

# **nascocom journal**

**Zeitschrift für Anwender des NASCOM 1 oder NASCOM 2**

---

2. Jahrgang · Juni 1981 · Ausgabe 6

---

**Herausgeber:**

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

---

**MK-Systemtechnik** Thomas Gräfenecker · Kriegsstraße 164 · 7500 Karlsruhe

---

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:**

- |    |   |                                      |
|----|---|--------------------------------------|
| 1  | <b>NASCOM</b> Journal Intern                                  | Red.                                 |
| 4  | <b>Soundgenerator</b>   | G. Böhm                              |
| 7  | <b>NASCOMPL, bonbon</b>                                       | Red.                                 |
| 8  | Inversionsprobleme<br>Dez/Hex in BASIC                        | H. Molle<br>U. Wurditsch             |
| 9  | Verb. Melodien<br>Streifenkiller<br>Thema: <b>Erweiterung</b> | Red.<br>U. Wurditsch<br>U. Wurditsch |
| 10 | Druckerinterface<br>2716 auf RAM Karte                        | J. C. Lotter<br>J. C. Lotter         |
| 11 | Dig. Cassettenrecorder  | J. C. Lotter                         |
| 12 | <b>Relocator</b>  | G. Kreidl                            |
| 15 | Jahresinhaltsverzeichnis 80                                   | Red.                                 |
| 17 | Relocator   | G. Böhm                              |
| 19 | <b>Preis Ausschreiben</b><br>Kleinanzeigen                    | Red.                                 |

# NASCOM Journal



Liebe Leser,  
das NASCOM Journal hat nicht nur ein neues Cover, wie Sie sicher sofort bemerkt haben, sondern auch einen neuen Chefredakteur, wie Sie hoffentlich bald bemerken werden. Als das letzte Journal mit dem Aufruf in Druck ging, hatte mir Herr Klein die Aufgabe bereits angeboten, und ich hatte sofort zugesagt. Hoffentlich sind nun einige Leser nicht enttäuscht! Ich darf mich zunächst kurz vorstellen:  
1947 geboren - Pädagogik studiert - Realschullehrerausbildung - unterrichtete Englisch, Deutsch Musik - kam zum Computer über den Bau von Musiksynthesizern - bin auf diesem Gebiet absoluter Autodidakt - hoffe, mich auch durch Erfahrungsaustausch mit Lesern weiterzubilden - interessiere mich eigentlich für alles außer Sport - habe große Freude an der Gestaltung des Journals - wünsche, daß ich es noch eine Weile zu Ihrer Zufriedenheit leiten kann.

Seit der ersten Ausgabe habe ich das Journal selbst abonniert und, wie Sie, liebe Leser, alle "Höhen und Tiefen" miterlebt. Es hat mir oft mehr gebracht als andere Computerzeitschriften. Aber ich habe auch oft unter dem schlechten Druck "gelitten", habe wochenlang den Postboten mit der Frage nach dem Journal gequält und habe Stunden vertan, um ein Programm zum Laufen zu bringen, das an entscheidenden Stellen "verwanzt" war.

Diese Fehler sollen nun nach Möglichkeit nicht mehr auftreten.

Ich habe vor, Ihnen das Journal immer rechtzeitig fertigzustellen; lieber eine dünnere Ausgabe (die geringere Seitenzahl kann man später wieder ausgleichen) als gar keine.

Es sollen keine schlecht gedruckten Listings mehr erscheinen. Alle Artikel, die sich zum Druck nicht eignen, werden im Blocksatz zweispaltig verarbeitet, unscharfe Listings nochmals ausgedruckt. Dieses zweispaltige Verfahren hat den Vorteil, daß wir etwa die doppelte Textmenge in einem Heft unterbringen können. Ein 16-seitiges Journal bringt Ihnen praktisch 32 Schreibmaschinenseiten! Zudem ist dieses Format leichter lesbar. Wenn Sie Manuskripte einschicken, halten Sie den Text bitte auf 11,5 cm Breite. Dann können wir ihn möglicherweise direkt drucken.

Wir wollen versuchen, Fehler in Programmen zu vermeiden. Dazu haben wir ein Team gebildet, das die eingesandten Programme prüfen und aufbereiten soll, da diese Arbeit von einer Person absolut nicht bewältigt werden könnte.

Herr Günter Kreidl stellte sich zur Verfügung, die NASSYS-Programme zu bearbeiten. Er hat diesbezüglich noch ein paar gute Ideen, die er Ihnen im nächsten Heft vorstellen wird.

Herr Wolfgang Mayer-Gürr wird freundlicherweise die BASIC-Programme überarbeiten. (Er ist dafür vorzüglich mit Diskettensystem etc. ausgerüstet.)

Für die Bearbeitung der Hardware haben wir Herrn Josef Zeller gefunden. Wenn Sie seinen Artikel im letzten Heft über den ECB-Bus und die hervorragende Zeichnung gesehen haben, wissen Sie, daß er dafür genau der richtige Mann ist.

Ich selbst werde mich mit den T2/T4 Problemen herumschlagen, wenn die übrige Redaktionsarbeit Zeit dazu läßt.

Sie sollten Ihre Beiträge in Zukunft gleich an die zuständigen Herren senden, die sie nach Überprüfung sofort weiterleiten.

Und damit sind wir bereits am Kernpunkt: DAS NASCOM JOURNAL KANN OHNE DIE BEITRÄGE DER LESER NICHT BESTEHEN!!

Wir sind auf Ihre Mitarbeit angewiesen

und freuen uns über den kleinsten Beitrag. Es müssen nicht gleich großartige Programme sein; viele Leser fangen jetzt erst klein an und können jeden Tip gut gebrauchen. Das soll allerdings nicht heißen, daß wir nicht weiterhin auf niveauevolle Beiträge Wert legen, auf die wir stolz sein können.

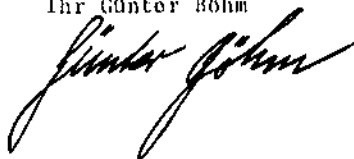
Auch Ihre Vorschläge zur Gestaltung des Journals sind stets willkommen. Dies ist übrigens ein Punkt, den ich bisher am Journal vermißt habe. Wir brauchen unbedingt eine Rückkopplung mit den Lesern. Man muß ja wissen, was die Leute überhaupt wollen, um ihnen das Richtige bieten zu können. Deshalb auch die beigelegte Postkarte! Nützen Sie Ihr Mitspracherecht!

