=PEEK(2125+PH+PV+2)
5124 K5=PEEK(2125+PH+PV+62)
5125 K6=PEEK(2125+PH+PV-66)
5126 K7=PEEK(2125+PH+PV+66)
5127 K8=PEEK(2125+PH+PV-62)
5129 REM
5130 IFPH>703 THEN K1=0:K5=0:K7=0
5135 RETURN
5140 REM
5141 IFK1>76THEN5151
5142 IFK2>76THEN5151
5143 IFK3>76THEN5151
5144 IFK4>76THEN5151
5145 IFK5>76THEN5151
5146 IFK6>76THEN5151
5147 IFK7>76THEN5151
5148 IFK8>76THEN5151
5149 IFK0<>46THEN5154
5150 RETURN
5151 GOSUB5200
5152 B=B-1:ONDGOTO510,610,710,810,1610
5154 GOSUB5400
5155 B=B-1:ONDGOTO510,610,710,810,1610
5200 SCREEN28,14:PRINT"   Zu nah!     ";
5210 FORA=1TO1500:NEXT
5220 PRINTCHR$(27);
5230 RETURN
5400 SCREEN 30,14:PRINT"Schon besetzt!!  ";
5410 FORA=1TO1500:NEXT
5420 PRINTCHR$(27);
5430 RETURN
5500 REM
5505 SCREEN 1,14
5506 PRINTCHR$(27);
5508 PRINT"Korrektur? (Platzmangel?/Fehler?) (J
/N)";
5510 INPUT J$
5512 IF J$="N"THEN 5580
5514 IF J$<>"J"THEN5500
5518 SCREEN 1,14
5520 PRINTCHR$(27);
5522 PRINT"Geben Sie das Korrekturfeld!";
5524 SCREEN38,14:INPUT KF$
5526 PV=PEEK(2929):PH=DEEK(2930)
5530 IF PH<8250 THEN PH=PH-8240
5532 IF PH>8249 THEN PH=(PH-12081)/256+9
5534 PV=(PV-65)*2: PH=(PH-1)*64
5535 IFPV<0THEN5500
5536 IFPV>22THEN5500
5537 IFPH<0THEN5500
5538 IFPH>704THEN5500
5540 SCREEN 1,14
5541 PRINTCHR$(27);
5542 PRINT"Nun das Zeichen (Punkt,U,M,Z):";
5543 SCREEN38,14:INPUT KZ$
5545 IFASC(KZ$)=46THEN5550
5546 IFASC(KZ$)=85THEN5550
5547 IFASC(KZ$)=77THEN5550
5548 IFASC(KZ$)=90THEN5550
5549 GOTO5540
5550 POKE2125+PV+PH,ASC(KZ$)
5560 SCREEN1,14:PRINTCHR$(27);
5562 PRINT"Noch eine Korrektur?  (J/N) ";
5564 INPUT J$
5566 IF J$="N"THEN 5580
5568 IF J$<>"J"THEN 5560
5570 GOTO 5518
5580 SCREEN1,14:PRINTCHR$(27);:RETURN
5600 SCREEN 30,14
5610 PRINT "Falsche Eingabe!!";
5620 FORA=1TO1500:NEXT
5630 PRINTCHR$(27);
5640 RETURN
6000 REM
6010 E=1:S1$="Feld"
6020 E=E+1
6030 F=F+1
6035 SCREEN1,14:PRINTCHR$(27);
6040 PRINT"Eingabe: ";E;". ";S1$;": ";
6050 SCREEN38,14:INPUT F$
6060 PV=PEEK(2929):PH=DEEK(2930)
6080 IF PH<8250 THEN PH=PH-8240
6090 IFPH>8249THENPH=(PH-12081)/256+9
6100 IFPV<65THEN6115
6101 IFPV>77THEN6115
6102 IFPH<1THEN6115
6103 IFPH>12THEN6115
6110 PV=(PV-65)*2: PH=(PH-1)*64
6114 GOTO 6117
6115 GOSUB 5600
6116 F=F-1:GOTO 6030
6117 K0=PEEK(2125+PH+PV)
6118 IF K0<>46 THEN 6120
6119 GOTO 6125
6120 GOSUB 5400
6121 F=F-1:GOTO 6030
6125 POKE 2125+PH+PV,X
6130 GOSUB 5050
6140 G=G-1:IF G=0THEN RETURN
6145 GOTO 6020
6150 RETURN
6200 REM Spielfeld EINGRENZEN
6240 IFS=2149THENS=2189
6242 IFS=2213THENS=2253
6244 IFS=2277THENS=2317
6246 IFS=2341THENS=2381
6248 IFS=2405THENS=2445
6250 IFS=2469THENS=2509
6252 IFS=2533THENS=2573
6254 IFS=2597THENS=2637
6256 IFS=2661THENS=2701
6258 IFS=2725THENS=2765
6260 IFS=2789THENS=2829
6270 RETURN
6500 REM
6505 A=RND(1):A=A*1.55:A=INT(A*100)
6510 IFA>154THEN6505
6515 A1=FRE(1):IFA1<80THENCLEAR:GOTO6530
6520 T2=A:RETURN
6530 CLS:PRINT:PRINT:PRINT
6532 PRINT"Ich finde keine freien Felder mehr "
6533 PRINT"fuer die restlichen Schiffe! Neuanfa
ng!"
```

```
6535 FORA=1TO4000:NEXT:GOTO150
6650 REM
6660 H1=PEEK(K1):C0=K1:GOSUB 6900
6670 H2=PEEK(K2):C0=K2:GOSUB 6900
6680 IFH1+H2=306THENRETURN
6690 GOTO1140
6700 REM
6710 H1=PEEK(K1):C0=K1:GOSUB 6900
6720 IFH1=153THENRETURN
6730 GOTO1240
6750 REM
6760 H1=PEEK(K-14)
6761 IFK-14<3888THEN6770
6762 IFH1=153THEN6770
6764 IFH1=170THEN6770
6766 GOTO6840
6770 H2=PEEK(K-12)
6771 IFK-12<3888THEN6780
6772 IFH2=153THEN6780
6774 IFH2=170THEN6780
6776 GOTO6840
6780 H3=PEEK(K+14)
6781 IFK+14>4042THEN6790
6782 IFH3=153THEN6790
6784 IFH3=170THEN6790
6786 GOTO6840
6790 H4=PEEK(K+12)
6791 IFK+12>4042THEN6800
6792 IFH4=153THEN6800
6794 IFH4=170THEN6800
6796 GOTO6840
6800 H5=PEEK(K-1)
6801 IFK-1<3888THEN6810
6802 IFH5=153THEN6810
6804 IFH5=170THEN6810
6806 GOTO6840
6810 H6=PEEK(K+1)
6811 IFK+1>4042THEN6820
6812 IFH6=153THEN6820
6814 IFH6=170THEN6820
6816 GOTO6840
6820 H7=PEEK(K-13)
6821 IFK-13<3888THEN6830
6822 IFH7=153THEN6830
6824 IFH7=170THEN6830
6826 GOTO6840
6830 H8=PEEK(K+13)
6831 IFK+13>4042THENRETURN
6832 IFH8=153THENRETURN
6834 IFH8=170THENRETURN
6840 ONZUGOTO1040,1140,1240,1340
6900 REM
6905 IFZA=13GOTO6950
6910 C1=PEEK(C0+13):C4=PEEK(C0+14)
6915 C2=PEEK(C0-13):C5=PEEK(C0-12)
6920 C3=PEEK(C0+1)
6925 IFC0+13>4042THENC1=153
6926 IFC0+14>4042THENC4=153
6930 IFC0-13<3888THENC2=153
6931 IFC0-12<3888THENC5=153
6935 IFC0+1>4042THENC3=153
6940 IFC1+C2+C3+C4+C5=765THENRETURN
6945 ONZUGOTO1040,1140,1240
6950 C1=PEEK(C0+1):C4=PEEK(C0+12)
6955 C2=PEEK(C0-1)
6960 C3=PEEK(C0+13):C5=PEEK(C0+14)
6965 IFC0+1>4042THENC1=153
6966 IFC0+12>4042THENC4=153
6970 IFC0-1<3888THENC2=153
6975 IFC0+13>4042THENC3=153
6976 IFC0+14>4042THENC5=153
6980 IFC1+C2+C3+C4+C5=765THENRETURN
6985 ONZUGOTO1040,1140,1240
7000 REM
7003 SCREEN29,2:PRINT"     NASCOM     hat: ";
7004 SCREEN29,4:PRINT"     "
7005 SCREEN29,3:PRINT"                    ";
7010 SCREEN29,5:PRINT"noch";Z2;"Zerstoerer"
7020 SCREEN29,8:PRINT"noch";M2;"Minensucher"
7030 SCREEN29,11:PRINT"noch";U2;"U-Boote"
7045 RETURN
7100 REM
7105 SCREEN1,14:PRINTCHR$(27);
7110 A2=1
7115 FORA=1TO24
7120 SCREEN A2,14
7130 PRINT ""
7140 FORA1=1TO20:NEXT
7160 SCREEN A2,14:PRINT" ";
7165 A2=A2+2:NEXT
7170 RETURN
5142 RETURN
7200 REM
7210 FORA=1TO5
7220 FORA1=1TO10
7230 POKE 2125+PV+PH,X:NEXT
7240 FORA1=1TO10
7250 POKE 2125+PV+PH,32:NEXT
7270 NEXTA
7290 RETURN
7300 REM
7302 SCREEN29,1:PRINT"* * * * "
7303 SCREEN29,2:PRINT"                    ";
7305 SCREEN29,3:PRINTN1$;
7308 SCREEN29,4:PRINT"hat:"
7310 SCREEN29,5:PRINT"noch";Z1;"Zerstoerer"
7320 SCREEN29,8:PRINT"noch";M1;"Minensucher"
7330 SCREEN29,11:PRINT"noch";U1;"U-Boote "
7340 RETURN
7400 REM
7410 A=RND(1)
7411 A=(A+.1)*1.18
7430 ZZ=INT(A*10)
7440 A=RND(1)
7441 A=(A+.1)*1.18
7442 A=INT(A*10)+64
7465 ZB=A
7470 I1=RND(1):I1=INT(I1+.5)+1
7472 IFI1<1THEN7470
7474 IFI1>2THEN7470
7490 RETURN
7500 REM
7501 N=2125+PH+PV
7502 K1=PEEK(N+64):K2=PEEK(N-64)
7504 K3=PEEK(N+128):K4=PEEK(N-128)
7505 K5=PEEK(N+192):K6=PEEK(N-192)
7506 IFPH=0THENK2=0:K4=0:K6=0
7507 IFPH=64THENK4=0:K6=0
7508 IFPH=128THENK6=0
7509 IFPH=640THENK3=0:K5=0
7510 IFPH=576THENK5=0
7511 IFK1+K2=92THEN7515
7512 IFK2+K4=92THENIFK6<76THEN2030
7513 IFK1+K3=92THENIFK5<76THEN2030
7514 DG=7:ZZ=ZZ-1:ZB=ZB-1:GOTO2030
7515 IFK4<15THENNS=1
7516 IFK6>76THENNS=1
7517 GOTO2030
7520 REM
7521 N=2125+PH+PV
7522 K1=PEEK(N+2):K2=PEEK(N-2)
7524 K3=PEEK(N+4):K4=PEEK(N-4)
7525 K5=PEEK(N+6):K6=PEEK(N-6)
7526 IFPV=0THENK2=0:K4=0:K6=0
7527 IFPV=2THENK4=0:K6=0
7528 IFPV=4THENK6=0
7529 IFPV=20THENK3=0:K5=0
7530 IFPV=18THENK5=0
7531 IFK1+K2=92THEN7535
7532 IFK2+K4=92THENIFK6<76THEN2030
7533 IFK1+K3=92THENIFK5<76THEN2030
7534 DG=7:ZB=ZB-1:ZZ=ZZ-1:GOTO2030
7535 IFK4<15THENNW=1
7536 IFK6>76THENNW=1
7537 GOTO2030
7600 REM
7610 IFDG=1THENZB=ZB+1:RETURN
7620 IFDG=2THENZB=ZB-2:RETURN
7622 IFG2+G3=0THEN7624
7623 GOTO7626
7624 IFDG=3THENDG=7:ZB=ZB+1:GOTO 7670
7625 IFDG>7THEN7680
7626 IFG3=0THEN7628
7627 GOTO7630
7628 IFDG=3THENZB=ZB-2:DG=5
7629 IFDG=4THENDG=6:ZB=ZB+4
7630 IFDG=3THENZB=ZB-1:RETURN
7635 IFDG=4THENIFG4=0THENZB=ZB-1:DG=5
7640 IFDG=4THENZB=ZB-1:RETURN
7650 IFDG=5THENZB=ZB+5:RETURN
7660 IFDG=6THENZB=ZB+1:RETURN
7670 IFDG=7THENZZ=ZZ-1:RETURN
7680 IFDG=8THENZZ=ZZ+2:RETURN
7685 IFG9=0THEN7688
7686 GOTO7690
7688 IFDG=9THENDG=11:ZZ=ZZ+2
7689 IFDG=10THENDG=12:ZZ=ZZ-4
7690 IFDG=9THENZZ=ZZ+1:RETURN
7695 IFG0=0THEN7697
7696 GOTO7700
7697 IFDG=10THENZZ=ZZ+1:DG=11
7700 IFDG=10THENZZ=ZZ+1:RETURN
7710 IFDG=11THENZZ=ZZ-5:RETURN
7720 IFDG=12THENZZ=ZZ-1:RETURN
7722 IFDG>12THEN8400
7800 REM
7810 IFDG=2THENG2=1
7815 IFDG=3THENG3=1
7820 IFDG=4THENG4=1
7825 IFDG=9THENG9=1
7830 IFDG=10THENG0=1
7835 RETURN
7845 REM
7900 REM
```