Richten Sie Anfragen in Zukunft direkt an die Redaktion, wir leiten Sie entsprechend weiter. Bisher wurden solche Fragen "im Untergrund" durch die Autoren erledigt. So wurde ein Leser zufriedengestellt, wo doch vielleicht viele an der Antwort interessiert gewesen wären. Überhaupt sollten wir versuchen, gewisse "Computerkontakte" durch das Journal zu knüpfen. Ich habe selbst schon erfahren, wie interessant ein Gespräch oder ein Briefwechsel mit einem Leser sein kann; oder sogar ein Besuch! Abschließend noch eine Bitte: das Journal wird zwar für "ernsthafte NASCOM-Anwender" inseriert, meint aber bestimmt nicht "humorlose" Anwender. Stören Sie sich deshalb bitte nicht daran, wenn die Gestaltung des NASCOM Journals manchmal nicht "streng wissenschaftlich" gemeint ist.

In diesem Sinne wünsche ich viel Freude mit den Beiträgen aus diesem Heft und hoffe auch auf Ihre Mitarbeit zum Gelingen UNSERER nächsten Ausgaben.

P.S. Glauben Sie bloß nicht, ich hätte vor, immer ein so langes Editorial zu schreiben. Aber als Einstieg mußte ich Ihnen ja meine Vorstellungen klarmachen.

Ihr Günter Böhm



## Impressum

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

RESSORTS:

NASSYS Günter Kreidl  
[redacted]  
[redacted] Straelen  
Tel. [redacted]

BASIC Wolfgang Mayer-Gürr  
[redacted]  
[redacted] Recklinghausen  
Tel. [redacted]

HARDWARE Josef Zeller  
[redacted]  
[redacted] Bayreuth  
Tel. [redacted]

T2/T4 Günter Böhm  
[redacted]  
[redacted] Karlsruhe  
Tel. [redacted]

VERLAG

Verlag NASCOM Journal, c/o MK -Systemtechnik, Pater-Mayer-Str.6, 6728 Germersheim  
Tel.07274/2756, Telex 453 500 mks d.

Vertrieb

Direktvertrieb durch den Verlag.

Erscheinungsweise

Monatlich

Bezugspreis

Im Inland 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 M K - Systemtechnik.

(beigefügte Bestellkarte)

Bankverbindungen

Alle Zahlungen für das NASCOM-JOURNAL unter Angabe der Rechnungsnummer nur (!!) an das folgende Konto:

Fa. Michael Klein  
Sonderkonto  
299 26 - 674 beim Postscheckamt Ludwigshafen.

Zahlungen

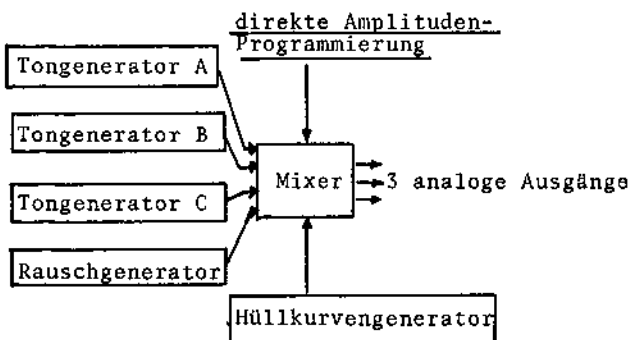
Nach Eingang Ihrer Bestellung erhalten Sie von uns die ausstehenden Hefte bis zur aktuellen Ausgabe sowie eine Rechnung. Bitte, zahlen Sie dann den Rechnungsbetrag auf unser Sonderkonto (s.o.) ein. Bitte keine Vorauszahlungen!!

# SOUNDGENERATOR

Im Januar 1979 stellte die Zeitschrift Elektor zum ersten Mal den programmierbaren Soundgenerator AY-3-8910 vor. Die Hersteller gaben damals über eine Londoner Telefonnummer sogar Hörproben dieses Chips, welche mich sehr beeindruckten. Seither versuchte ich überall, dieses Wunderding zu bekommen. Endlich ist es nun auch in Deutschland erhältlich, und die Zeiten des "dünnen Piepsens" sind nun auch für den NASCOM vorbei; denn der Generator läßt sich leicht über das PIO anschließen.

Hier zunächst eine kurze Charakteristik des Generators.

Der AY-3-8910 enthält 3 (!) Tongeneratoren, (die allerdings nur Rechteckschwingungen erzeugen), einen Rauschgenerator und einen Hüllkurvengenerator; also eine Menge Hardware, mit der man schon komplexere Klänge erzeugen kann. Die Ton- und Rauschgeneratoren lassen sich direkt in der Lautstärke variieren oder aber (zusammen oder getrennt) in Abhängigkeit des Hüllkurvengenerators, der 8 verschiedene Hüllkurven produzieren kann, die jeweils in ihrer Zeitdauer programmierbar sind.



Sämtliche Informationen, die Tonhöhen, Rauschspektrum oder Hüllkurvenform erzeugen, werden in 14 Register des AY-3-8910 eingespeichert und lassen den Generator so lange arbeiten, bis neue Informationen eintreffen, d.h. die CPU kann sich anderen Aufgaben widmen, während der Generator fröhlich vor sich hin summt.

Wenden wir uns zunächst den einzelnen Registern zu. Die Tonhöhen der Tongeneratoren werden durch jeweils 2 Register bestimmt: z.B. R1 Grobstimmung (Coarse Tune = CT) und R0 Feinstimmung (Fine Tune = FT).

Die absolute Tonhöhe hängt allerdings mit dem

Taktsignal zusammen, mit dem der Generator betrieben wird. Bei 1 MHz kann man über größere Bereiche die CT Information auf 0 lassen, dabei wird aber ein Frequenzfehler relativ groß, der auf der internen Teilung der Daten im Generator beruht, die natürlich keine Kommastellen berücksichtigt. Empfehlenswert ist deshalb der Betrieb mit dem 2 MHz Systemtakt, wobei der Frequenzfehler so gering bleibt, daß keine wirklich störenden Schwebungen auftreten. (Mehr als 2 MHz verkraftet der Generator allerdings nicht)

Wenn die Formel zur Berechnung der Registerdaten zu einer gewünschten Frequenz interessiert, der kann sie dem folgenden Basic-Programm entnehmen, das nach Eingabe der gewünschten Oktave (C1 = 1; C = 2; c = 3; c' = 4; c'' = 5 etc.) sämtliche Frequenzen der zugehörigen Halbtöne mit den entsprechenden Registerdaten in dez. und hex angibt.

Als Beispiel: input "Oktave?" 4 drückt die Töne c' bis h' auf dem Schirm aus, wobei man aus der Tabelle entnehmen kann, daß die Daten 1 dez. (= 1 hex) in Register 1 und 28 dez. (= 1 C hex) in Register 0 dem Kammerton a' auf Kanal A erzeugen.

```

10 input 'oktave';o :ifo=0 then end
15 restore
20 print 'ton   frequenz  reg1 dez reg0 reg1 hex
   reg0
21 z=0
22 readt0:readf:z=z+1
23 f=f0(2g(o-1))
30 tp=2000000/(16*f)
40 ct=int(tp/256)
50 ft=int((tp-ct*256)+0.5)
60 fo=2000000/(16*(ct*256+ft))
90 fh=int(ft/16)
100 fl=ft-fh*16
110 ch=ct
120 if fh 9 then fh=fh+7
130 if fl 9 then fl=fl+7
140 if ch 9 then ch=ch+7
150 b0=chr0(ch+48)
160 c0=chr0(fh+48)
170 d0=chr0(fl+48)
180 printt0;tab(3);o;tab(5);f;tab(15);ct;tab(24)
   );ft;tab(31);b0;tab(40);
185 printc0;d0

```

```

190 ifz=12then10
191 goto22
200 data'c',32.703,'cis',34.648,'d',36.708,'dis
',38.891,'e',41.203,'f'
210 data 43.654,'fis',46.249,'g',48.999,'gis',5
1.913,'a',55,'b',58.270
220 data'h',61.735

```

Die gebräuchlichsten Töne werden wohl c bis h' sein, deshalb drucke ich hier gleich eine Tabelle für die entsprechenden Töne ab.

ton	frequenz	reg1 dez	reg0	reg1 hex	reg0	
c	3	130.812	3	188	3	bc
cis	3	138.592	3	134	3	86
d	3	146.832	3	83	3	53
dis	3	155.564	3	36	3	24
e	3	164.812	2	246	2	f6
f	3	174.616	2	204	2	cc
fis	3	184.996	2	164	2	a4
g	3	195.996	2	126	2	7e
gis	3	207.652	2	90	2	5a
a	3	220	2	56	2	38
b	3	233.08	2	24	2	18
h	3	246.94	1	250	1	fa
c	4	261.624	1	222	1	de
cis	4	277.184	1	195	1	c3
d	4	293.664	1	170	1	aa
dis	4	311.128	1	146	1	92
e	4	329.624	1	123	1	7b
f	4	349.232	1	102	1	66
fis	4	369.992	1	82	1	52
g	4	391.992	1	63	1	3f
gis	4	415.304	1	45	1	2d
a	4	440	1	28	1	1c
b	4	466.16	1	12	1	0c
h	4	493.88	0	253	0	fd
c	5	523.248	0	239	0	ef
cis	5	554.368	0	225	0	e1
d	5	587.328	0	213	0	d5
dis	5	622.256	0	201	0	c9
e	5	659.248	0	190	0	be
f	5	698.464	0	179	0	b3
fis	5	739.984	0	169	0	a9
g	5	783.984	0	159	0	9f
gis	5	830.608	0	150	0	96
a	5	880	0	142	0	8e
b	5	932.32	0	134	0	86
h	5	987.76	0	127	0	7f

Wie bekommt man aber nun die Information in die entsprechenden Register?

Dazu sind die Steuerleitungen BC1 und BDIR des AY.... erforderlich. Sie haben folgende Funktion:

BDIR	BC1	
0	0	Chip nicht angesprochen
1	1	Adresse einlesen
1	0	Daten einlesen
(0	1)	(Lesen von Chip, für unsere Zwecke nicht nötig)

Wichtig für die Funktion ist, daß beide Signale wechseln müssen, um die entsprechende Aufgabe zu erfüllen. Eine Änderung von 1 1 auf 1 0 geht nicht, zunächst müssen beide Leitungen auf 0 gehen. Weiterhin darf der Latchimpuls nicht länger als 50 ns dauern, deshalb sind Basic-outputbefehle zu langsam, und man muß hier Maschinenunterprogramme einfügen, wie es im Basicprogramm "Tonerzeugung" geschehen ist.

Man geht bei der Programmierung nun so vor, daß man die entsprechende Registeradresse (über Port A) anlegt und durch einen 00/11/00 Impuls (an Port B) einliest. Dann werden die Daten für die entsprechende Adresse angelegt und mit 00/10/00 eingelesen.

Das folgende Basicprogramm liest die Daten aus Zeile 270 der Reihe nach in die Adressen 0 bis 13. Dabei erzeugt der Soundgenerator einen Dreiklang c-e-g und wiederholt ihn immer wieder mit verklingender Hüllkurve. Experimentieren Sie mit den Tonhöhen, indem Sie die ersten 6 Daten (vielleicht unter Verwendung der Tabelle) verändern.

```

1 rem tonerzeugung mit ay-3-8910
2 rem databeisp.dreiklang,repeat
10 doke 3200,830:doke3202,15827
20 doke 3204,62:doke3206,15827
30 poke3208,201
40 doke4100,3200
50 out62,15:out63,15
60 dimr(14)
70 for i=0to13
80 read r(i)
90 gosub200
100 next
110 goto 300
200 out61,0
210 out60,1
220 poke3201,3 :z=usr(0)
230 out61,0
240 out60,r(i)
250 poke3201,2 :z=usr(0)
260 return
270 data222,1,123,1,63,1,0,248,16,16,16,0,32,8
300 end

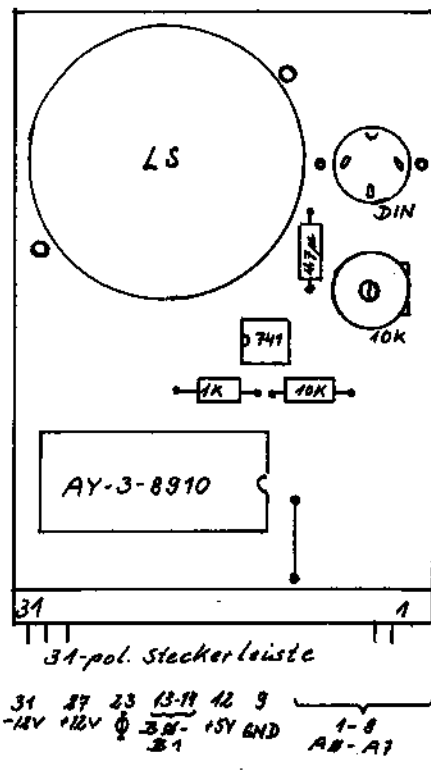
```