```
7910 IFZB<65THEN2248
7915 IFZB>76THEN2248
7920 IFZZ<1THEN2248
7925 IFZZ>12THEN2248
7930 PV=(ZB-65)*2: PH=(ZZ-1)*64
7940 GOTO2063
7950 REM
7951 IFPH<64THENK6=50:K2=50:K8=50
7952 IFPH>640THENK5=50:K1=50:K7=50
7953 IFPV<2THENK6=50:K3=50:K5=50
7954 IFPV>20THENK8=50:K4=50:K7=50
7955 IFDG=0THEN7971
7956 IFDG=1THENK3=50:GOTO7971
7957 IFDG=2THENK4=50:GOTO7971
7958 IFDG=3THENK4=50:GOTO7971
7959 IFDG=4THENK4=50:GOTO7971
7960 IFDG=5THENK3=50:GOTO7971
7961 IFDG=6THENK3=50:GOTO7971
7962 IFDG=7THENK1=50:GOTO7971
7963 IFDG=8THENK2=50:GOTO7971
7964 IFDG=9THENK2=50:GOTO7971
7965 IFDG=10THENK2=50:GOTO7971
7966 IFDG=11THENK1=50:GOTO7971
7967 IFDG=12THENK1=50:GOTO7971
7971 IFK1>76THEN2247
7972 IFK2>76THEN2247
7973 IFK3>76THEN2247
7974 IFK4>76THEN2247
7975 IFK5>76THEN2247
7976 IFK6>76THEN2247
7977 IFK7>76THEN2247
7978 IFK8>76THEN2247
7979 GOTO2075
8000 CLS
8010 PRINT:PRINT:PRINT:PRINT:PRINT
8020 PRINT"                    ";N1$
8030 PRINT:PRINT
8040 PRINT"      Sie sind der Sieger unseres Kam
pfes!"
8050 PRINT:PRINT
8055 PRINT"      Sie haben";S1;"Schuesse gebrauc
ht
8056 PRINT"      NASCOM hat";S2;"Schuesse gebrau
cht
8057 PRINT:PRINT
8060 PRINT"                 Ich gratuliere"
8065 PRINT
8070 PRINT"      Wollen Sie Ihr Feld zur Kontrol
le?"
8075 INPUT"                                (J/N)";J$
8078 IF J$="J"THEN8200
8080 IF J$<>"N"THEN 8000
8082 CLS:END
8100 CLS
8110 PRINT:PRINT:PRINT:PRINT:PRINT:PRINT
8120 PRINT"  Sie haben diesen Kampf leider ver!
oren!
8130 PRINT:PRINT:PRINT
8140 PRINT"Aber ich freue mich auf einen neuen
Kampf!
8145 SCREEN1,13
8150 GOTO8070
8200 CLS
8210 GOSUB500
8220 DOKE3201,3504:DOKE4100,3200:A=USR(2)
8230 GOSUB 7000
8240 SCREEN1,14
8250 INPUT"Auch mein Feld nochmal? (J/N)";J$
8260 IFJ$="J"THEN 8300
8270 IFJ$<>"N"THEN8200
8280 CLS:END
8300 CLS
8310 GOSUB500
8320 DOKE3201,3360:DOKE4100,3200:A=USR(2)
8330 GOSUB 7300
8340 PRINT:PRINT:END
8400 CLS
8410 PRINT:PRINT:PRINT:PRINT:PRINT
8420 PRINT"        Sie haben Ihre Schiffe nicht"
8430 PRINT"        nach den Spielregeln verlegt!
"
8440 PRINT:PRINT
8450 PRINT"        Sie muessen daher neu anfange
n"
8460 FOR A=1TO 8000:NEXT:GOTO100
9000 RESTORE
9010 FORA=3200TO3359STEP2
9020 READB:DOKEA,B:NEXT
9030 GOTO9300
9100 DATA-8159,4430,2125,3078,4734,4883,1315
9110 DATA-2016,-29423,1544,32268,4882,8979
9120 DATA8197,4600,2253,3078,4734,4883,1315
9130 DATA-2016,3345,1545,32268,4882,8979,8197
9140 DATA4600,2381,3078,4734,4883,1315,-2016
9150 DATA-29423,1545,32268,4882,8979,8197,4600
9160 DATA2509,3078,4734,4883,1315,-2016,3345
9170 DATA1546,32268,4882,8979,8197,4600,2637
9180 DATA3078,4734,4883,1315,-2016,-29423,1546
9190 DATA32268,4882,8979,8197,4600,2765,3078
9200 DATA4734,4883,1315,-2016,3345,1547,32268
9210 DATA4882,8979,8197,-13832
9300 REM
9310 FORB=3648TO3672STEP2
9320 READA:DOKEB,A: NEXT
9330 GOTO1006
9340 DATA12065,22033,14113,32273,15506,288
9350 DATA31433,8960,30464,-13568,6175,-15369
9360 DATA-4608
```

Anmerkung der Redaktion:
Das Spiel läuft nicht mit NASSYS 3, da es zur Tastatureingabe mit SCREENx,y den Input an einer bestimmten Stelle auf dem Bildschirm erwartet, um ihn durch PEEK an den entsprechenden Adressen auszuwerten. NASSYS 3 setzt aber den Cursor beim INPUT Befehl grundsätzlich an den Anfang der nächsten Zeile, sodaß das Spielfeld "gescrolled" wird und die Eingabe in Adressen steht, die der Rechner nicht untersucht.
Mit folgender Änderung wird das Programm auch für NASSYS 3 lauffähig. Es wird zur Eingabe ein Maschinenprogramm benutzt, das die eingetippten Werte direkt in die Adressen lädt, die der Rechner erwartet. Ich hätte das Laden des Maschinenprogramms gerne mit READ und DATA etwas eleganter gestaltet, aber Herr Mombaur hat da eine Routine eingebaut (Zeile 9310), die eine Manipulation mit dem Programm verhindern soll und deren Aufruf ich im Programm nicht finden konnte. So werden die Opcodes einfach mit DOKE geladen. Etwas umständlich, aber es funktioniert. Das Maschinenprogramm läßt sich nicht komfortabler schreiben (z.B.Cursoranzeige etc.), da es bei einer Vergrößerung in den Stackbereich käme.
Ändern Sie also für NASSYS 3 einfach die folgenden Zeilen:

```
130 GOSUB9500
Die Zeilen: 620, 720, 820, 1630, 5510,
            5524, 6050 Jeweils GOSUB9610
1635 REM (kein Schutz vor zu langer Eing.)
5512 IFPEEK(2929)=78THEN5580
5514 IFPEEK(2929)<>74THEN5500
5543 GOSUB9610:KZ$=CHR$(PEEK(2929))
5564 GOSUB9610,J$=CHR$(PEEK(2929))

9500 REM LADEN DES MASCH.PROGR. ZUR
9510 REM EINGABE MIT NASSYS 3
9520 DOKE4053,28961:DOKE4055,-12533
9530 DOKE4057,3582:DOKE4059,-312
9540 DOKE4061,8200:DOKE4063,11013
9550 DOKE4065,8246:DOKE4067,-3304
9560 DOKE4069,9079:DOKE4071,-4328
9570 RETURN

9600 REM AUFRUF DES NASSYS 3 MASCH.PROGR.
9610 SA=DEEK(4100):DOKE4100,4053
9620 ZØ=USR(Ø):DOKE4100,SA
9630 SCREEN41,14:PRINT
9640 RETURN
```

G.B. Jan.82

MK-SYSTEMTECHNIK

Jetzt auch in Solingen!

EUROCOM-2 1670,-

Einplatinencomputer im Doppel-Europaformat. Sehr leistungsfähige Graphik.

- 48k RAM, 4k Betriebssystem
- 6809 CPU (interne 16 Bit Struktur!)
- 128 Zeichen, Groß/Kleinschreibung
- Graphik und Text beliebig mischbar
- Kansas-City-Standard-Interface
- 40 E/A-Leitungen, RS 232 C-Anschluß
- Ausbaubar auf Farbgraphik
- auf 240k RAM erweiterbar

Zubehör für EUROCOM-2

Floppy-Controller Single-Density ohne DMA	1127,-
5"-Laufwerk	847,-
BUS-Karte	84,-
RAM-Karte 32k	779,-
RAM-Karte 96k	1977,-
5A EUROCOM II-Netzteil	384,-
ASCII-Tastatur	279,-
Joystick	110,-

Software f. EUROCOM-2

BASIC	220,-
Assembler/Editor	220,-
DEBUG	179,-
Disassembler	113,-
PASCAL	350,-
FORTH	220,-

wahlweise auf Audiocassette oder Mini-Digitalcassette

MIKOS 1, «Steckdosenfertiges»

Komplettgerät auf Basis d. EUROCOM II inkl. Tastatur und Betriebssystem **2975,-**

mit 1 Mini DCR	3495,-
mit 2 Mini DCR	3975,-
mit 3 Mini DCR	4425,-

apple computer

ITT 2020

Unser PASCAL-System:

Enthält: Wahlweise Apple II oder ITT 2020, 64k RAM (Hardware für PASCAL-Language System), 12" 18 Mhz-Monitor grün, 2 Stück 5.25" Floppy-Disk-Laufwerke mit Controller, **plotfähiger Drucker EPSON MX 82 FT,** UCSD-PASCAL-Software. Komplett mit allen Handbüchern und Verbindungskabeln **9985,-**

Unser Farb-System:

Enthält: ITT 2020 mit PAL-Ausgang, dadurch besonders gutes Farbbild, 48k RAM, 14" SANYO echter Farbmonitor, mit Grünschalter für Computertextdarstellung, auch als vollwertiger 8-Kanal-Farbfernseher zu verwenden, 2 Stück 5.25" Floppy-Disk-Laufwerke mit Controller, plotfähiger Drucker EPSON MX 82 F/T, BASIC-Lehrgang auf Diskette. Komplett mit allen Handbüchern und Verbindungskabeln **8885,-**

Unser Grafik-System:

Enthält: Apple II 48k RAM, 5.25" Floppy-Laufwerk mit Controller, Apple-Grafics-Tablet, plotfähiger Silentype-Drucker, passend zum Grafics-Tablet. Komplett mit allen Handbüchern und Verbindungskabeln **7985,-**

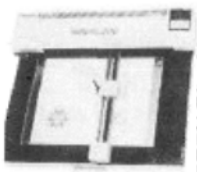
Für Einzelkomponenten oder andere Konfigurationen übersenden wir Ihnen gerne ein individuelles Angebot!

Erweitern Sie Ihren Apple II / ITT 2020:

Timer Modul	295,-	Asynchron Interface	425,-
3 3/4 BCD A/D Wandler	295,-	Synchron Interface	495,-
IEEE Bus Interface	785,-	Parallel Interface	325,-
Arithmetic Processor	995,-	Kalender/Uhr Modul	335,-

Fordern Sie unseren kostenlosen Zubehör- und Software-Katalog sowie unsere CP/M-Sonderliste an!

WATANABE Plotter 3365,-



An alle Microcomputer mit Parallelschnittstelle anschließbar. DIN A/3.

Zubehör:

Interface u. Kabel IEEE 488	449,-
Interface u. Kabel für Apple und ITT 2020	395,-
Interface u. Kabel RS 232 C	849,-

Neu!

Plotbibliothek in UCSD-PASCAL	895,-
Plotsoftware, dialogorientiert	499,-
WATANABE WX 4675	4945,-

Intelligenter 6-Farben-Plotter DIN A/3

Endlich lieferbar!

MX 82 F/T

mit einem Interface n. Wahl*	2595,-
oh. Int.face (8 Bit Parall.-Eing.)	2325,-

Der neue MX 82 F/T besitzt neben allen bewährten Eigenschaften des MX 80 F/T die Fähigkeit, hochauflösende Grafik zu plotten.

MX 80 F/T

o. Interf. (8 Bit Parall.-Eing.)	1625,-
m. einem Interface n. Wahl	1895,-

*Interfacekarten für alle gängigen Rechnersysteme lieferbar: PETICBM, TRS 80, MZ 80 K, IEE 488 (HP), HEATH-Computer, NASCOM, COMPUCORP oder RS 232 C (V24).

ATARI 400 16k RAM, BASIC-ROM, deutsche Handbüch., PAL-Ausg. m. 16 x 8 Farb.	1698,-
ATARI 800 16k RAM, BASIC-ROM, deutsche Handbüch., PAL-Ausg. m. 16 x 8 Farb.	2998,-
16k RAM-Erweiterungsmodul	285,-
5.25" Floppy incl. dt. Handb.	1749,-

VIDEO-GENIE 3003

(neue Ausführung, mit Cassetten-Laufwerk) **1395,-**

VIDEO-GENIE 3008

(mit Kleinschreibmodul, 10er Tastatur u. Cassett.-Anschl.) **1595,-**

Zubehör:

Expansion Interface mit 32k Speichererweiterung	1275,-
5.25" Floppy-Laufwerk mit Gehäuse und Netzteil, 40 Track-Ausführung	995,-
Zweites Laufwerk 40 Track	775,-
Verbindungskabel für 2 Laufwerke	90,-
Kleinschreibmodul für 3003	145,-
RS 232 C Schnittstelle	175,-
S 100 Adapter	295,-

MZ 80 K (48k RAM) **2195,-**

Interface Box	525,-
DIN-Tastatur	375,-

SANYO 12" Monitor

grün, 18 Mhz für augenschonende Dauerarbeit, blendfrei **698,-**

SANYO 12" Monitor

grün, 25 Mhz, angeätzte Bildröhre, für höchste Ansprüche **898,-**

BMC 12" Monitor

grün, 18 Mhz, reflexionsarmer Bildschirm **575,-**

Sonderposten!

Original BASF 5.25" Laufwerk 6106, fabrikneu, originalverpackt, ideal als Zweitlaufwerk **595,-**

16k dyn. RAM 4116, 200ns, orig. MITSUBISHI, allererste Wahl, stückgeprüft!

8 Stück	59,60
16 Stück	115,30

nascom

Lucas Logic

Neu!

Liebe Leser,

Anfang 1981 wurde die englische Firma NASCOM-Microcomputers übernommen. In Wedgnock werden seit Frühjahr in modernsten Produktionsstätten des Konzerns LUCAS Ltd. alle NASCOM-Systeme und Peripheriegeräte in großen Stückzahlen produziert und weiterentwickelt.

MK-Systemtechnik ist der autorisierte Generalvertreter in Deutschland, Österreich und der Schweiz. Nur bei uns bekommen Sie Original NASCOM-Teile aus der neuen Produktion. Nur wir können für Sie Garantieleistungen an den neuen Geräten durchführen!

NASCOM 1

Einplatinencomputer für «stand-alone» und OEM-Anwendungen · Z 80 CPU · 2k statisches RAM · Neues 2k NAS-SYS 3 Betriebssystem · Hochwertiges Keyboard mit 58 Magnettasten · Video-Interface mit 16 x 48 Zeichen · Groß/Kleinschreibung mit Unterlängen · 128 ASCII-Zeichen · BAS-Ausgang · UHF-Ausgang · RS 232 C und 20mA Serien-Schnittstelle · Cassetten-Interface · Mini-DCRs anschließbar · 16 Ein/Ausgabeleitungen (PIO) · 2 Vektorinterrupts · Erweiterbar auf 256k RAM/ROM

Bausatz 855,- Fertiggerät **985,-** Netzteil hierzu (Fertiggerät) **210,-**

Für OEMs auch ohne Keyboard und in Sonderbestückungen lieferbar.

NASCOM II

Dieses Gerät wird häufig als Entwicklungssystem eingesetzt, z. B. um Software für den NASCOM I als OEM-Modul zu erstellen. Es ist voll softwarekompatibel mit dem NASCOM I, hat hardwareseitig jedoch folgende zusätzliche Vorzüge:

· Z 80 CPU 4 Mhz Taktfrequenz · Bis zu 8k RAM (4118) oder EPROM auf der Grundplatine · 8k BASIC in einem 8k x 8 organis. ROM Typ 36000 (MOSTEK) · Voll gepufferter BUS

NASCOM II Bausatz, 8k stat. RAM, BASIC, 2k Monitor	1665,-
NASCOM II Bausatz, 16k dyn. RAM, BASIC, 2k Monitor	1995,-

Fertiggeräte NASCOM II bitte anfragen.

Floppy Disk 1749,-

5" Floppy, Fertiggerät mit DOS, BASIC, Macroassembler, Debug, Texteditor, für NASCOM 1 oder NASCOM 2 — 1 Jahr Software-Pflege kostenlos.

5" Floppy (s.o.) mit Gehäuse und Netzteil für 2 Laufwerke	2144,-
NAS-SYS-Assembler 4k	299,-
NAS-SYS-Disassembler	199,-
NAS-SYS-Debug	75,-

Schach für NASCOM 1/2 128,-

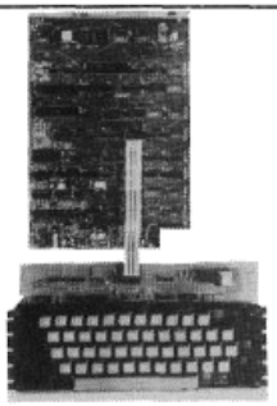
Schachgrafik ROM für NASCOM 2	98,-
Grafik-Zusatzkarte f. NASCOM 1	198,-

NASCOM-JOURNAL

Zeitschrift für den ernsthaften NASCOM-Anwender. Zahlreiche Programmier-Beispiele, Anregungen zur Hardware etc.

Im NASCOM-JOURNAL stellen wir auch laufend neue Produkte vor!

Jahresabonnement 1981	48,-
Jahrgang 1980 komplett (nur solange Vorrat reicht)	39,-



NASCOM 1

Alle Preise sind in DM und schließen die Mehrwertsteuer ein. Versand per Nachnahme oder nach Vorausrechnung. Preisänderung, Irrtum und Zwischenverkauf vorbehalten

MK-Systemtechnik
Pater-Mayer-Straße 6
6728 Germersheim/Rhein
Telefon (0 72 74) 27 56
Telex 0453500 mks d

MK-Systemtechnik
Kriegsstraße 164
7500 Karlsruhe
Telefon (07 21) 2 92 43

MK-Systemtechnik
Pfaffenberg 4
5650 Solingen 1
Telefon (0 21 22) 4 72 67

nascom journal Inhaltsverzeichnis 1981

Grafikerweiterung

von D. Oberle und H.J. Winter

Anwendungsmöglichkeiten:

Durch den Einbau der hier beschriebenen Erweiterungskarte ist es auf einfache Art und Weise möglich, ohne wesentliche Änderungen auf der CPU-Platine die Grafikbefehle des NASCOM-Basic oder auch Grafikausgabe über Assemblerprogramme auf dem NASCOM I auszunutzen.
Die Auflösung beträgt bei Anwendung der Basic SET-Funktion dabei maximal 45 Pkt. in vertikaler und 96 Pkt. in horizontaler Richtung. Es handelt sich lediglich um eine Pseudografik die einen speziellen Charaktergenerator benutzt. Für Spielprogramme und einfache Grafikanwendungen kann die tatsächliche Auflösung durch sinnvolle Auswahl der verwendeten Grafiksymbole erheblich verbessert werden, sodaß kaum ein Unterschied zu echter Grafikdarstellung mit Auflösungen von 256*384 Punkten festgestellt werden kann.
Für den Programmierer eröffnen sich damit eine Menge neuer Ausgabe- und Darstellungsmöglichkeiten.
Außerdem kann durch Umschalten eines Schalters auch inverse Videodarstellung der normalen ASCII-Zeichen ausgewählt werden.
Bei Einsatz des im Handel erhältlichen Schach-Grafikgenerators laufen dann natürlich auch die original Schachprogramme für NASCOM Computer.
Die Karte kommt gänzlich ohne zusätzliche Kabelverbindungen zur CPU-Platine aus und kann nach Umstecken von IC16 u. IC15 auf die Erweiterungskarte direkt in deren freie Sockel eingesteckt werden.

Funktionsbeschreibung:

Wie beim NASCOM II wird auch hier das für ASCII Charakter nicht benötigte Datenbit 7 am Video-Ram Speicher zur Umschaltung auf den Grafikzeichensatz benutzt (Bild 1). Bit 7 wird auf der CPU-Platine an IC18 (18) abgegriffen und über ein noch freies Puffer-Flipflop in IC17 (Pin 18 u. 19) über Pin 10 von IC16 (NC) auf die Erweiterungskarte geführt. Dort muß es über ein Verzögerungsglied (IC7474) der Durchlaufzeit der Datenbits 0-6 angepasst werden, da sonst Doppeldarstellungen oder unsaubere Charakter auf dem Bildschirm erscheinen können.
Abhängig vom Zustand von Bit 7 und von der Schalterstellung des Schalters "S1" wird jeweils eines der beiden Videoschieberegister IC15 oder IC265 freigegeben, sodaß einmal die Datenbits aus dem Charaktergenerator IC16 (ASCII) und einmal aus dem Grafikgenerator IC2716 auf den Videoausgang IC7486 (8) geschaltet werden.
Als Grafikgenerator wird hier ein EPROM 2716 verwendet, sodaß auch der original Charakergenerator des NASCOM II verwendet werden kann. Außerdem kann mit dem nachfolgend beschriebenen Programm jeder selbst einen speziellen Grafikgenerator entwerfen und einfach programmieren.

Funktion und Aufbau des Grafikgenerator-EPROM´s

Die Adressierung eines Grafikcharakters geschieht auf folgende Weise:

Die Adressbits A0-A3 werden durch den Videotaktgenerator generiert und als Grafik-Zeilenadresse interpretiert. Mit diesen 4 Bits können dann also genau 16 Grafikzeilen pro ausgewähltem Grafikcharakter erzeugt werden, wovon jedoch nur 14 durch den Zähler adressiert werden, was multipliziert mit der Zeilenzahl 16 des NASCOM-Bildschirmes eine Pseudoauflösung von (256-32)=224 Grafikzeilen ergibt.
Die Adressbits A4-A10 werden aus den Datenbits des Video-Ram´s direkt über einen Puffer-Speicher (IC17) generiert. Je nach Inhalt des Video-Ram´s wird damit ein ganz bestimmter Charakter ausgewählt und auf dem Bildschirm dargestellt. Für einen Charakter mit 16 Grafikzeilen (nur 14 werden ausgenutzt) werden genau 16-Bytes EPROM-Speicher benötigt, wobei die Matrixbreite eines Charakters genau der Wortbreite des Charaktergenerators (hier 8-Bit) entspricht. Mit den Adressbits A4-A10 können demnach 128 Grafikcharakter ausgewählt werden, was der Speicherkapazität des 2716-EPROM´s entspricht. Aus Bild 2 ist der Aufbau eines Grafikcharakters und dessen Adressierung ersichtlich.

Mit Hilfe des unten abgdruckten BASIC-Programmes kann sich somit jeder seinen eigenen Grafikzeichensatz zusammenstellen und, falls ein 2716 programmiert werden kann, diesen auch gleich als Charaktergenerator einsetzen.

Aufbauanleitung:

Vorweg: Layoutkopie und Bestückungsplan sind gegen Freiumschlag bei der Redaktion erhältlich.
Am besten wird beim Aufbau so vorgegangen wie bereits im letzten NASCOM-Journal (Heft 11) bei der dort von mir vorgestellten Monitorerweiterungskarte.
Die Karte kann nach Fertigstellung direkt in die Sockel von IC15 und IC16 eingesteckt werden wenn unter diese Fassungen auf der Karte DIL-Stecker gelötet werden.
Auf der CPU-Platine des NASCOM I sind noch folgende Änderungen vorzunehmen:

Drahtverbindungen von:

IC18 Pin 18 ---> IC17 Pin 18
IC17 Pin 19 ---> IC16 Pin 10

LITERATURANGABEN:

NASCOM I Construction Articel
NASCOM II Hardware Manual
NASCOM 8K ROM-BASIC Beschreibung

Hilfsprogramme für Erstellung und Test eines eigenen Grafikgenerators

geschrieben
von
H.J. Winter

```
10 REM ***********************************
20 REM ** Ausgabe des Zeichengenerators **
30 REM ** auf den  Drucker              **
40 REM ***********************************
50 REM der Inhalt des Grafikgenerators muss
60 REM ab Adresse 8000h im Speicher liegen
70 WIDTH 80
80 REM Hier Drucker einschalten !!!!!!!!!!
81 REM
82 REM K1 = Anzahl der Bytes pro Charakter
83 REM K2 = Speicher-Anfangsadresse fuer
84 REM      Grafikgenerator (dezimal)
85 REM K5 = Anzahl der auszugebenden Zeichen
87 REM K4 = erstes ausgegebenes Zeichen
88 REM K3 = Anzahl der Zeichen pro Zeile
89 REM
90 K1=16:K2=32768: K3=8: K4=128: K5=128
100 REM
110 REM
120 FOR GZ = INT(K4/K3) TO INT((K5+K4-1)/K3)
130 PRINT:PRINT
140 FOR Z=1 TO K3
141 ZE=Z-1+GZ*K3
142 H1= INT(ZE/16)
143 IF H1 > 9 THEN H1 = H1 + 7
144 HE$ = CHR$(H1+48)
145 H2 = ZE - INT(ZE/16)*16
146 IF H2 > 9 THEN H2 = H2 + 7
147 HE$ = HE$ + CHR$(H2+48) + "H"
150 PRINT TAB((Z-1)*10);HE$;ZE;  :NEXT Z
160 PRINT :PRINT
170 FOR LZ=0 TO 13
180 FOR Z=1 TO K3
185 ZE = Z - 1 + GZ * K3
190 AD = K1 * ZE - K2 + LZ
200 BY = PEEK(AD)
204 ZE=ZE AND 127
205 IF ZE<32 THEN ZE=ZE+64
207 IF ZE=127 THEN ZE=63
 10 DI = 128
220 FOR K=1 TO 8
230 IF(BY AND DI)THENPRINT CHR$(ZE);:GOTO250
240 PRINT" ";
250 DI = DI/2
260 NEXT K
270 IF Z < K3 THEN PRINT"  ";
280 NEXT Z
290 PRINT
300 NEXT LZ
310 NEXT GZ
320 REM Hier Drucker ausschalten !!!!!!!!
330 SCREEN 1,14
```

```
10 REM ***************************
15 REM **  TEST FUER GRAPHIK  ***
16 REM ***************************
20 CLS : CLEAR 300
30 WIDTH 255:K1=16:K2=32768:K3=7:K4=K3+14
31 DOKE 4175, -6649
32 SCREEN K4+2,2:INPUT"Ein oder Aus";B$
34 IF B$="E" THEN 200
36 IF B$ <> "A" THEN  32
```

```
40 GOSUB 1400:CLS
45 SCREEN K4+2,3:PRINT"Zeichen: ";HE$
50 REM ADRESSE = AD
60 AD=DE*K1-K2
70 FOR I=0 TO 13
80 BY=PEEK(AD+I)
90 DI=128:SCREEN 1,I+1:PRINT TAB(K3-1);
100 FOR K=K3 TO K4 STEP 2
105 SCREEN K,I+1:PRINT " "
110 IF(BYANDDI)=DITHENSCREENK,I+1:PRINT "*"
120 DI=DI/2
130 NEXT K
150 NEXT I
160 SCREEN K4+2,1:GOTO 30
170 REM ENDE
200 REM EINGABE
205 GOSUB 1400
210 CLS
211 SCREEN K4+2,3:PRINT"Zeichen: ";HE$
215 AD = DE*K1 - K2
220 FOR Y=1 TO 14
230 FOR X=K3 TO K4 STEP 2
 ,0 SCREEN X,Y:PRINT"*";
250 NEXT X,Y
260 DOKE 4175,-6670
270 FOR Y = 0 TO 13
280 SCREEN 1,Y+1:PRINT Y+1;
290 INPUT ZE$
300 LE = K4
310 IF LEN(ZE$)<K4 THEN LE = LEN (ZE$)
320 BY = 0 : I = 128
330 FOR L = K3-1 TO LE STEP 2
340 IF MID$(ZE$,L,1) = "*" THEN BY = BY OR I
350 I = I/2
360 NEXT L
370 POKE AD+Y,BY
380 NEXT Y
390 GOTO 30
1400 REM Convert HEX to DEZ
1402 DOKE 4175, -6649
.1404 REM SCREEN K4+2,1:PRINT CHR$(27);
1405 SCREEN K4+2,1:INPUT"Sedezimal";HE$
1406 IF LEN(HE$)=0 THEN GOTO 1400
1410 DE=0:K=1
1420 FOR L=LEN(HE$)-1 TO 0 STEP -1
.430 I=ASC(MID$(HE$,K,1))
1440 K=K+1
1450 IF I>47 AND I<58 THEN I=I-48:GOTO 1480
1460 IF I>64 AND I<71 THEN I=I-55:GOTO 1480
1470 GOTO 1400
1480 DE=DE+I*16^L
1490 NEXT:DE=INT(DE+.5):RETURN
```

Programmbeschreibung

Das Programm zur Druckausgabe muß in den Zeilen 80 und 320 noch mit Routinen zum Ein- bzw. Ausschalten des verwendeten Druckers versehen werden. Hier können z.B. mit DOKE Befehlen die NASCOM "X20" Routinen aktiviert oder deaktiviert werden.
Vor dem Programmstart muß der Inhalt des Grafikgenerators in den Speicher ab Adresse K2 kopiert werden. K2 muß außerhalb der beim Basicstart angegebenen Speichergröße liegen.

Nach Starten des Programms gibt der Drucker die Grafikzeichen in folgender Form aus:

```
Adr.  Hex Dez
---------------
z.B.  ADH 173

      ********
      ******
      *****
      ***
      **
      *
      *
      *
      **
      ***
      ****
      *****
      ******
      ********
```

die Anzahl der Zeichen pro Papierbreite kann im Programm mit K3 vorgegeben werden.

Das Programm zum Erstellen, Testen oder Ändern eines beliebigen Grafikzeichensatzes fragt nach dem Start ab, ob ein Grafikzeichen aus dem Speicher (ab Adresse K2) aus- oder eingegeben werden soll, was mit "A" oder "E" zu beantworten ist. Danach wird nach der hexadezimalen Zeichenadresse gefragt, die entsprechend von 80h-FFh angegeben werden kann.
Bei Ausgabe wird das entsprechende Grafikzeichen in einer 8*14 Matrix links auf dem Bildschirm ausgegeben und dann eine neue Eingabe erwartet.
Bei Eingabe wird eine Matrix aus 8*14 "*" links auf dem Bildschirm dargestellt und zu

jeder der 14 Zeilen eine Eingabe erwartet. Wird das "*" an einer Stelle stehen gelassen so bedeutet dies Bit="1", wird es dagegen gelöscht so bedeutet dies BIT="0".
Nach den 14 Eingaben ist das Muster aus dem Bildschirm direkt in den Speicher ab Adresse (K2+Zeichenadresse) übertragen und kann von dort direkt in ein EPROM geschrieben werden.
Auch hier muß die Adresse K2 außerhalb des von Basic benutzten Speicherbereiches liegen.

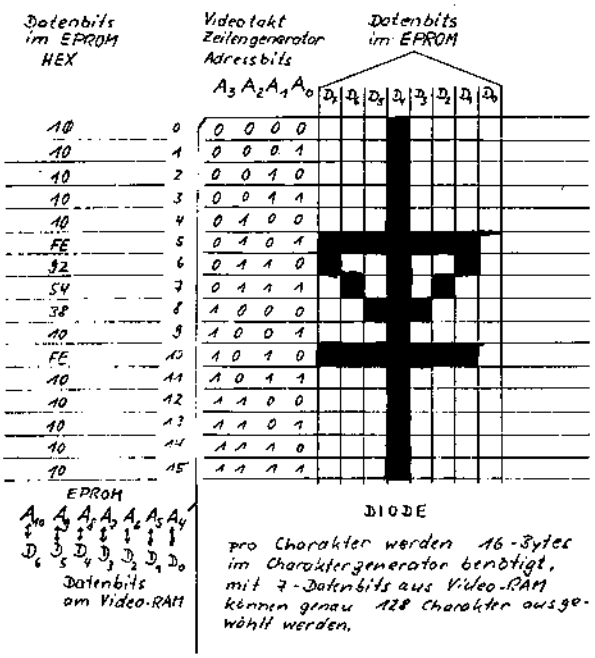


Bild 2

NASCOM I GRAFIKERWEITERUNG

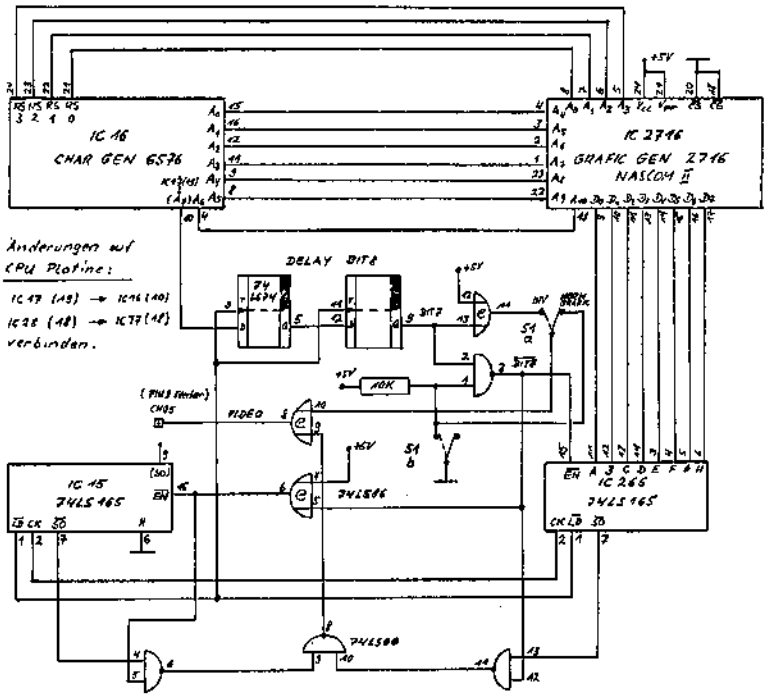


BILD 1

(Es wäre noch darauf hinzuweisen, daß der NASCOM 2 nur 14 der 16 Punktreihen abbildet. Ein Maschinenprogramm zum Ausdrucken des Zeichenvorrats finden Sie in Heft 10-81 des NASCOM Journals. Red.)

Die Mitarbeiter dieser Ausgabe waren:
Clemens Ballarin, [redacted], [redacted] Überlingen, [redacted] - Günter Böhm, siehe Impressum - Bernd im Brahm, [redacted], [redacted] Wickede/Ruhr - Eberhard Horch, [redacted], [redacted] Hannover [redacted] - Günter Kreidl, siehe Impressum - Johannes Christian Lotter, [redacted], [redacted] Darmstadt, [redacted] - Wolfgang Mayer-Gürr, siehe Impressum - Klaus Mombaur, [redacted], [redacted] Nürnberg [redacted] - Dieter Oberle, [redacted], [redacted] Vollmersweiler - Christian Peter, [redacted], A[redacted] Wien - Hans-Martin Pohl, [redacted], [redacted] Stuttgart [redacted], [redacted] - Peter Urban, [redacted], [redacted] Niefern [redacted], [redacted] - H.J.Winter, [redacted], [redacted] Landau - und virtuos am Typenraddrucker: Gabi Böhm -
Die Autoren tragen die Verantwortung für ihre Beiträge selbst.

Format f. Nassys

von Günter Böhm

Ein Leser fragte an, warum denn trotz meines Lobes auf NASSYS als komfortables Betriebssystem das Formatierprogramm in Heft 8-81 für T4 geschrieben sei. Nun, der Grund dafür ist einfach: ich besaß damals noch kein NASSYS. Inzwischen habe ich es mir zugelegt (bin vollkommen damit zufrieden). Da nun schon einige Leser nach einer "FORMAT" Version für NASSYS fragten, habe ich mich nun endlich darangesetzt.

Daß sich die Anwendung des vergleichsweise einfachen Programmes trotz Bestehen sehr komfortabler Textsysteme lohnt, wurde mir bestätigt, als ich versuchsweise einige Texte mit NASPEN editierte. Eine Unmenge von Befehlen, die für meine Zwecke (den blockweisen Ausdruck relativ kurzer Texte) nicht benötigt werden, das dauernde Eintipoen des "c", wenn mehrere Buchstaben geändert werden sollten und einiges an umständlicher Bedienung mehr. Nicht zuletzt störte mich die Festlegung auf einen bestimmten Speicherbereich. Diese Nachteile finde ich in FORMAT nicht; zudem können Texte dafür von jedem erstellt werden, auch wenn er kein Textsystem besitzt. FORMAT Texte auf Cassette können zwischen allen NASCOM Benutzern ausgetauscht werden.

Die Version für NASSYS ist etwas komfortabler als das Original und ein kleines bißchen länger geworden. Deshalb kommt das Programm in gefährliche Nähe des Stackbereiches und kann z.T. überschrieben werden, wenn man es im alten Speicherbereich läßt. So habe ich es auf #1000 gelegt. (Allerdings habe ich es auch in einer Version ab C000 vorliegen. Wer Interesse an einer EPROM Version hat, kann ja mal anfragen).

Der Vorteil gegenüber der Originalversion liegt hauptsächlich darin, daß nach jedem Arbeitsgang die Endadresse des Textspeichers angezeigt wird.So kann man durch Eingabe dieser Adresse beim Einlesen von Cassette oder Eintippen neuen Text anhängen. Auch die Bedientasten wurden vereinfacht(auch mit Hilfe der Cursortasten). Hier nun die Bedienungsanleitung:

Gestartet wird mit E1335

Das Programm meldet sich mit FORMAT

Folgende Arbeitsgänge können nun durchgeführt werden:

Eingabe von Text:

IXXXX, wobei das Argument den Beginn des Textspeichers festlegt. Beendet wird diese Routine durch ESC (SHIFT NL) oder durch das Schnecken A. Die Routine meldet dann "NEXT MEM AT XXXX", wobei das Argument die Adresse meint, ab der man neuen Text eingeben kann.

Lesen von Cassetten:

RXXXX Argument ist Speicherbeginn (oder z B. die oben ausgegebene Adresse zum Anhängen des Cassettentextes. Während des Lesens wird der Text auf dem Schirm ausgegeben. Das Lesen wird jederzeit gestopot durch Anhalten des Recorders und Eingabe von @ (das ja auf dem Band als Endmarkierung benutzt wird. Nun erfolgt ebenfalls die NEXT MEM Anzeige.

WXXXX Beschreibt eine Cassette mit dem Text ab XXXX. (Es muß nicht der Speicheranfang sein). Nach dem Schreiben wieder die obige Anzeige.

Ändern des Textes (Modify):

MXXXX Die Textzeile ab XXXX wird auf dem Schirm ausgegeben. Ein ↑ dient als Cursor. Durch Eintippen beliebiger Tasten wird das entsprechende Zeichen geändert, und der Cursor springt zum nächsten. So können leicht fortlaufende Wörter geändert werden. Durch →wird der Cursor nach rechts, mit ← nach links bewegt. ↓ zeigt die nächste Textzeile, ↑ rückt eine Zeile weiter. Das Ende des Textes wird durch @ markiert. Beendet wird die Routine wiederum durch ESC, sie zeigt dann auf dem Bildschirm die Adresse des Cursors im Textspeicher an. Falls ein Wort getrennt werden muß, zeigt der ↑ cursor auf das Zeilenende. Mit BACKSPACE wird nun der abzutrennende Teil nach rechts geschoben. NL trennt ohne, "-" mit Trennungszeichen. Will man die ganze Zeile nach dem Hinausschieben von z.B. einem ganzen Wort völlig ändern, so gibt man ESC ein und kann mit Hilfe der Cursortasten wie gewohnt manipulieren. NL gibt dann die geänderte Zeile an den Drukker. Manchmal muß mehrmals NL gedrückt werden, wenn mehr als jeweils ein Space eingefügt werden muß.

Nachzutragen wäre noch die Funktion zweier Tasten: SHIFT → schiebt den Text ab Cursor nach rechts und schafft Platz zum Einfügen, mit SHIFT ← wird der Text rechts neben dem Cursor nach links geschoben und dadurch gelöscht. Besonders angenehm ist das Edititieren natürlich mit der Repeatfunktion von NASSYS3.

Formatieren:

FZZBB XXXX Dabei bedeuten ZZ die Zeichenanzahl pro Zeile, BB die Zeilenanzahl pro Block und XXXX wieder die Startadresse im Textspeicher. Die Argumente müssen als HEX-Zahlen eingegeben werden.(Leider!) Am Ende eines Blocks meldet sich NEW PARAMETER Y/N. Durch Eingabe von N startet der Ausdruck des nächsten Blocks mit gleicher Zeichen/Zeilenanzahl. Durch Y kehrt die Routine zu FORMAT zurück mit Angabe der Cursorposition. Von hier ab kann nun mit anderen Werten formatiert werden. (Für uns interessant zum Einschieben von Bildern oder Tabellen).

Falls Interesse besteht könnten Sie das Programm im "Rundlaufverfahren" auf Cassette erhalten. Ebenfalls könnte der Source Code veröffentlicht werden. (Fotokopieren müßten Sie ihn selbst - aber bitte Rückgabegarantie des Originals!)

Zum Verständnis des Programms verweise ich nochmals auf die Dokumentation in Heft 8-81

Am grundlegenden Prinzip hat sich nichts geändert.

Wer eine eigene Druckerroutine einbauen will: der Aufruf des Druckprogrammes muß in #1368 eingetragen werden. (Da steht jetzt SRLX drin).

Und auf geht's mit F2000 ..... sehen Sie dem Text an, wie das flutscht ?

```
T1000 136B
1000 2A 0C 0C E5 DF 7B E1 FE 70
1008 08 20 04 2B F7 18 F4 FE 70
1010 1B 20 04 3E 40 18 06 FE F9
1018 0A 20 02 3E 20 77 23 F7 43
1020 FE 40 20 DF EF 0D 4E 45 FC
1028 58 54 20 4D 45 4D 20 41 44
1030 54 20 00 C5 DF 66 C1 DF 5E
1038 6A C3 38 13 EF 0C 00 2A E5
1040 0E 0C ED 4B 0C 0C 11 4A 15
1048 08 C5 48 06 00 7E FE 20 0F
1050 20 03 23 18 F8 1A FE 00 CE
1058 20 03 11 8A 08 7E FE 0D B7
1060 20 06 C1 12 23 C3 F3 10 52
1068 FE 40 28 F6 ED A0 78 B1 8A
1070 20 E3 C1 1B 1A CD 7F 10 D5
1078 FE 00 C2 F3 10 18 1A FE 7B
1080 21 C8 FE 2D C8 FE 2C C8 5E
1088 FE 2E C8 FE 3A C8 FE 3B C5
1090 C8 FE 3D C8 FE 3F C8 AF 1F
1098 C9 1A FE 20 C2 6A 11 C5 AB
10A0 06 00 04 1B 1A FE 20 28 35
10A8 F9 13 E5 21 4A 08 D5 23 14
10B0 7E E5 A7 ED 52 E1 30 08 22
10B8 28 06 FE 20 28 07 18 EF 4A
10C0 D1 E1 C3 AF 11 1A F5 13 27
10C8 1A FE 00 28 FA F1 12 1B 30
10D0 1B 1A FE 00 20 0C 13 E5 37
10D8 D5 E1 1B 1A A7 28 FB 77 14
10E0 1B E1 A7 E5 ED 52 E1 20 B8
10E8 DC 1A 13 12 23 D1 13 10 2A
10F0 BD E1 C1 E5 C5 21 49 08 7B
10F8 23 7E FE 40 20 11 C1 E1 BA
1100 EF 0C 54 45 58 54 20 45 B6
1108 4E 44 45 00 C3 24 10 FE E5
1110 0D 28 09 FE 00 28 E1 CD 33
1118 68 13 10 DC 3E 0D CD 68 10
1120 13 C1 0D 79 A7 E1 28 06 41
1128 EF 0C 00 C3 46 10 EF 0C 48
1130 42 4C 4F 43 4B 20 45 4E 5F
1138 44 45 0D 4E 45 57 20 50 39
1140 41 52 41 4D 45 54 45 52 A2
1148 20 59 2F 4E 20 3F 00 E5 93
1150 D5 DF 7B FE 59 28 06 FE 13
1158 4E 28 07 18 F4 D1 E1 C3 67
1160 24 10 D1 E1 EF 0C 00 C3 15
1168 42 10 7E FE 20 28 84 C5 D8
1170 06 00 04 1A FE 00 28 37 02
1178 1B FE 20 28 0A 13 CD 7F 53
1180 10 FE 00 1B 28 EC 05 E5 B8
1188 21 4A 08 AF F5 23 E5 A7 5F
1190 ED 52 E1 28 09 7E FE 20 8E
1198 20 F3 F1 3C 18 EE F1 E1 C1
11A0 B8 38 0C C5 28 13 3E 20 0E
11A8 12 10 F9 C1 C3 AA 10 C1 D3
11B0 C5 11 49 08 13 1A A7 28 E4
11B8 FB 10 F9 D5 E5 EB 11 40 C3
11C0 00 19 36 5E E1 D1 D5 C1 C6
11C8 D5 E5 7E FE 20 28 0B 13 75
11D0 1A FE 00 28 FA 7E 12 23 CE
11D8 18 F0 E1 D1 E5 D5 DF 7B B7
11E0 D1 E1 FE 08 20 07 3E 20 2E
11E8 02 0B 28 18 DB FE 20 20 6F
11F0 04 03 02 18 04 FE 0D 20 51
11F8 04 13 C3 72 10 FE 1B 20 9E
1200 DB ED 43 29 0C D5 DF 63 69
1208 D1 13 18 EE 2A 0C 0C 11 57
1210 4A 0B 01 30 00 EF 0C 00 A3
1218 ED B0 3E 5E 32 8A 0B E5 0F
1220 D5 DF 7B D1 E1 FE 14 28 4D
1228 E6 FE 13 20 07 06 60 2B E9
1230 10 FD 18 DB FE 3E 20 44 E2
1238 E5 CD 67 12 E1 E5 2B 10 76
1240 FD 01 01 00 03 23 7E FE F3
1248 40 20 F9 E5 D1 13 ED B8 21
1250 23 36 20 CD 67 12 48 06 6F
1258 00 11 7A 0B 21 79 0B ED 92
1260 B8 23 36 20 E1 18 B8 11 65
1268 BA 0B 06 00 1B 04 1A FE 7C
1270 5E 20 F9 21 40 00 A7 EB EC
1278 ED 52 EB C9 FE 11 20 18 C4
1280 11 BA 0B 1B 1A FE 5E 20 19
1288 FA 1B 1A FE 00 28 90 3E BD
1290 5E 12 13 3E 20 12 18 87 34
1298 FE 12 20 1A 11 89 0B 13 AC
12A0 1A FE 5E 20 FA 13 1A FE 6D
12A8 00 CA 1F 12 3E 5E 12 1B 7E
12B0 3E 20 12 C3 1F 12 FE 3C 60
12B8 20 28 E5 CD 67 12 E1 E5 03
12C0 2B 10 FD E5 D1 01 00 00 C1
12C8 23 03 3E 40 BE 20 F9 D5 2A
12D0 E1 E5 23 ED B0 CD 67 12 AE
12D8 48 06 00 E1 ED B0 E1 C3 5A
12E0 1F 12 E5 F5 E5 CD 67 12 28
12E8 E1 2B 10 FD F1 FE 1B 20 3D
12F0 14 D1 EF 0D 0D 43 55 52 DA
12F8 53 4F 52 20 41 54 20 00 D3
1300 DF 66 C3 38 13 77 12 E1 D0
1308 C3 9D 12 2A 0C 0C DF 5F 0D
1310 00 00 00 7E 23 F5 F7 DF 8F
1318 6F F1 FE 40 20 F5 DF 5F 1C
1320 C3 24 10 2A 0C 0C DF 5F AA
1328 23 CF F7 77 FE 40 20 F8 F1
1330 DF 5F C3 24 10 EF 0C 00 73
1338 EF 46 4F 52 4D 41 54 0D 10
1340 00 DF 63 1A F5 13 DF 79 0F
1348 F1 FE 49 CA 00 10 FE 46 B1
1350 CA 3C 10 FE 4D CA 0C 12 AC
1358 FE 57 CA 0B 13 FE 52 CA C2
1360 23 13 FE 4E 20 D2 DF 5B 21
1368 DF 6F C9 00 00 00 00 00 92
```

Seite(n) für Einsteiger

BCD-Arithmetik für Einsteiger

Bei der Programmierung arithmetischer Prozeduren sind zwei verschiedene Zahlenrepräsentationen üblich, einmal im Binärformat, üblicherweise im Zweierkomplement (Damit rechnen z.B. fast alle Basic-Interpreter), zum anderen als Binär-Codierte-Dezimalzahlen. Um diese Darstellung zu erläutern, sei zunächst auf die hexadezimale Darstellung zurückgegriffen, die Günter Böhm im letzten NASCOM-Journal erklärt hat. Dabei wird jedes Byte in zwei Hälften ("Nibbles") zerlegt, die durch die Hex-Ziffern Ø - F dargestellt werden. Schränkt man diesen Wertebereich nun auf die Werte Ø -9 ein, so kann man mit einem Byte die Dezimalzahlen ØØ - 99 darstellen, also mit jedem Nibble eine Dezimalstelle. Es handelt sich um eine Abbildung des Dezimalsystems auf das binäre System des Prozessors, wobei nur eine Untermenge der möglichen binären Werte zugelassen ist. Wie kann der Prozessor nun mit solchen Werten rechnen?

Der DAA-Befehl

Der Z8Ø verfügt wie die meisten Prozessoren über einen besonderen Befehl für das Rechnen mit BCD-Werten: Decimal-Adjust-Accumulator. Um die Arbeitsweise dieses Befehls zu verstehen, nehmen wir einmal an, der Akku enthalte den Wert Ø9, und wir würden Ø1 addieren. Der Akku enthält dann den Wert ØA, also einen Wert, der im BCD-System nicht zugelassen ist. Der Befehl DAA prüft den Akkumulator und addiert (in diesem Fall!) den Wert Ø6 hinzu. Das führt zu einem Uebertrag vom niederwertigen Nibble in den höherwertigen. Im Akku steht dann der Wert 1Ø, also ein im BCD-System zugelassener Wert. Steht im Akku der Wert 99 und wir addieren wiederum Ø1, so steht zunächst 9A. Lautet der nächste Befehl DAA, so wird 66 hinzuaddiert. Das ergibt den Wert ØØ und gleichzeitig wird das Carry-Flag gesetzt. Damit kann dann ein Uebertrag in die nächste Stelle (in einem weiteren Byte) vorgenommen werden. Der DAA-Befehl arbeitet nach folgenden Befehlen:
ADD, ADC, INC, SUB, SBC, DEC, NEG
Einschränkung: Er wirkt nur auf den Akku und die Operanden müssen BCD-Format haben.

RRD und RLD

Wenn man im BCD-Format Rechenprogramme erstellen will, muß man des öfteren auf einzelne Nibble, also Halbbytes zugreifen. Der Z8Ø verfügt deshalb über zwei spezielle Befehle, die diesen Zugriff auf einfache Weise ermöglichen: Rotate Right Decimal und Rotate Left Decimal. Ich will die Arbeitsweise am Beispiel RLD erläutern. Das Register HL muß auf eine Speicherstelle zeigen. Mit RLD wird dann das niederwertige Nibble von A an die Stelle des niederwertigen Nibble von (HL) gesetzt, der dort vorher stehende Wert wird in das höherwertige Nibble von (HL) geschoben, und der Wert, der zuvor in diesem höherwertigen Nibble stand, steht nun im niederwertigen Halbbyte des Akkus. (Graphische Darstellung in den Z8Ø-Unterlagen zu Hilfe nehmen!) Mit RLD und RRD kann man also einzelne Stellen des "gepackten" BCD-Formats (zwei Stellen pro Byte) in den Akku holen.

Anwendungen

Das BCD-Format kommt hauptsächlich dann zur Anwendung, wenn man mit beliebiger, frei programmierbarer Stellenzahl arbeiten will, wenn die benötigte Stellenzahl oder Genauigkeit größer ist, als z.B. in den arithmetischen Routinen der meisten Basic-Interpreter (Man versuche einmal, dem Finanzamt zu verklickern, daß der Heimcomputer eben nur auf 8 Stellen genau rechnen kann!), oder wenn bei der Binär-Dezimal-Umwandlung keine Rundungsfehler auftreten dürfen. Ich persönlich finde das Programmieren von arithmetischen Routinen im BCD-Format durchsichtiger u nd leichter zu "entwanzen". Die BCD-Zahlen brauchen jedoch etwa 2,5 mal soviel Speicherplatz wie reine Binärzahlen und die Rechengeschwindigkeit ist um den gleichen Faktor langsamer.
Wer die Arbeitsweise der in diesem Heft vorgestellten BCD-Routinen näher kennenlernen will, der sollte einmal das zugehörige Testprogramm mit Breakpoints in den Schleifen laufen lassen, und sich bei jedem Breakpoint mit dem Tabulate-Kommando die Rechenregister anzeigen lassen.

Günter Kreidl