Das gleiche Programm ist im folgenden als reines Maschinenprogramm abgedruckt. (Die Unterprogramme outa bzw. outd mußten im Basicprogramm aus oben erwähnten Gründen "gedoked" werden.) Wenn Sie hier mit Tonhöhen experimentieren wollen, müssen Sie die Adressen OCB6 bis OCB6 ( in absteigender Reihenfolge)verändern.

```

0001 :testprogramm fuer ay-3-8910
0002 :dreiklang c-e-g
0010 org Bc80
0c80 3e0f 0020 st1 ld a,15:ports output
0c82 d33e 0030 out {62},a
0c84 d33f 0040 out {63},a
0c86 21ae0c 0050 ld hl,tab:tabellenanfang
0c89 060e 0060 ld b,14 :registeranzahl+1
0c8b 05 0070 st2 dec b
0c8c 78 0080 ld a,b :register nr.
0c8d cd980c 0090 call outa
0c90 7e 0100 ld a,(hl):daten aus tabelle
0c91 cda30c 0110 call outd
0c94 23 0120 inc hl
0c95 20f4 0130 jr nz st2
0c97 cf 0140 end rst 8
0c98 d33c 0150 outa out (60),a:subr.adr.uebergeben
0c9a 3e03 0160 ld a,3
0c9c d33d 0170 out (61),a
0c9e 3e00 0180 ld a,0
0ca0 d33d 0190 out (61),a
0ca2 c9 0200 ret
0ca3 d33c 0210 outd out (60),a:daten uebergeben
0ca5 3e02 0220 ld a,2
0ca7 d33d 0230 out (61),a
0ca9 3e00 0240 ld a,0
0cab d33d 0250 out (61),a
0cad c9 0260 ret
0cae 08 0270 tab defb 8:huellkurvenform
0caf 64 0280 defb 100:huellkurvenperiode grob
0cb0 00 0290 defb 0 :periode fein
0cb1 10 0300 defb 16 :max.amplitude kanal c
0cb2 10 0310 defb 16 :kanal b
0cb3 10 0320 defb 16 :kanal a
0cb4 f8 0330 defb 248:enable a,b,c
0cb5 00 0340 defb 0 :rauschperiode
0cb6 00 0350 defb 0 :c grob
0cb7 9f 0360 defb 159 :c fein
0cb8 00 0370 defb 0 :b grob
0cb9 be 0380 defb 190 :b fein
0cba 00 0390 defb 0 :a grob
0cbb ef 0400 defb 239 :a fein

```

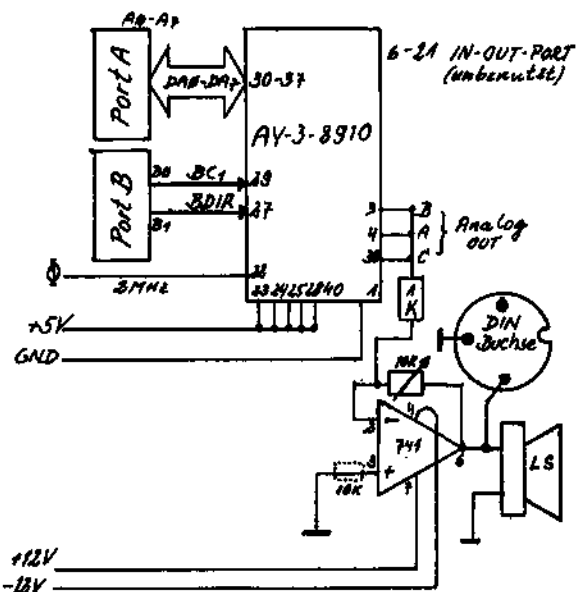


Um die obigen Programme ausprobieren zu können, muß natürlich zuerst die Hardware vorhanden sein. Ich habe sie hier ans Ende gestellt, weil es sich beim Nachbau wirklich um einen Klacks handelt. Weiter unten finden Sie eine Schaltung, die den PSG (Programmable Sound Generator) an das PIO des Nascom anschließt. Falls die Nachfrage besteht, könnten wir Ihnen eine fertige Platine zuschicken. Schreiben Sie uns diesbezüglich.

Die Schaltung ist so konzipiert, daß die Töne sowohl über einen Lautsprecher als auch über eine Verstärkeranlage abgenommen werden können. Das Trimpoti sollte auf minimale Verzerrung eingestellt werden.

Und nun wünsche ich viel Spaß beim Bau und Test des PSG-Interfaces. (Es hat übrigens die gleiche "Busbelegung" wie der AD/DA-Wandler aus Heft ...?).

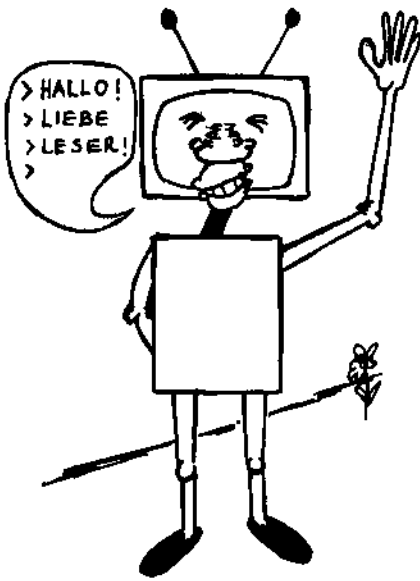
Im nächsten Heft möchte ich dann über die Programmierung des Hüllkurvengenerators, des Rauschens und des Enable Registers berichten. (Hierbei bin ich selbst noch am Experimentieren, um brauchbare d.h. in der Praxis verwertbare Ergebnisse zu erhalten.



**Unbedingt beachten:** Bei Benutzung des normalen PIO auf der Grundplatine müssen Sie die Portadressen der obigen Programme ändern.

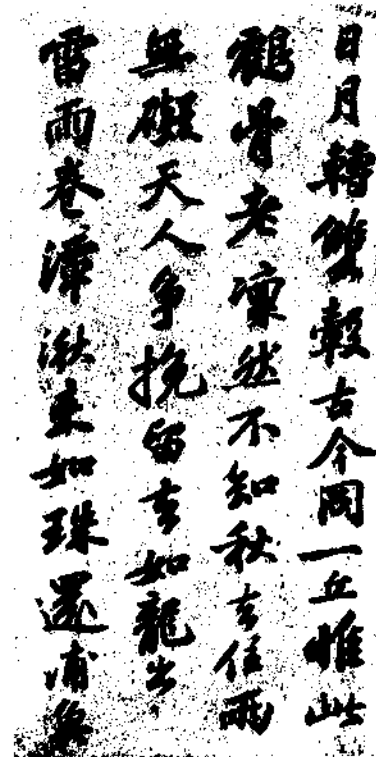
- 60 ≙ 04
- 61 ≙ 05
- 62 ≙ 06
- 63 ≙ 07

*G. Böhm*



Hallo, liebe Leser,  
 Sie werden sich nun langsam an mein Gesicht gewöhnen müssen, denn ich werde von nun an häufiger unvermutet auf den Seiten des NASCOM Journals auftauchen und Sie durch meine dummen Sprüche frustrieren. Wir Computer haben eben unsere eigene Art von Humor, die unsere Besitzer leider nicht immer verstehen. Oder warum glauben Sie, daß Ihr Computer manchmal unerklärlicherweise Speicherstellen verändert, bei einem logischen Programm plötzlich unverständliche Sprünge ausführt oder aber manchmal ein Programm einfach nicht laden will? Haben Sie dabei noch nie das versteckte Lachen aus dem Interface an SK1 gehört? Oder vielleicht das Schmunzeln der Zeile 16, die sich vor Lachen fast scrollen möchte, wenn Sie verzweifelt auf das Keyboard hämmern?  
 Als linke Hand vom Chefproduktour habe ich diesen auch schon oft zur Verzweiflung gebracht; der sieht die Coputerei eben viel zu eng. Hoffentlich schon Sie das alles etwas lockerer.  
 In diesem Sinne verbleibe ich mit freundl. Grüßen

Ihr NASCOMPL



Endlich ist es uns gelungen, ein echtes nationalchinesisches HEXDUMP-Listing zu ergattern. Wir haben es getestet und mußten leider feststellen, daß es weder mit NASBUG t2 oder t4 läuft, noch mit NASSYS. Wahrscheinlich muß man hierfür einen speziellen kleinen Monitor entwerfen, um das Ding zum Laufen zu bringen. Dies wäre eine lohnende Aufgabe für diejenigen Leser des NASCOM Journals, die sich gerne an besonders knifflige Aufgaben heranmachen.

Probieren Sie und schreiben Sie uns, wenn Sie Erfolg hatten. Wir werden selbstverständlich jede mögliche Lösung veröffentlichen, denn die Leserschaft ist sicher brennend daran interessiert, endlich ein original fernöstliches Programm einlesen zu können.

Hier muß ich nun abschließen, denn die Post aus Rußland ist gerade angekommen und wartet auf Bearbeitung.

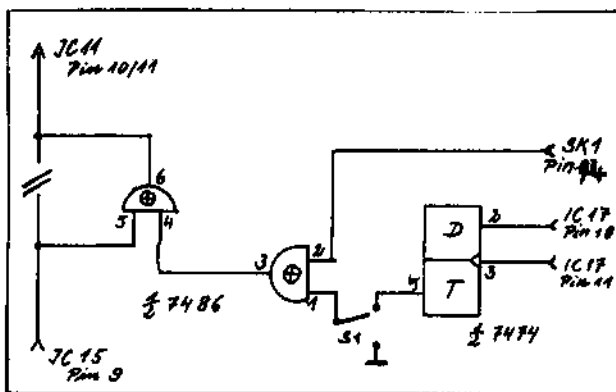
Mit gedrückten Daumen

Ihr NASCOM Journal Auslandskorrespondent

# Inversionsprobleme

Nach Einbau der Zusatzschaltung für die Reversdarstellung (Heft 4, 80 S. 8-9) fiel mir ein unangenehmer Effekt auf: Beim Arbeiten mit dem Assembler (Zeap 1.0) stehen, z.B. bei der Eingabe mit dem I-Kommando, hinter den eingegebenen Zeichen lauter "reverse Blanks". Dies ist wegen der starken Kontrastunterschiede ziemlich hinderlich. Der Mangel läßt sich "beheben", indem man die Schaltung mit einem zusätzlichen Schalter versieht.

Man legt nun einfach beim Arbeiten im Assembler den Schalter um, da hier die Reversdarstellung nicht benötigt wird.



HARALD MOLLE DF78Q  
 SLAUSTEIN  
 Tel. ...

# BUGS BUGS BUGS

Und wieder haben wir einige dieser unangenehmen Wanzen erwischt.

Im Programm "Reaktionszeitmesser" aus Heft 2/81 Seite 6/7 sind folgende Fehler zu berichtigen:

Startadresse nicht ØF18 sondern ØE18  
 In Adr. ØF82 steht "63". Es muß "62" heißen.

Im "Sortierprogramm" aus Heft 5/80 sind folgende Befehle zu ändern:

- ØD28 falsch DØ richtig DD
- ØD2E falsch 3Ø richtig 3D
- ØD5A falsch 3Ø richtig 3D

Wird hoffentlich nicht wieder vorkommen!

# DEZ HEX UMWANDLUNG BASIC

BASIC-Programm:

```
1Ø INPUT
2Ø B= -URS(A): PRINT B
```

Maschinenprogramm (NAS-SYS):

```
CD 8B E9 Call DEINT
EB EX DE,HL
DF 66 Call TBCD 3
C9 Ret
```

Wie arbeitet das Programm?

A ist eine Integer-Dezimalzahl, die das BASIC-Programm anfordert. Das Maschinenprogramm, dessen Startadresse in USRLOC (1ØØ4H) eingetragen werden muß, erhält vom Unterprogramm DEINT die Zahl A in HEX-Form. Unterprogramm TBCD3 zeigt den HEX-Wert an.

HEX-DEZ-Umrechnung

Dieses BASIC-Programm ist noch kürzer:

```
1Ø PRINT USR(A)
```

A ist eine (beliebige) Dummy-Variable.

Maschinenprogramm NAS-SYS):

```
DF 63 Call INLIN (Zeile öffnen)
DF 79 Call RLIN (wenn NL, Zeile lesen und auswerten)
2A ØC ØC LD HL, (ARG1)
DF 66 Call TBCD3 (HEX-Zahl anzeigen)
7C LD A, H
45 LD B, L (und in A, B laden)
CD F2 FØ Call #F FØF2
C9 Ret (übergibt A, B an BASIC)
```

Nach "RUN" erscheint in der nächsten Zeile ein blinkender Cursor; jetzt wird die HEX-Zahl eingegeben und die Entertaste betätigt.

Die nächste Zeile bringt dann zuerst die HEX-, dann die entsprechende Dezimalzahl.

Auch hier muß natürlich USRLOC entsprechend der Startadresse geändert werden!  
 U. Wurditsch



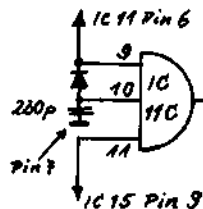
## NASCOM »barock«

Vielleicht haben Sie schon das Programm "NASCOM spielt Melodien" aus dem Journal 6,7/1980 S. 4 eingetippt und waren über die Geschwindigkeit der Melodie erstaunt? Kein Wunder, das Programm war für 1 MHz Takt konzipiert! Obwohl die Spielweise unserem hektischen Zeitalter entspricht, haben sich einige Leser daran gestört. Herr Peter Block aus Neckargemünd hat das Problem einfach gelöst, indem er die Tonperioden durch eine zusätzliche Zeitschleife verlängerte. Auf diese Weise muß in der Tonhöhen- bzw. Tonlängentabelle nichts geändert werden, und die Melodie erklingt dennoch gemütlich barock 1 Oktave tiefer und halb so schnell. Zudem ist das Programm nun voll verschiebbar; es existiert kein absoluter Sprung mehr. Zum Abspielen anderer Melodien als in Heft 6,7 beschrieben, sind nur die unterstrichenen Codes zu verändern (Tonhöhen-tabelle, Tonlängen, Tonanzahl) Hinweise dazu in Heft 6,7/80 Seite 18.