```
                    0010 ;Einfache arithmetische Routinen
                    0020 ;im "gepackten" BCD-Format
                    0030 ;
                    0040 ;Die Zahlen befinden sich in
                    0050 ;Rechenregistern im Speicher.
                    0060 ;Es wird jeweils im Hauptprogramm
                    0070 ;die Stellenzahl und die Anzahl
                    0080 ;der Nachkomma-Stellen bestimmt.
                    0090 ;Ueberlaufkontrolle und negative
                    0100 ;Zahlen sind in diesen einfachen
                    0110 ;Unterprogrammen noch nicht ent-
                    0120 ;halten.
                    0130 ;Mit "rechts" sind jeweils die
                    0140 ;beiden niederwertigsten Stellen
                    0150 ;einer Zahl gemeint, die immer
                    0160 ;an der höchsten Adresse des je-
                    0170 ;weiligen Rechenregisters (R1, R2,
                    0180 ;R3) stehen.
                    0190 ;
                    0200 ;BCD-Addition
                    0210 ;R1+R2 nach R1
                    0220 ;IX: R2 rechts
                    0230 ;IY: R1 rechts
                    0240 ; B: Anzahl Bytes des Registers
0D00                0250         ORG  #D00
0D00 AF             0260 ADDBCD XOR  A
0D01 FD7E00         0270 LOOPA  LD   A,(IY)
0D04 DD8E00         0280        ADC  A,(IX)
0D07 27             0290        DAA
0D08 FD7700         0300        LD   (IY),A
0D0B DD2B           0310        DEC  IX
0D0D FD2B           0320        DEC  IY
0D0F 10F0           0330        DJNZ LOOPA
0D11 C9             0340        RET
                    0350 ;Das UPRO ist verschieblich.
                    0360 ;
                    0370 ;BCD-Subtraktion
                    0380 ;R1-R2 nach R1
                    0390 ;IX, IY, B wie oben
0D12 AF             0400 SUBBCD XOR  A
0D13 FD7E00         0410 LOOPS  LD   A,(IY)
0D16 DD9E00         0420        SBC  A,(IX)
0D19 27             0430        DAA
0D1A FD7700         0440        LD   (IY),A
0D1D DD2B           0450        DEC  IX
0D1F FD2B           0460        DEC  IY
0D21 10F0           0470        DJNZ LOOPS
0D23 C9             0480        RET
                    0490 ;Das UPRO ist verschieblich.
                    0500 ;
                    0510 ;BCD-Linksschieben
                    0520 ;Die Zahl im Register R wird um
                    0530 ;eine Stelle "nach links" versch-
                    0540 ;ben, genau wie man Zahlen auf dem
                    0550 ;Papier verschoben untereinander
                    0560 ;schreiben kann.
                    0570 ;HL: Register rechts
                    0580 ; B: Registerlänge (Bytes)
                    0590 ;
0D24 AF             0600 BCDLI  XOR  A
0D25 ED6F           0610 DEZROT RLD
0D27 2B             0620        DEC  HL
0D28 10FB           0630        DJNZ DEZROT
0D2A C9             0640        RET
                    0650 ;auch verschieblich
                    0660 ;
                    0670 ;N-mal BCD-Rechtsschieben
                    0680 ;genau wie oben, nur andersrum!
                    0690 ;HL: Register links
                    0700 ; B: Registerlänge
                    0710 ; C: N
                    0720 ;
0D2B C5             0730 BCDNR  PUSH BC
0D2C E5             0740        PUSH HL
0D2D AF             0750        XOR  A
0D2E ED67           0760 ROTDEZ RRD
0D30 23             0770        INC  HL
0D31 10FB           0780        DJNZ ROTDEZ
0D33 E1             0790        POP  HL
0D34 C1             0800        POP  BC
0D35 0D             0810        DEC  C
0D36 20F3           0820        JR   NZ,BCDNR
0D38 C9             0830        RET
                    0840 ;verschiebl.
                    0850 ;
                    0860 ;BCD-Multiplikation
                    0870 ;R2 * R3 nach R1
                    0880 ;Das ist nun schon erheblich kom-
                    0890 ;plizierter als die bisher gezeigten
                    0900 ;Routinen. Man könnte das Programm
                    0910 ;etwas einfacher schreiben, wenn man
                    0920 ;mit einem festen Format arbeiten
                    0930 ;würde. In der hier gezeigten Form
                    0940 ;sind die Gesamtzahl der Stellen und
                    0950 ;die Zahl der Nachkomma-Stellen frei
                    0960 ;wählbar, mit einer Einschränkung:
                    0970 ;Die Stellenzahl muß durch 4, die
                    0980 ;Zahl der Stellen nach dem Komma
                    0990 ;durch 2 teilbar sein.
                    1000 ;Da bei der Multiplikation das Er-
                    1010 ;gebnis maximal doppelt so viele
                    1020 ;Stellen haben kann wie die maximale
                    1030 ;Stellenzahl der beiden Operanden,
                    1040 ;müssen die Rechenregister doppelt
                    1050 ;"so breit" sein wie die zulässigen
                    1060 ;Operanden. Einfache Regel:
                    1070 ;Die Rechenregister umfassen soviel
                    1080 ;Bytes, wie die Operanden maximal
                    1090 ;Stellen haben. Es empfiehlt sich
                    1100 ;daher, für die Multiplikation
```

```
                1110 ;eigene Rechenregister anzulegen,
                1120 ;in die man die Zahlen hineinkopiert.
                1130 ;Auch die Position der Zahlen in den
                1140 ;Registern ist wichtig: "links" müssen
                1150 ;soviele Bytes freibleiben, wie
                1160 ;man auch für die Vorkomma-Stellen
                1170 ;bereitstellt, rechts ebensoviel wie
                1180 ;für die Nachkomma-Stellen. Das wird
                1190 ;anhand eines Testprogramms erläutert.
                1200 ;
                1210 ;IX: R2 rechts, IY: R1 rechts
                1220 ;HL: letzte Nachkommastelle in R3
                1230 ; B: Reg.-Länge  C: NK-Stellenzahl
OD39 E5         1240 BCDMUL PUSH HL
OD3A DDE5       1250        PUSH IX
OD3C E1         1260        POP  HL
OD3D 1600       1270        LD   D,0
OD3F 58         1280        LD   E,B
OD40 1D         1290        DEC  E
OD41 AF         1300        XOR  A
OD42 ED52       1310        SBC  HL,DE
OD44 B1         1320        OR   C
OD45 2805       1330        JR   Z,INT
OD47 C5         1340        PUSH BC
OD48 CD2B0D     1350        CALL BCDNR
OD4B C1         1360        POP  BC
OD4C E1         1370 INT    POP  HL
OD4D 50         1380        LD   D,B
OD4E CB3A       1390        SRL  D
OD50 CB3A       1400        SRL  D
OD52 79         1410        LD   A,C
OD53 CB3F       1420        SRL  A
OD55 82         1430        ADD  A,D
OD56 57         1440        LD   D,A
OD57 CBC3       1450 SET    SET  0,E
OD59 ED67       1460 ROT    RRD
OD5B 4F         1470        LD   C,A
OD5C F5         1480        PUSH AF
OD5D B7         1490        OR   A
OD5E 2810       1500        JR   Z,SHIFT
OD60 DDE5       1510 MULT   PUSH IX
OD62 FDE5       1520        PUSH IY
OD64 C5         1530        PUSH BC
OD65 CD0000     1540        CALL ADDBCD
OD68 C1         1550        POP  BC
OD69 FDE1       1560        POP  IY
OD6B DDE1       1570        POP  IX
OD6D 0D         1580        DEC  C
OD6E 20F0       1590        JR   NZ,MULT
OD70 E5         1600 SHIFT  PUSH HL
OD71 DDE5       1610        PUSH IX
OD73 E1         1620        POP  HL
OD74 C5         1630        PUSH BC
OD75 CD240D     1640        CALL BCDLI
OD78 C1         1650        POP  BC
OD79 E1         1660        POP  HL
OD7A F1         1670        POP  AF
OD7B CB43       1680        BIT  0,E
OD7D 2007       1690        JR   NZ,RESET
OD7F ED67       1700        RRD
OD81 2B         1710        DEC  HL
OD82 15         1720        DEC  D
OD83 20D2       1730        JR   NZ,SET
OD85 C9         1740        RET
OD86 CB83       1750 RESET  RES  0,E
OD88 18CF       1760        JR   ROT
                1770 ;
                2000 ;BCD-Test
                2010 ;Die Routine prüft, ob das Zei-
                2020 ;chen im Akku eine Zahl zwischen
                2030 ;0 und 9 (ASCII) ist und setzt
                2040 ;das Carry-Flag, falls es sich
                2050 ;um irgendein anderes Zeichen
                2060 ;handelt.
                2070 ;
OD8A FE3A       2080 BCDTST CP   #3A
OD8C 3003       2090        JR   NC,SET
OD8E FE30       2100        CP   #30
OD90 D0         2110        RET  NC
OD91 37         2120 SET    SCF
OD92 C9         2130        RET
                2140 ;
                2150 ;BCD-Eingabe
                2160 ;Diese Routine liest eine BCD-
                2170 ;Zahl stellengerecht vom Bild-
                2180 ;schirm in ein Register ein.
                2190 ;Es sind sowohl Dezimalpunkt als
                2200 ;auch Dezimalkomma zugelassen.
                2210 ;Der Zahl muß ein Leerzeichen
                2220 ;folgen. Falsche Zeichen werden
                2230 ;erkannt und mit "Error" angezeigt.
                2240 ;DE zeigt dann auf das falsche
                2250 ;Zeichen. Eine Ueberlaufkontrolle
                2260 ;ist hingegen nicht vorhanden.
                2270 ;Das Register muß "links" soviel
                2280 ;Bytes Reservespeicher bieten,
                2290 ;wie "rechts" Bytes für Nachkomma-
                2300 ;Stellen vorgesehen sind.
                2310 ;Die Rechenregister müssen vorher
                2320 ;"gelöscht", d.h. auf 0 gesetzt sein.
                2330 ;Enthält die Eingabe ein unzulässiges
                2340 ;Zeichen, dann wird "Error" ausgegeben
                2350 ;und das Unterprogramm kehrt mit ge-
                2360 ;setztem Carry-Flag zurück. DE weist
                2370 ;dann auf das falsche Zeichen.
                2380 ;HL: Register links
                2390 ;DE: erste Stelle der Zahl auf dem
                2400 ;    Bildschirm
                2410 ; B: Registerlänge (Bytes)
                2420 ; C: max. Anzahl von NK-Stellen
                2430 ;
```