0c50	0010	org	Bc50
0c76	0020	th	equ Bc76
0d00	0030	tl	equ Bd00
0080	0040	tz	equ B80
0c50	<u>21760c</u>	0050	start ld hl,th
0c53	<u>dd21000d</u>	0060	ld ix,tl
0c57	<u>0680</u>	0070	ld b,tz
0c59	<u>dd5e00</u>	0080	laenge ld e,{ix}
0c5c	56	0090	ld d,{hl}
0c5d	3e04	0100	ld a,4
0c5f	d300	0110	hoehe out {0},a
0c61	15	0120	dec d
0c62	20fb	0130	jr nz hoehe
0c64	56	0140	ld d,{hl}
0c65	3e00	0150	pause ld a,0
0c67	d300	0160	out {0},a
0c69	15	0170	dec d
0c6a	20fb	0180	jr nz2pause
0c6c	1d	0190	dec e
0c6d	20ed	0200	jr nz3hoehe
0c6f	23	0210	inc hl
0c70	dd23	0220	inc ix
0c72	10e5	0230	djnz laenge
0c74	18da	0240	jr start

G.B.

## Noch ein »Streifenkiller«

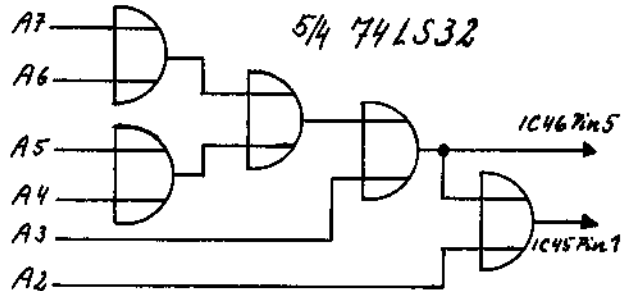


Si-Universal-  
diode  
Kondensatorgröße ausprobieren  
Beinchen 10 hochbiegen,  
Teile direkt an IC löten.

### THEMA "ERWEITERUNG"

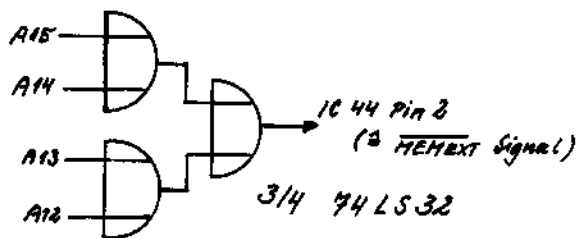
#### ERZUGUNG DES $\overline{\text{IOEXT}}$ SIGNALS UND ÄNDERUNG SEINER VERSCHALTUNG

IC 46 Pin 2 direkt an A2  
IC 46 Pin 5 und IC 45 Pin 1 an folgende  
Schaltung:



Damit ist das PIO auf die Adressen 4-7 festgelegt, auch die anderen Ports sind nicht mehr in ihren Adressen veränderbar.

Mit den übrigen Gattern kann man das MEMEXT-Signal erzeugen. (Dann funktioniert der Monitor auch ohne RAM-Board.)



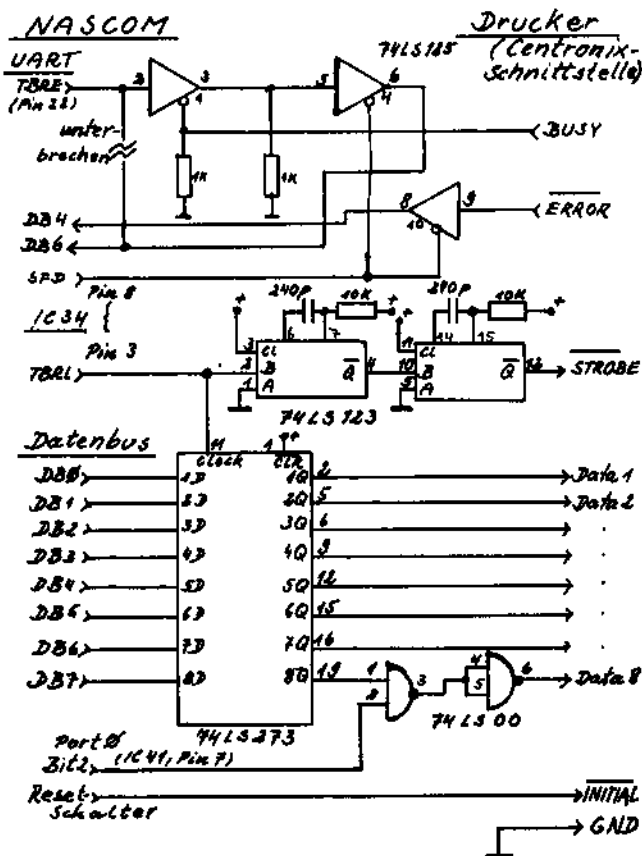
Michael Bach

Stegen

# Parallelinterface für Drucker

Viele billige Drucker, z.B. der EPSON MX-80 oder der SEIKO GP-80, haben standardmäßig eine Centronix-Schnittstelle und bieten eine Nascom-kompatible serielle Schnittstelle nur gegen Aufpreis (250 Mark). Wer das bezahlt, ist selbst schuld. Man kann natürlich aus beiden Ports der PIO eine Parallelschnittstelle softwaremäßig aufbauen. Dann sind jedoch die leistungsfähigen X-Befehle von T4 oder NAS-SYS nicht ausgenutzt, und man muß die ganze Software mit neuen Druckprogrammen versehen. Ich habe für den Anschluß meines Seiko-Druckers eine andere Lösung gewählt: eine Centronix-Schnittstelle wurde aus 4 TTL-IC's aufgebaut und parallel zu dem UART geschaltet.

Alle Daten, die an den UART ausgegeben werden, gelangen nun auch zum Drucker. Die Baudrate kann beliebig hoch eingestellt werden, da über das TBRE-Signal nun auch die BUSY-Meldung des Druckers abgefragt wird. Über Port 0, Bit 2 kann das höchstwertige Datenbit gesperrt werden; dies ist für normalen Betrieb erforderlich, da der X-Befehl Bit 7 als Paritätsbit ausgibt. Will man Bit 7 in speziellen Druckerprogrammen mitbenutzen, muß vorher also Bit 2 in Port 0 auf 1 gesetzt werden. In Port 2, Bit 4 kann das Error-Signal des Druckers abgefragt werden.



## 2716 auf RAM-Karte

Nachdem die 2716'er technisch gegenüber den 2708'ern sowieso schon alle Trümpfe auf ihrer Seite hatten (doppelte Kapazität, geringerer Stromverbrauch, nur 5 V, einfacher zu programmieren), sind sie mittlerweile auch billiger geworden. Dies war für mich ein Anlass, um die 4 Sockel auf der 32k RAM-Karte zur Aufnahme von 5 V-2716'ern vorzubereiten; so finden 8kEeprom auf der Karte Platz. Eine spätere Erweiterung auf 16k in 2732'ern ist leicht möglich. Die notwendigen Änderungen sind in 10 Minuten gemacht. Man unterbreche mit einem scharfen Instrument die Leiterbahn von IC 20, Pin 13 zu IC 24, Pin 2; außerdem die Bahn von +12 V zu Pin 19

der Eprom-Sockelreihe und von -5V zu Pin 21. Die erste Unterbrechung wird am besten an der Platinenunterseite über IC 20 gemacht, die beiden anderen auf der Oberseite. Von IC 20, Pin 13 wird nun ein langer Draht zu Pin 19 eines Eprom-Sockels geführt. Pin 21 der Sockelreihe wird an +5 V (Pin 24 an einem Sockel) gelegt. Einen dritten Draht löte man von Signal A12 (IC 25, Pin 7) zu Pin 2 von IC 24; damit ist die Änderung fertig. Wenn man die Kontaktdurchbrüche der Platine ausnützt und einlötbare DIL-Einzelkontakte verwendet, kann man die Änderung durch Umstecken von 3 Drähten wieder rückgängig machen.