```
0D93 C5        2440 BCDIN  PUSH BC
0D94 E5        2450        PUSH HL
0D95 CB39      2460        SRL  C
0D97 78        2470        LD   A,B
0D98 91        2480        SUB  C
0D99 3D        2490        DEC  A
0D9A 4F        2500        LD   C,A
0D9B 0600      2510        LD   B,0
0D9D 09        2520        ADD  HL,BC
0D9E 0E00      2530        LD   C,0
0DA0 1A        2540 COUNT1 LD   A,(DE)
0DA1 FE2C      2550        CP   ","
0DA3 2811      2560        JR   Z,COUNT2
0DA5 FE2E      2570        CP   "."
0DA7 280D      2580        JR   Z,COUNT2
0DA9 FE20      2590        CP   #20
0DAB 2818      2600        JR   Z,TRANS
0DAD CD8A0D    2610        CALL BCDTST
0DB0 3849      2620        JR   C,ERR
0DB2 04        2630        INC  B
0DB3 13        2640        INC  DE
0DB4 18EA      2650        JR   COUNT1
0DB6 13        2660 COUNT2 INC  DE
0DB7 1A        2670        LD   A,(DE)
0DB8 FE20      2680        CP   #20
0DBA 2809      2690        JR   Z,TRANS
0DBC CD8A0D    2700        CALL BCDTST
0DBF 383A      2710        JR   C,ERR
0DC1 04        2720        INC  B
0DC2 0C        2730        INC  C
0DC3 18F1      2740        JR   COUNT2
0DC5 CB40      2750 TRANS  BIT  0,B
0DC7 2805      2760        JR   Z,TRANS1
0DC9 CB38      2770        SRL  B
0DCB 04        2780        INC  B
0DCC 1802      2790        JR   TRANS2
0DCE CB38      2800 TRANS1 SRL  B
0DD0 1B        2810 TRANS2 DEC  DE
0DD1 1A        2820        LD   A,(DE)
0DD2 FE2C      2830        CP   ","
0DD4 28FA      2840        JR   Z,TRANS2
0DD6 FE2E      2850        CP   "."
0DD8 28F6      2860        JR   Z,TRANS2
0DDA E60F      2870        AND  #F
0DDC ED67      2880        RRD
     1B        2890 TRANS3 DEC  DE
0DDF 1A        2900        LD   A,(DE)
0DE0 FE2C      2910        CP   ","
0DE2 28FA      2920        JR   Z,TRANS3
0DE4 FE2E      2930        CP   "."
0DE6 28F6      2940        JR   Z,TRANS3
0DE8 E60F      2950        AND  #F
0DEA ED67      2960        RRD
0DEC 2B        2970        DEC  HL
0DED 10E1      2980        DJNZ TRANS2
0DEF E1        2990        POP  HL
```

```
0DF0 D1        3000        POP  DE
0DF1 AF        3010        XOR  A
0DF2 B1        3020        OR   C
0DF3 2804      3030        JR   Z,ENDE
0DF5 42        3040        LD   B,D
0DF6 CD2B0D    3050        CALL BCDNR
0DF9 AF        3060 ENDE   XOR  A
0DFA C9        3070        RET
0DFB DF        3080 ERR    RST  #18
0DFC 6B        3090        DEFB #6B;ERRMS
0DFD E1        3100        POP  HL
0DFE C1        3110        POP  BC
0DFF 37        3120        SCF
0E00 C9        3130        RET
               3140 ;
               3150 ;BCD-Ausgabe
               3160 ;Rechtsbündige Ausgabe eines Registers
               3170 ;HL: Register links
               3180 ; B: Länge der auszugebenden Zahl
               3190 ;    in Bytes, bei unverkürzter
               3200 ;    Ausgabe (ohne Rundung etc.)
               3210 ;    = Registerlänge
               3220 ; C: NK-Stellen
               3230 ;
0E01 CB39      3240 RBAUSG SRL  C
0E03 78        3250        LD   A,B
0E04 91        3260        SUB  C
0E05 47        3270        LD   B,A
0E06 7E        3280 TEST1  LD   A,(HL)
     B7        3290        OR   A
     200D      3300        JR   NZ,TEST2
     23        3310        INC  HL
     DF        3320        RST  #18
     69        3330        DEFB #69;SPACE
     DF        3340        RST  #18
0E0E 69        3350        DEFB #69
0E0F 10F5      3360        DJNZ TEST1
0E11 EF        3370        RST  #28
0E12 083000    3380        DEFB #08,#30,0
0E15 1815      3390        JR   KOMMA
0E17 E6F0      3400 TEST2  AND  #F0
0E19 7E        3410        LD   A,(HL)
0E1A 200B      3420        JR   NZ,AUSG2
0E1C F5        3430        PUSH AF
0E1D DF        3440        RST  #18
0E1E 69        3450        DEFB #69
0E1F F1        3460        POP  AF
0E20 DF        3470        RST  #18
0E21 7A        3480        DEFB #7A;B1HEX
0E22 23        3490        INC  HL
0E23 05        3500        DEC  B
0E24 2806      3510        JR   Z,KOMMA
0E26 7E        3520 AUSG1  LD   A,(HL)
0E27 DF        3530 AUSG2  RST  #18
0E28 68        3540        DEFB #68;B2HEX
0E29 23        3550        INC  HL
```

```
0E2A 10FA     3560         DJNZ AUSG1
0E2C 3E2C     3570 KOMMA   LD   A,",
0E2E F7       3580         RST  #30
0E2F 7E       3590 AUSG3   LD   A,(HL)
0E30 DF       3600         RST  #18
0E31 68       3610         DEFB #68
0E32 23       3620         INC  HL
0E33 0D       3630         DEC  C
0E34 20F9     3640         JR   NZ,AUSG3
0E36 C9       3650         RET
              3660 ;
              3670 ;Register-Löschen
              3680 ;HL: Register (links)
              3690 ;BC: Reg.-Länge
              3700 ;
0E37 E5       3710 CLEAR   PUSH HL
0E38 D1       3720         POP  DE
0E39 13       3730         INC  DE
0E3A 0B       3740         DEC  BC
0E3B AF       3750         XOR  A
0E3C 77       3760         LD   (HL),A
0E3D EDB0     3770         LDIR
0E3F C9       3780         RET
              3790 ;
              3800 ;BCD-Rundung auf N-Stellen
              3810 ;N muß eine gerade Zahl sein
              3820 ;HL: Register (links)
              3830 ; C: gew. Stellenzahl
              3840 ;
0E40 CB39     3850 RNDBCD  SRL  C
0E42 0600     3860         LD   B,0
0E44 09       3870         ADD  HL,BC
0E45 7E       3880         LD   A,(HL)
0E46 FE32     3890         CP   50
0E48 D8       3900         RET  C
0E49 2B       3910 RND     DEC  HL
0E4A 7E       3920         LD   A,(HL)
0E4B 3C       3930         INC  A
0E4C 27       3940         DAA
0E4D 77       3950         LD   (HL),A
0E4E D0       3960         RET  NC
0E4F 18F8     3970         JR   RND
              3980 ;
              3990 ;Testprogramm
              4000 ;Das folgende Testprogramm soll
              4010 ;die Anwendung der Routinen er-
              4020 ;leichtern. Es enthält alle kom-
              4030 ;plexeren Unterprogramme (Eingabe,
              4040 ;Ausgabe, Multiplikation) und
              4050 ;soll vor allem die Handhabung
              4060 ;der Register erläutern.
              4070 ;
              4080 ;Zunächst werden 3 Register angelegt.
              4090 ;Wir wollen Operanden von insgesamt
              4100 ;16 Stellen zulassen. Das Ergebnis
              4110 ;einer Multiplikation kann dann 32
              4120 ;Stellen haben, also müssen auch
              4130 ;die Register entsprechen angelegt
              4140 ;werden:
0E51 0010     4150 LEN     EQU  16 ;(Bytes)
0E51 0010     4160 R1      DEFS LEN
0E61 0010     4170 R2      DEFS LEN
0E71 0010     4180 R3      DEFS LEN
              4190 ;
              4200 ;Wir müssen nun noch die Anzahl
              4210 ;der NK-Stellen festlegen:
0E81 0008     4220 NK      EQU  8
              4240 ;Nun können wir das Testprogramm
              4250 ;mit der Eingabe der 2 Operanden
              4260 ;beginnen:
0E81 21510E   4280 DEMO    LD   HL,R1;Reg. löschen
0E84 013000   4290         LD   BC,48;3*LEN
0E87 CD370E   4300         CALL CLEAR
0E8A DF       4310 IN1     RST  #18
0E8B 63       4320         DEFB #63;INLIN
0E8C 21610E   4330         LD   HL,R2
0E8F 0610     4340         LD   B,LEN
0E91 0E10     4350         LD   C,NK+NK;Reserve
0E93 CD930D   4360         CALL BCDIN
0E96 3006     4370         JR   NC,IN2
0E98 ED53290C 4380         LD   (CURPOS),DE
0E9C 18EC     4390         JR   IN1
0E9E DF       4400 IN2     RST  #18
0E9F 63       4410         DEFB #63
0EA0 21710E   4420         LD   HL,R3
0EA3 0610     4430         LD   B,LEN
0EA5 0E10     4440         LD   C,NK+NK
0EA7 CD930D   4450         CALL BCDIN
0EAA 3006     4460         JR   NC MULTI
0EAC ED53290C 4470         LD   (CURPOS),DE
0EB0 18EC     4480         JR   IN2
              4490 ;jetzt kann multipliziert werden:
0EB2 FD21600E 4500 MULTI   LD   IY,R1+LEN-1
0EB6 DD21700E 4510         LD   IX,R2+LEN-1
0EBA 217C0E   4520         LD   HL,R3+LEN-1-4;= -NK/2
0EBD 0610     4530         LD   B,LEN
0EBF 0E08     4540         LD   C,NK
0EC1 CD390D   4550         CALL BCDMUL
0EC4 21510E   4560         LD   HL,R1;Runden nach der 8.
              4570 ;          NK-Stelle
0EC7 0610     4580         LD   B,LEN
0EC9 0E18     4590         LD   C,24
0ECB CD400E   4600         CALL RNDBCD
0ECE 21510E   4610         LD   HL,R1;Ergebnisanzeige mit 8
              4620 ;          NK-Stellen
0ED1 060C     4630         LD   B,12;wegen Rundung
0ED3 0E08     4640         LD   C,8
0ED5 CD010E   4650         CALL RBAUSG
0ED8 DF       4660         RST  #18
0ED9 6A       4670         DEFB #6A  4690 ;G.K.  -  10.1.82
0EDA 18A5     4680         JR   DEMO
```

Hexdump der BCD-Arithmetik

```
0D00 AF FD 7E 00 DD 8E 00 27 C9
0D08 FD 77 00 DD 2B FD 2B 10 C9
0D10 F0 C9 AF FD 7E 00 DD 9E 7B
0D18 00 27 FD 77 00 DD 2B FD C5
0D20 2B 10 F0 C9 AF ED 6F 2B 57
0D28 10 FB C9 C5 E5 AF ED 67 B6
0D30 23 10 FB E1 C1 0D 20 F3 2D
0D38 C9 E5 DD E5 E1 16 00 58 04
0D40 1D AF ED 52 B1 28 05 C5 FB
0D48 CD 2B 0D C1 E1 50 CB 3A 51
0D50 CB 3A 79 CB 3F 82 57 CB 89
0D58 C3 ED 67 4F F5 B7 28 10 AF
0D60 DD E5 FD E5 C5 CD 00 0D B0
0D68 C1 FD E1 DD E1 0D 20 F0 EF
0D70 E5 DD E5 E1 C5 CD 24 0D C8
0D78 C1 E1 F1 CB 43 20 07 ED 3A
0D80 67 2B 15 20 D2 C9 CB 83 3D
0D88 18 CF FE 3A 30 03 FE 30 15
0D90 D0 37 C9 C5 E5 CB 39 78 93
0D98 91 3D 4F 06 00 09 0E 00 DF
0DA0 1A FE 2C 28 11 FE 2E 28 7E
0DA8 0D FE 20 28 18 CD 8A 0D 84
0DB0 38 49 04 13 18 EA 13 1A 84
0DB8 FE 20 28 09 CD 8A 0D 38 B0
0DC0 3A 04 0C 18 F1 CB 40 28 53
0DC8 05 CB 38 04 18 02 CB 38 FE
0DD0 1B 1A FE 2C 28 FA FE 2E 8A
0DD8 28 F6 E6 0F ED 67 1B 1A 81
0DE0 FE 2C 28 FA FE 2E 28 F6 83
0DE8 E6 0F ED 67 2B 10 E1 E1 3B
0DF0 D1 AF B1 28 04 42 CD 2B 94
0DF8 0D AF C9 DF 6B E1 C1 37 AD
0E00 C9 CB 39 78 91 47 7E B7 60
0E08 20 0D 23 DF 69 DF 69 10 06
0E10 F5 EF 08 30 00 18 15 E6 4D
0E18 F0 7E 20 0B F5 DF 69 F1 ED
0E20 DF 7A 23 05 28 06 7E DF 3A
0E28 68 23 10 FA 3E 2C F7 7E AA
0E30 DF 68 23 0D 20 F9 C9 E5 7C
0E38 D1 13 0B AF 77 ED B0 C9 C1
0E40 CB 39 06 00 09 7E FE 32 0F
0E48 D8 2B 7E 3C 27 77 D0 18 99
0E50 F8 CE 67 33 99 CC E6 73 7C
0E58 39 9C CE 67 33 99 CC E6 EE
0E60 73 39 9C CE 67 33 99 CC 83
0E68 E6 73 39 9C CE 67 33 99 A5
0E70 CC E6 73 39 9C CE 67 33 E0
0E78 99 CC E6 73 39 9C CE 67 4E
0E80 33 21 51 0E 01 30 00 CD 3F
0E88 37 0E DF 63 21 61 0E 06 B3
0E90 10 0E 10 CD 93 0D 30 06 6F
0E98 ED 53 29 0C 18 EC DF 63 61
0EA0 21 71 0E 06 10 0E 10 CD 4F
0EA8 93 0D 30 06 ED 53 29 0C 01
0EB0 18 EC FD 21 60 0E DD 21 4C
0EB8 70 0E 21 7C 0E 06 10 0E 13
0EC0 08 CD 39 0D 21 51 0E 06 6F
0EC8 10 0E 18 CD 40 0E 21 51 99
0ED0 0E 06 0C 0E 08 CD 01 0E F0
0ED8 DF 6A 18 A5 CC E6 73 39 4A
```

Erweiterter Texteditor

von Peter Urban

-Der Texteditor aus NC 2/1981 hat sieben zusaetzliche Befehle, die die Bedienung des Kassettenrecordes vereinfachen (SAVE, VERIFY, READ), die Umschaltung der Tastatur ermoeglichen (K0, K1) und die Druckerroutine ein- und ausschalten (Print n1 n2, List n1 n2 -mit Zeilennr.).

```
2230  3E 53 41 56 45 0D        = Erkennung "SAVE"
2236  C3 70 22                 = JP 2270
2239  3E 56 45 52 49 46 59 0D  = Erkennung "VERIFY"
2241  C3 90 22                 = JP 2290
2244  3E 52 45 41 44 0D        = Erkennung "READ"
224A  C3 A0 22                 = JP 22A0
224D  3E 4B 30 0D              = Erkennung "K0"
2251  C3 B0 22                 = JP 22B0
2254  3E 4B 31 0D              = Erkennung "K1"
2258  C3 C0 22                 = JP 22C0
225B  3E 50 52 49 4E 54 0D     = Erkennung "PRINT n1 n2"
2262  C3 D0 22                 = JP 22D0
2265  3E 4C 49 53 54 0D        = Erkennung "LIST n1 n2"
226B  C3 E0 22                 = JP 22E0

2270  21 00 24       = LD HL,2400   Beginn Workspace
2273  22 0C 0C       = LD (0C0C),HL
2276  2A 02 24       = LD HL,(2402)
2279  23             = INC HL
227A  22 0E 0C       = LD (0C0E),HL
227D  CD E8 04       = CALL 04E8
2280  C3 D1 15       = JP 15D1
2290  DF 56          = Restart "V"
2292  C3 D1 15       = JP 15D1
22A0  21 28 0C       = LD HL,0C28
22A3  36 52          = LD (HL),N
22A5  DF 52          = Restart "R"
22A7  C3 D1 15       = JP 15D1
22B0  21 27 0C       = LD HL,0C27
22B3  36 00          = LD (HL),00
22B5  C3 D1 15       = JP 15D1
22C0  21 27 0C       = LD HL,0C27
22C3  36 01          = LD (HL),01
22C5  C3 D1 15       = JP 15D1
22D0  CD 1E 18       = CALL 181E
22D3  21 00 00       = LD HL,0000
22D6  DF 50          = Restart "X0"
22D8  CD CF 12       = CALL 12CF
22DB  DF 4E          = Restart "N"
22DD  C3 D1 15       = JP 15D1
22E0  CD 1E 18       = CALL 181E
22E3  21 00 00       = LD HL,0000
22E6  DF 50          = Restart "X0"
22E8  CD D6 12       = CALL 12D6
22EB  DF 4E          = Restart "N"
22ED  C3 D1 15       = JP 15D1
2300  21 00 24       = LD HL,2400   Kaltstart
2303  11 01 24       = LD DE,2401
2306  01 FF DB       = LD BC,DBFF
2309  36 00          = LD (HL),00
230B  DF 43          = Restart "C"
230D  21 00 24       = LD HL,2400
2310  23             = INC HL
2311  36 25          = LD (HL),25   Startadresse auf File
2313  23 23 23       = INC HL / INC
2316  36 FF          = LD (HL),FF   Top of RAM
2318  23             = INC HL
2319  36 FF          = LD (HL),FF
231B  21 00 24       = LD HL,240A
231E  36 C3          = LD(HL),C3        Initialisierung
2320  23             = INC HL
2321  36 50          = LD(HL),50
2323  23             = INC HL
2324  36 23          = LD (HL),23
2326  23             = INC HL
2327  36 C3          = LD (HL),C3       Output Rout.
2329  23             = INC HL
232A  36 30          = LD (HL),30
232C  23 23          = INC HL / INC HL
232E  36 DF          = LD (HL),DF       Rückkehr zu NAS9
2330  23             = INC HL
2331  36 5B          = LD (HL),5B
2333  23 23          = INC HL / INC HL
2335  36 C9          = LD (HL),C9
2340  DD 21 00 24    = LD IX,2400       Warmstart
2344  C3 BF 15       = JP 15BF
```

Der Autor schickt den Editor mit Erweiterung auf Cassette gerne in Form eines "Rundlaufs" an Interessenten. Bitte Cassettenformat angeben.

Sprachsynthese
von Günter Böhm

Im Dezember 81 veröffentlichte die Zeitschrift Elektor eine Sprachplatine, die um den Chip TMS 5100 von Texas Instruments aufgebaut ist. Die Platine ist leicht über PIO an den NASCOM anzuschließen und befähigt den Rechner zu sprechen. Der Wortschatz besteht bisher aus 200 Wörtern, die in einem externen EPROM Speicher auf der Sprachplatine abgelegt sind. Bisher sind 16 K belegt, der Wortschatz kann aber erweitert werden bis auf 64k, ohne den Speicherbereich des Rechners zu belasten. Im nächsten Heft wollen wir noch einige Information zum Sprachprozessor TMS 5100 veröffentlichen. Für diesmal nur Anschlußhinweise und Programme zur Anwendung auf dem NASCOM.

Der Anschluß an den "PIO-Bus" ist sehr einfach. Ich habe mir ein Stück Lochrasterplatine auf die Sprachplatine geschraubt und einen 31-pol. Stecker eingelötet. Die Anschlüsse zwischen Stecker und Platine wurden frei verdrahtet. So läßt sich die Sprachplatine in gewohnter Weise in den "Bus" stekken.

Die Daten D0 bis D7 wurden an Port B gelegt, die 5 Kontrolleitungen an Port A wie in folgender Tabelle.

Die benötigten Spannungen wurden der Grundplatine über den 31-pol. Stecker entnommen. Nachfolgende Tabelle zeigt die Steuerworte, die zur Kontrolle des VSP (voice-synthesis processor) von Port A übergeben werden müssen.

```
PORT A
Contr.Mode  0     0     0     0   0
BIT         4     3     2     1   0
VSP         LDA1  LDA0  CCLK  C1  C0   HEX
RESET       1     1     1     1   1    1F
TEST BUSY   1     1     1     0   0    1C
LDA1        0     1     1     0   0    0C
LDA0        1     0     1     0   0    14
TALK        1     1     1     1   0    1E

CCLK Impulse entstehen durch jeweiliges Rücksetzen und Setzen von Bit 2


Steuerworte für PORT MODE
PORT B OUTPUT  0F

PORT A CONTRUL CF
Controlword    20
```

Zur Software:
TALK1 enthält die Unterprogramme, die zur Steuerung des Sprachprozessors nötig sind.
INIT initialisiert die Ports zum gewünschten Modus.
RESET macht den Sprachprozessor zur Sprachausgabe bereit. (Beide Routinen werden nur einmal am Programmbeginn mit C80 aufgerufen).
ADDR lädt die Wortadresse, die in (HL) stehen muß in den Wortspeicher des Prozessors und liest das Wort mit TALK aus. Das Unterprogramm wartet dann in der Schleife BUSY, bis das Wort beendet ist, und kehrt zum Hauptprogramm zurück.
Wenn Sie das Hauptprogramm bei CF7 starten (START1), gibt der Prozessor jeweils das Wort aus, dessen Startadresse Sie eingeben. Mit NL alleine wird das Wort wiederholt.
Start2 (D04) erwartet, daß Sie irgendwo im Speicher die Adressen der gewünschten Wörter abgelegthaben, wobei 00 FF als Pause und 00 FE als Ende der Mitteilung interpretiert wird. Durch Eingabe der Anfangsadresse des "Satzes" und NL gibt der Prozessor den ganzen Satz aus.

```
                0001 ; TALK1-DEZ81
                0010 ; PROGR.ZUR SPRACHPLATINE
                0020 ; 1.SUBROUTINEN
                0030 ; 2.WORTTEST
                0040 ; 3.SATZTEST
                0050 ;
                0060 ; NASSYS 3
                0070 ; 28.DEZ.1981
                0080 ; GUENTER BOEHM, K'HE
                0090 ;
0C80            0100        ORG   #C80
0006            0110 ACONTR EQU   #06
0007            0120 BCONTR EQU   #07
0004            0130 ADATA  EQU   #04
0005            0140 BDATA  EQU   #05
                0150 ;
0C80 3E0F       0160 INIT   LD    A,#F ;B OUTPUT
0C82 D307       0170        OUT   (BCONTR),A
0C84 3ECF       0180        LD    A,#CF    ;A CONTR.
0C86 D306       0190        OUT   (ACONTR),A
0C88 3EE0       0200        LD    A,#E0
0C8A D306       0210        OUT   (ACONTR),A
0C8C 3E1F       0220 RESET  LD    A,#1F
0C8E D304       0230        OUT   (ADATA),A
0C90 0603       0240        LD    B,3       ;ANZ.IMPULS
0C92 D709       0250        DEFW  #9D7 ;RCAL IMPULS
0C94 3E1C       0260        LD    A,#1C
0C96 D304       0270        OUT   (ADATA),A
0C98 0603       0280        LD    B,3
0C9A D701       0290        DEFW  #1D7 ;RCAL IMPULS
0C9C C9         0300        RET
                0310 ;
0C9D CB97       0320 IMPULS RES   2,A
0C9F D304       0330        OUT   (ADATA),A
0CA1 E5         0340        PUSH  HL      ;DELAY
0CA2 E1         0350        POP   HL
0CA3 CBD7       0360        SET   2,A
```

```
0CA5 D304      0370         OUT  (ADATA),A
0CA7 10F4      0380         DJNZ IMPULS
0CA9 C9        0390         RET
0CAA 7E        0400 ADDR    LD   A,(HL)
0CAB D305      0410         OUT  (BDATA),A
0CAD 3E0C      0420         LD   A,#C      ;LDA1
0CAF D304      0430         OUT  (ADATA),A
0CB1 00        0440         NOP
0CB2 3E1C      0450         LD   A,#1C
0CB4 D304      0460         OUT  (ADATA),A
0CB6 23        0470         INC  HL
0CB7 7E        0480         LD   A,(HL)
0CB8 D305      0490         OUT  (BDATA),A
0CBA 3E14      0500         LD   A,#14     ;LDA0
0CBC D304      0510         OUT  (ADATA),A
0CBE 00        0520         NOP
0CBF 3E1C      0530         LD   A,#1C
0CC1 D304      0540         OUT  (ADATA),A
0CC3 3E1E      0550 TALK    LD   A,#1E
0CC5 D304      0560         OUT  (ADATA),A
0CC7 0601      0570         LD   B,1
0CC9 D7D2      0580         DEFW #D2D7 ;RCAL IMPULS
0CCB 3E1C      0590         LD   A,#1C
0CCD D304      0600         OUT  (ADATA),A
0CCF 0602      0610         LD   B,2
0CD1 D7CA      0620         DEFW #CAD7 ;RCAL IMPULS
0CD3 DB04      0630 BUSY    IN   A,(ADATA)
0CD5 CB6F      0640         BIT  5,A
0CD7 28FA      0650         JR   Z BUSY
0CD9 0601      0660         LD   B,1
0CDB D7C0      0670         DEFW #C0D7 ;RCAL IMPULS
0CDD C9        0680         RET
               0690 ;
0CDE 23        0700 SNTCE   INC  HL
0CDF 7E        0710         LD   A,(HL)
0CE0 FEFE      0720         CP   #FE ;ENDZEICHEN
0CE2 C8        0730         RET  Z
0CE3 FEFF      0740         CP   #FF ;PAUSEZEICHEN
0CE5 2806      0750         JR   Z PAUSE
0CE7 2B        0760         DEC  HL
0CE8 D7C0      0770         DEFW #C0D7 ;RCAL ADDR
0CEA 23        0780         INC  HL
0CEB 18F1      0790         JR   SNTCE
0CED 23        0800 PAUSE   INC  HL
0CEE 0630      0810         LD   B,48
0CF0 CD4000    0820         CALL #40   ;TDEL2
0CF3 18E9      0830         JR   SNTCE
               0840 ;SPRUNGHILFE
0CF5-1889      0850 RCAL    JR   INIT
               0860 ;START FUER AUFRUF
               0870 ;EINZELNER ADRESSEN
               0880 ;
0CF7 D787      0890 START1  DEFW #87D7;RCAL INIT
63DF           0900 INLIN   EQU  #63DF
79DF           0910 RLIN    EQU  #79DF
0C0C           0920 ARG1    EQU  #C0C
0CF9 DF63      0930         DEFW INLIN
0CFB DF79      0940         DEFW RLIN
0CFD 210C0C    0950         LD   HL,ARG1
0CFD 210C0C    950          LD   HL,ARG1
0D00 D7A8      0960         DEFW #A8D7 ;RCAL ADDR
0D02 18F5      0970         JR   2START1
               0980 ;
               0990 ;START ZUM SPRECHEN
               1000 ;VON SAETZEN,DIE IN
               1010 ;HL BEGINNEN
               1020 ;
0D04 D7EF      1030 START2  DEFW #EFD7 ;RCAL INIT
0D06 DF63      1040         DEFW INLIN
0D08 DF79      1050         DEFW RLIN
0D0A 2A0C0C    1060         LD   HL,(ARG1)
0D0D D7CF      1070         DEFW #CFD7 ;RCAL SNTCE
0D0F 18F5      1080         JR   2START2
```

Die Anwendung des TMS 5100 scheint aber nicht nur zur Sprachausgabe interessant, vielmehr bietet er eine Fülle von komplexen Geräuschen, wenn man als Wortadresse nicht den Anfang sondern einen Wortteil angibt. Wenn man den Prozessor mit Informationen aus dem RAM versorgt (meine Experimente in dieser Richtung waren noch nicht erfolgreich) müßte er sich auch möglicherweise für musikalische Anwendung eignen.

Das Programm TALK4 gibt die Möglichkeit, sämtliche Einsprungadressen in ihrer Verwertbarkeit als Teilwort oder Geräusch zu testen. Die folgende Tabelle gibt die Adressen von möglichen Teilwörtern oder Lauten, die allerdings nicht alle so überzeugend klingen, daß man sie durchweg verwerten könnte. Auf eine Tabelle von Geräuschen habe ich verzichtet, da sich diese erstens schlecht niederschreiben lassen und zweitens eine Wertung der Anwendungseignung haupsächlich eine Geschmacksfrage ist.