Nun kann der 8k-Bereich der Eproms ausgewählt und der Kontakt P5 mit den beiden entsprechenden Decode-Kontakten verbunden werden; z.B. für den Bereich E000-F7FF mit den Kontakten 11 und 12. Die Sockel würden dann folgendermaßen adressiert: Sockel 27: E000-E7FF; Sockel 28: F000-F7FF; Sockel 29: E800-EFFF; Sockel 30: F800-FFFF.

Leider ist der Eprom-programmierer von Bernd Ploß für 2716'er in der 5 V-Version nicht ausgelegt und muß modifiziert werden; auch ist ein neues Programmierprogramm erforderlich.

Darum biete ich allen NASCOM-Besitzern folgenden Service an: Ich programmiere 2716'er mit Ihrer Software. Schicken Sie mir das Programm auf Kassette (im NASCOM 1-Verfahren beschrieben) sowie einen an Sie adressierten und frankierten Rückumschlag. Da ich kein uneigennütziger Mensch bin, muß ich leider noch einen kleinen Unkostenbeitrag verlangen. Schicken Sie mir ein funktionsfähiges Eprom mit, kostet Sie das Ganze 5 Mark (10 Mark, wenn das Eprom beschrieben ist und erst noch gelöscht werden muß). Ansonsten muß ich 30 Mark in Rechnung stellen. Für zusätzliche 5 Mark bekommen Sie noch ein Druckerlisting Ihres Programms. Legen Sie das Geld in Scheinen bei oder überweisen Sie es auf mein Konto (unten angegeben).



## In Darmstadt nichts Neues... dig. Cassettenrecorder

Einige Leute haben bei mir angefragt, wo denn das versprochene Bedienprogramm für den MDCR bleibe. Es läuft bereits schon längere Zeit, jedoch in einer Form, die es für andere NASCOM-Besitzer uninteressant macht (Timing über CTC, diverse Hardware- und Monitoränderungen). Es so umzuschreiben, daß es auf jedem NASCOM mit dem im Heft 3/81 beschriebenen Interface läuft, erfordert noch einige Stunden Arbeit. Nur Geduld: ich werde mich daranmachen, sobald ich Zeit und Lust dazu habe. Wer noch nähere Informationen über den Betrieb des MDCR braucht, möge mich in Darmstadt anrufen (am besten um 17<sup>00</sup>).

-Nachtrag zum letzten Artikel: Die linke Seite des Interface-Schaltplans ist in einigen Heften teilweise unleserlich abgedruckt worden. Die Signale noch einmal von oben nach unten: 12 V; Port A STB; 12 kHz (verbunden mit Port A STB; Port B Bit 5; Bit 2; Bit 4; Bit 6; Bit 7; Bit 0; Bit 1; Bit 3; Port 0 Bit 7(S7). Die mit der Zahl 2 gekennzeichneten NAND's müssen Open-collector-Gatter des Typs 74LS03 oder 74LS26 sein.

Ich habe im letzten Artikel behauptet, der Datenbus des NASCOM 1 sei überbelastet. Dies muß ich jetzt einschränken, da bei meinem Nascom mittlerweile am gleichen Datenbus zusätzlich noch 2 Latches, 1 weitere PIO, 1 CTC und 2 Eproms hängen und alles auch bei 4 Mhz noch einwandfrei läuft. Offenbar waren die anfänglichen Schwierigkeiten beim Ansteuern meiner PIO auf andere Ursachen zurückzuführen.

Johannes Christian Lotter

Darmstadt

Tel.

Kto.Nr. , PSchA Frankfurt/Main

# RELOCATOR

Hochwertige Assembler ermöglichen die Erzeugung eines relocalisierbaren Objektcodes, der mit einem zugehörigen Lader in jeden beliebigen Speicherbereich lauffähig geladen werden kann. In seinem Buch MIKRO-COMPUTER HARD- UND SOFTWAREPRAXIS beschreibt R.D.Klein einen Lader, der sowohl für das Intel-Hex-Format als auch für das relocalisierbare TDL-Format geeignet ist. Dieses Programm ist aber für ein CP/M-System ausgelegt und zudem nicht ganz fehlerfrei. Das hier beschriebene Programmpaket umfaßt einen Lader und einen Relocator, mit denen man relocalisierbaren Code erzeugen und von Hand oder Cassette laden kann. Es wird ein modifiziertes TDL-Format verwendet, das dem T-Befehl des Monitors angepaßt ist. Der Lader kann jedoch auch Standard-TDL verarbeiten, dagegen kein Intel-Hex-Format.

## Das TDL-Format

Das relocalisierbare TDL-Format ist im Grunde dem Format des TABULATE-Kommandos des Monitors sehr ähnlich; es enthält ebenso wie dieses die Startadresse einer Zeile, Datenbytes und eine Prüfsumme in hexadezimaler Darstellung, darüber hinaus die (variable) Anzahl der Datenbytes, die Relokalisierungsinformation und Formatierungsanweisungen. Die hier verwendete Modifikation des TDL-Formats beschränkt die sonst variable Anzahl der Datenbytes auf acht wie beim T-Befehl. Damit ist aber auch eine Veränderung der Ladelogik verbunden, denn bei einer fixen Anzahl von Daten wird eine Übertragungsbildung der Relokalisierungsinformation in die nächste Zeile nötig, was bei einer variablen Datenzahl vermieden werden kann. Dies ist auch der Grund, warum das von Klein beschriebene Ladeprogramm diese modifizierte Form nicht verarbeiten kann. Eine Zeile des modifizierten TDL-Formats sieht nun folgendermaßen aus:

Zeilenbeginn: CRLF (0D) ";" (3 B)  
09 (das Byte, das die Relokalisierungsinformation enthält, wird mitgezählt)

Ladeadresse der Zeile: z.B. 0000 (Der Programm-anfang wird durch

den Relokator stets auf 0000 gelegt)

Kennzeichnung als  
relocalisierbares  
Format:

01 (ist im modifizierten Format eigentlich überflüssig, wurde aber aus Kompatibilitätsgründen beibehalten)

Relokalisierungs-  
information:

z.B. 08 (Dieser Wert gibt an, welches der folgenden Datenbytes relocalisiert werden soll. Der Wert wird bitweise ausgewertet, so bedeutet etwa 08: das fünfte und sechste Datenbyte werden als Adresse aufgefaßt und entsprechend relocalisiert. Dabei entspricht Bit 7 dem ersten Byte etc.)  
8 Bytes wie beim T-Befehl, jedoch ohne Zwischenräume.  
Zweierkomplement der Summe über alle Bytes außer CRLF und ";".