```
               0010 ; TALK4-JAN82
               0020 ; TESTPROGRAMM FUER SPRACH-
               0030 ; PLATINE IN EINZELSCHRITTEN
               0040 ; I=INPUT START
               0050 ; PUNKT=INCR.ZAEHLER
               0060 ; KOMMA=DECR.ZAEHLER
               0070 ; R=REPEAT
               0080 ; P=PRINT ZAEHLER (ES KOENNEN
               0090 ; KOMMENTARE EINGEGEBEN WERDEN
               0100 ; WEITER MIT "ENTER"
               0110 ; N=ZURUECK ZU NASSYS3
               0120 ; 3.1.82
               0130 ; GUENTER BOEHM, K'HE
               0140 ;
0E00           0150         ORG  #E00
0E00 EF        0160 START   RST  40
0E01 49        0170         DEFM "INPUT WORD START : "
0E14 0D00      0180         DEFW #D ;CRLF
0E16 CD800C    0190         CALL #C80 ;INIT
0E19 DF63      0200         DEFW #63DF
0E1B DF79      0210         DEFW #79DF
0E1D ED5B0C0C  0220         LD   DE,(#C0C)
               0230 ;
0E21 D5        0240 OUT     PUSH DE
0E22 E1        0250         POP  HL
0E23 DF66      0260         DEFW #66DF
0E25 DF6A      0270         DEFW #6ADF ;CRLF
0E27 210C0C    0280         LD   HL,#C0C
0E2A ED530C0C  0290         LD   (#C0C),DE
0E2E CDAA0C    0300         CALL #CAA ; ADDR;SPRECHEN
0E31 D5        0310 BLINK   PUSH DE
0E32 DF7B      0320         DEFW #7BDF;BLINK
0E34 D1        0330         POP  DE
0E35 FE2E      0340         CP   #2E ;PUNKT
0E37 2003      0350         JR   NZ KOMMA
0E39 13        0360         INC  DE
0E3A 18E5      0370         JR   OUT
0E3C FE2C      0380 KOMMA   CP   #2C
0E3E 2003      0390         JR   NZ RR
0E40 1B        0400         DEC  DE
0E41 18DE      0410         JR   OUT
0E43 FE52      0420 RR      CP   #52 ;REPEAT
0E45 2002      0430         JR   NZ PP
0E47 18D8      0440         JR   OUT
0E49 FE50      0450 PP      CP   #50 ;PRINT
0E4B 2010      0460         JR   NZ NN
0E4D 2E20      0470         LD   L,#20 ;X-OPTION
0E4F DF58      0480         DEFW #58DF ;X-COMMAND
0E51 D5        0490         PUSH DE
0E52 E1        0500         POP  HL
0E53 DF66      0510         DEFW #66DF ;PRINT HL
0E55 D5        0520         PUSH DE
```

```
0E56 DF63      0530         DEFW #63DF;INLIN/HIER WERDEN
               0540 ;KOMMENTARE AN DEN DRUCKER GEGEBEN
0E58 DF4E      0550         DEFW #4EDF ;NORMAL-MODE
0E5A D1        0560         POP  DE
0E5B 18D4      0570         JR   BLINK
0E5D FE4E      0580 NN      CP   #4E ;BACK TO MONITOR
0E5F 2002      0590         JR   NZ EE
0E61 DF5B      0600         DEFW #5BDF ;MRET
0E63 FE49      0610 EE      CP   #49 ;INPUT NEUER START
0E65 20CA      0620         JR   NZ BLINK
0E67 1897      0630         JR   START
```

Wenn Sie schon (oder bald) mit der Platine arbeiten, wäre ich für Erfahrungsberichte oder Vorschläge zur ernsthaften Anwendung sehr dankbar.
Eine Möglichkeit der "ernsthaften" Anwendung ist das Auslesen eines bestimmten Speicherbereichs in Hexwerten. Wenn ein Speichervergleich nicht möglich ist, da bestimmte Adressen oder Werte bewußt verändert wurden, hat das Programm TALK den Vorteil, daß man nicht immer Bildschirm und Hexdump vergleichen muß, sondern gemütlich das Listing abhakt, während der Rechner vorliest.
Die Hauptprogramme benutzen übrigens immer die zu Anfang beschriebenen Subroutinen an den angegebenen Adressen. Diese sind im Prinzip voll verschieblich.

Wortanalyse Sprachplatine (erste 4K)

```
0000 AGAIN    03D5 'S     0946 UN-
0010 GAIN     03E4 ALL    0970 OR
002B AN       03EF BALL   099A OUT
003D TOWN     041A AN     09D6 THE(E)
0048 DOWN     0446 AND    09E6 D  (D?
0058 BOMB     045B ENDE   09F1 D
0084 HELLO    0484 ANY    0A08 THE
009C LOW      04B4 ARE    0A13 GE-
00A7 NO       04D0 AT     0A44 THERE
00D0 MESSAGE  04F2 CAN    0A54 AL
00EB -ASSAGE  0522 DID    0A78 THIS
00FC -SAGE    0566 DO     0A9E USE
0103 DITCH    05A0 DOES   0AAE BOOZE
011F -DSCH    05C6 US     0AB9 DUCE
0138 MISTAKE  05FC FOR    0AF6 WHAT
0149 -ISTAKE  0603 CORPS  0B20 WHEN
014F STEAK    0634 FROM   0B6C WHERE
016B ACHE     063A DRUM   0B97 A(M)
0176 BAKE     064A -UM    0BB4 WILL
0198 NAME     0662 GOT    0BCA  ILL
01A8 AIM      0669 GOT    0C06 WITH
01CE "NIEBE"  069C HAVE   0C1C PINNED
01EE ENDE     06A9 -AVE   0C94 YES
0222 PLEASE   06DA HOW    0CB6 BEAU
0262 PUT      06E1 POW    0CC6 YOU
0269 -UT      06E8 OW     0CF8 YOUR
028C REPEAT   0724 IN     0D2E A
029B PEAT-    0760 IS     0D6C B
02CC RIGHT    079C IT     0DA8 C
02D7 BRIGHT   07B4 ME     0DEC D
02F2 -ITE     0800 MUCH   0E36 E
02FD BAKE     082A MY     0E60 F
0324 THANK    082F WHY    0E94 G
032A TANK     0856 NO     0EC4 H
032C BANK     0866 O      0F03 TIE
036E UP       0890 NOT    0EFE I
0388 WANT     08C4 NOW    0F34 J
0393 -ANT     0906 OF     0F4C DAY
03CE 'S       0911 BOVE   0F80 K
```

```
               0010 ;TALK2-DEZ81
               0020 ;PROGR.FUER SPRACHPL.
               0030 ;AUSLESEN VON HEXZ.
               0040 ;AUS SPEICHER AB HL
               0050 ;NASSYS 3
               0060 ;29.12.81
               0070 ;GUENTER BOEHM
               0080 ;
OD11           0090         ORG  #D11
0CAA           0100 ADDR    EQU  #CAA
OD11 7E        0110 RDHEX   LD   A,(HL); SPEICHER
OD12 E5        0120         PUSH HL       ; RETTEN
OD13 CB3F      0130         SRL  A
OD15 CB3F      0140         SRL  A
OD17 CB3F      0150         SRL  A
OD19 CB3F      0160         SRL  A        ; HIGHBYTE
OD1B 87        0170         ADD  A,A      ; 2BYTE WOER
TER
OD1C 215A0D    0180         LD   HL,TABLE   ; WORTTABEL
LE
OD1F 0600      0190         LD   B,0
OD21 4F        0200         LD   C,A
OD22 09        0210         ADD  HL,BC      ; +HEXWERT*2

OD23 CDAA0C    0220         CALL ADDR
OD26 E1        0230         POP  HL          ; SPEICHER
OD27 7E        0240         LD   A,(HL)
OD28 E60F      0250         AND  #F          ; LOWBYTE
OD2A 87        0260 ZWBTE   ADD  A,A         ; 2BYTE
OD2B E5        0270         PUSH HL        ; SP.RETTEN
OD2C 215A0D    0280         LD   HL,TABLE
OD2F 4F        0290         LD   C,A
OD30 09        0300         ADD  HL,BC      ; WORTT.+HEX

OD31 CDAA0C    0310         CALL ADDR
OD34 E1        0320         POP  HL
OD35 23        0330         INC  HL
OD36 C9        0340         RET  ; NEXT HEXC.
               0350 ;
0C80           0360 INIT    EQU  #C80
OD37 CD800C    0370 START3  CALL INIT
OD3A DF63      0380         DEFW #63DF     ;INLIN
OD3C DF79      0390         DEFW #79DF     ;RLIN
OD3E 2A0C0C    0400         LD   HL,(#C0C)  ;ARG1
OD41 ED5B0E0C  0410         LD   DE,(#C0E)  ;ARG2
OD45 CD110D    0420 LOOP    CALL RDHEX
OD48 0630      0421         LD   B,48       ;DELAY
OD4A CD4000    0422         CALL #40
OD4D AF        0430         XOR  A          ;RESET CARRY

OD4E ED52      0440         SBC  HL,DE     ;TEST AUF END
E
OD50 19        0450         ADD  HL,DE
OD51 38F2      0460         JR   C LOOP
OD53 3E10      0470         LD   A,#10     ;WORD"STOP"
OD55 CD2A0D    0480         CALL ZWBTE
OD58 DF5B      0490         DEFW #5BDF     ;RET.MONITOR
OD5A 4613      0500 TABLE   DEFW #1346 ;ZERO
OD5C A413      0510         DEFW #13A4       ;ONE
OD5E F613      0520         DEFW #13F6       ;TWO ETC.
OD60 3014      0530         DEFW #1430
OD62 7414      0540         DEFW #1474
OD64 C214      0550         DEFW #14C2
OD66 1015      0560         DEFW #1510
OD68 4E15      0570         DEFW #154E
OD6A A415      0580         DEFW #15A4
OD6C D415      0590         DEFW #15D4
OD6E 2E0D      0600         DEFW #D2E         ;A
OD70 6C0D      0610         DEFW #D6C
OD72 A80D      0620         DEFW #DA8
OD74 EC0D      0630         DEFW #DEC
OD76 360E      0640         DEFW #E36
OD78 600E      0650         DEFW #E60
OD7A 9C2C      0660         DEFW #2C9C
```

Kleinanzeigen

Jeder Abonnent kann Kleinanzeigen bis 40 Wörter aufgeben!

VERKAUFE NASCOM 1 , 32 K RAM, Graphic, erweiterte Tastatur im Gehäuse, 8K BASIC (ROM), EPROM Programmiergerät f.2708 & 2716, 2 Gehäuse, Cassettenrecorder
Preis VB 1200 .-
Außerdem 2 8" Floppy Laufwerke
1 TTY mit Lochstr.stanzer & Le.
Preis VS
Tel [redacted] (8-16³°)

NASCOM 2 , 32 K, ZEAP 2.0, NASSYS 3, Gehäuse, Lüfter, Recorder etc. umständehalber abzugeben VB 2300 .-
Böttchers [redacted] ab 17°°

VERKAUFE wegen Hobbyaufgabe NASCOM 1 mit Monitor in Industriegehäuse eingebaut. VB 1600 .-
Tel. [redacted] nach 18°°

VERKAUFE Fernschreiber LO 15 mit Lochstreifenstanzer u. Leser, Ansteuerinterface DM 135 .-
Biete sehr komfortables BASIC Programm zur Trafoberechnung zum Tausch oder Verkauf für N 2 auf CC an (8K)
Klaus Bott [redacted]

Speicherplatine für NASCOM mit Sockeln, Kondensatoren und Widerständen, jedoch ohne ICs für DM 100 .- (VB) zu verkaufen.
12 K EPROM Platine (2708) für DM 100 .- (VB) zu verkaufen.
B.Wiedemann Tel. [redacted]

VERKAUFE ZEAP 2.0 (in 4 orig.2708) 95.-
& BASIC 4.7 (in original 8K ROM) 120.-
Wer kann mir INMC-News zum Kopieren zusenden (gegen Unkosten u. Obulus)?
P.Hofmann [redacted]

SUCHE NASCOM-PIO-BOARD
DMA-Controller
8" Floppy-Controller für N 2
mit NASSYS 3
Klaus Roth Tel [redacted]

CLD-Floppy System komplett & Double orec. Microsoft BASIC & 10 Discs mit div. Software DM 1000 .-
Tel. [redacted]

Suche leihweise (für einige Tage) die Beschreibung des NASCOM PIO-Boards.
Angebote an
G.Steuerwald Tel. [redacted]

VERKAUFE MDCR-Controller UNICON 1.4 in EPROM (Beschreibung s.Heft12-81)
Standardversion DM 65.-
Spezialversion DM 95.-
NASSYS 5 DM 50.-
Johannes C. Lotter Tel. [redacted]

Wir bedauern zutiefst das Ableben des

STRICHCODE

Von vielen Lesern abgelehnt, konnte er nur einige bescheidene Versuche im NASCOM Journal erleben, bis er durch eifriges Bemühen seiner Widersacher nun endlich dahingerafft wurde.
Es verbleiben in tiefer Trauer die Hinterbliebenen
Hexdump und Listing

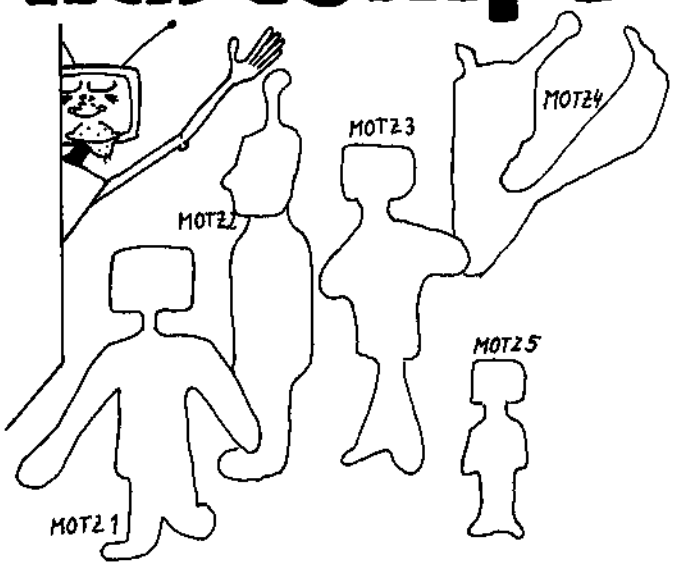
BUGS in Heft 11/12-81

FSK-Modem: UART OUT sollte mit Pin 30 (statt 28) und Clock mit Pin 25 (statt 26) am Stecker verbunden werden. (Ist auf der Folie bereits geändert.)

MDCR-Interface: Die Anschlüsse 1 und 16 an IC 3 müssen vertauscht werden. (Pin 16 an +12 V)

Die Monitor-Modifikation von Rüdiger Maurer bezog sich auf NASSYS 1 (nicht auf NASSYS 3, wie fälschlich angegeben)

nascompl



Hallo liebe Leser,
mit Bedauern mußte ich beim Wühlen durch die Leserpost feststellen, daß sich unser Leserkreis nun langsam in zwei Lager aufspaltet: einmal die wissenschaftlichen Typen und dann die Spielernaturen. Beide Gruppen haben nun die Eigenschaft, daß sie uns anbieten, Artikel nach ihrem Geschmack in der Mitte des Heftes abzudrucken, damit die "andere Partei" nach dem Entfernen der Seiten dadurch nicht belästigt wird. Ich habe nun mit dem Drucker verhandelt; aber er weigert sich hartnäckig, ein Heft mit zwei Mitten herzustellen. Da nun die Redaktion der Meinung ist, es wäre durchaus Raum für beide Gruppen in einem Heft dieses Umfangs und man müßte nicht unbedingt zwei Hefte herausbringen, habe ich mir diesmal einen besonderen Leser-Service ausgedacht:den MOTZ (Modul zur Oberflächenbehandlung Trivialer Zeichnungen). Dabei handelt es sich um Aufkleber, mit denen Sie z.B. alle NASCOMPLs in einem Heft zukleben können. Aus Platzgründen verschieben wir die Veröffentlichung des MAUL (Modul zum Auslöschen Unbequemer Listings) und des SABBER (SichtABdeckung Banaler Erzeugnisse) auf das nächste Heft. Für die "andere Gruppe" ist mir noch nichts eingefallen. Bis zu meinem Geistesblitz empfehle ich betroffenen Lesern, sofort beim Anblick eines Artikels mit unbequemem Niveau die Hand vor die Augen zu halten. Auf diese Art könnte doch wohl jeder Leser zufriedengestellt werden.
In diesem Sinne Ihr NASCOMPL

IMPRESSUM

REDAKTION: Günter Böhm, Günter Kreidl
Wolfgang Mayer-Gürr, Josef Zeller
RESSORTS:
MASCHINENPROGRAMME:
Günter Böhm, [redacted]
[redacted] Karlsruhe, 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.Ø7274/2756 Telex 453500 mksd
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.
Bezugsmöglichkeiten:Durch Bestellung bei MK Systemtechnik (beigefügte Bestellkarte).
Bankverbindungen: Alle Zahlungen für das NASCOM JOURNAL unter Angabe der Rechnungsnummer nur (!!) an das folgende Konto: Fa. Michael Klein, Sonderkonto 29926-674 beim Postscheckamt Ludwigshafen.
Zahlung: 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.