Datenbytes:

Prüfsumme:

Als Beispiel kann das Relokator-Programm dienen, das hier im relocalisierbaren Format wiedergegeben wird.

## Der LADER

Der Lader wird hier im Hex-Dump angegeben und belegt den Speicherbereich von 0C80 bis 0DEF. Er wird auf der Anfangsadresse gestartet und meldet sich mit: INP.B/REL, sowie einem blinkenden Cursor in der nächsten Zeile. Es müssen nun 2 Parameter eingegeben werden. B=Bias (Verschiebung) und REL=Relokalisierungsadresse. REL bestimmt die Anfangsadresse des geladenen Programms. Durch B wird

eine zusätzliche Verschiebung des Programms im Speicher erreicht, ohne daß die Relokalisierung davon beeinflusst wird. Dies ist z.B. sinnvoll, wenn das Programm noch nicht in den Bereich geladen werden kann, in dem es später laufen soll, etwa in den Bereich, den der Lader selbst belegt. Wird keine zusätzliche Verschiebung gewünscht, muß als erster Wert 0000 eingegeben werden. Die Eingabe der beiden Werte wird durch NEW LINE abgeschlossen (INLINE-Routine). Es erscheint nun eine weitere Abfrage: C/M?

C= Einlegen von Cassette, M= Manuelle Eingabe über die Tastatur.

Die Eingabe einer Zeile muß nun mit NEW LINE und ";" beginnen. Jeder Eingabefehler (unzulässiges Zeichen, Prüfsummenfehler, etc.) wird durch eine Error-Meldung angezeigt.

Es folgt die gegenwärtige Ladeadresse und die Abfrage: NZ?

Antwortet der Benutzer mit "N", so kann die Zeile neu (oder auch eine andere Zeile) geladen werden. Jede andere Eingabe führt zum Rücksprung ins Betriebssystem. Ein Programm wird durch Eingabe von 00000000 abgeschlossen. Auch dann erscheint die Abfrage: NZ?

Tritt beim Einladen von einer Cassette ein Lesefehler auf, so folgt ebenfalls die Error-Meldung, die Anzeige der letzten Ladeadresse und sofort der Rücksprung ins Betriebssystem.

## Der Relocator

Wer den Lader in Betrieb genommen hat, kann nun den Relocator an eine beliebige Speicheradresse laden. Er belegt wie der Lader 368 Bytes und wird ebenfalls auf der Anfangsadresse gestartet. Er meldet sich mit: INP.A/E/T und einem blinkenden Cursor in der nächsten Zeile. Es müssen 3 Parameter eingegeben werden: A=Anfangsadresse des Programms, von dem eine relokalisierbare File erzeugt werden soll, E=Endadresse +1, wobei nur Abstände zwischen A und E zugelassen sind, die ein Vielfaches von 8 betragen, T=Textpufferstartadresse. Die Eingabe der Parameter erfolgt mit der INLINE-Routine und wird mit NEW LINE abgeschlossen. Es wird nun die erste Zeile (Adresse + 8 Datenbytes) des Programms im Tabulate-Format angezeigt. In der nächsten Zeile erscheint die Abfrage: RELOC? mit blin-

kendem Cursor. Das Programm wartet nun auf die Eingabe der Relokalisierungsinformation. Jeweils das erste Byte einer zu relokalisierenden Adresse (CALL, JP und 16-Bit-Ladebefehle) muß als Zahl eingegeben werden, z.B. "1" für das erste Byte von links, "2" für das zweite etc. bis 8. Es können natürlich mehrere Zahlen eingegeben werden, doch niemals zwei aufeinander folgende. Dies führt zwar zu keiner Fehlermeldung beim Relocator, zerstört aber die Programmfile, indem sie unübersetzbar wird. Oberhaupt muß darauf aufmerksam gemacht werden, daß keinerlei Schutz gegen falsche Eingaben in das Programm eingebaut ist. Wird ein falsches Zeichen eingegeben (nicht 1 - 8, NEW LINE oder eins der unten beschriebenen Reset-Zeichen), so muß das Programm mit Reset verlassen und neu gestartet werden. Es gibt allerdings die Möglichkeit, falsch eingegebene Zahlenwerte (1 - 8) wieder zu löschen, doch keinesfalls mit BACKSPACE (!). Es muß jeweils ein zu dem Zahlenwert "komplementärer" Wert eingegeben werden und zwar "9", um die Eingabe "1" zu löschen, ":" für "2", ";" für "3" etc. (Man nehme eine ASCII-Tabelle zu Hilfe). Das hört sich aber alles komplizierter an als es ist. Man kann in wenigen Minuten ein mehrere hundert Bytes langes Programm in ein relokalisierbares Format verwandeln.

Ist die Eingabe der Relokalisierungsinformation abgeschlossen oder wird keine Relokalisierung in der angezeigten Zeile benötigt, so wird NEW LINE eingegeben. Es erscheint dann die nächste Zeile im Tabulate-Format, und das geht so weiter bis die Endadresse erreicht ist. Dann erscheint die Abschlußzeile: 0000000000. Im Textspeicher befindet sich nun die relokalisierbare File (die einzelnen Zeilen waren bei jeder Eingabe von NEW LINE auch schon auf dem Bildschirm sichtbar). Auf die Abfrage: W/N mit blinkendem Cursor kann man sich nun entscheiden: W=Write, die File wird auf Cassette geladen, N=NAS-SYS, das Programm springt in den Monitor zurück, wobei die letzte Adresse des Textpuffers angezeigt wird. Man kann nun also auch diesen Wert und die Startadresse des Textpuffers einem Druckerprogramm übergeben und den relokalisierbaren Objektcode ausdrucken lassen.

Mit Hilfe des Relokator-Programms kann man seine eigenen Programme in ein vollständig relokalisierbares Format bringen und in diesem Format abspeichern. Sie können dann in Zukunft an beliebiger Stelle wieder geladen und so vielseitig verwendet werden ohne umständliches Umrechnen von Adressen. Man sollte dies gleich nach dem Entwickeln und Austesten eines Programmes tun, da es im Nachhinein oft schwer zu entscheiden ist, ob es sich bei einem 16-Bit-Ladebefehl um eine Adresse handelt oder nicht. Noch wichtiger aber erscheint es mir, Programme in einem relokalisierbaren Format zu veröffentlichen. Noch einige Worte zum Programmaufbau. Beide Programme sind geradewegs herunterprogrammiert, mit einigen angehängten Unterprogrammen. Dabei werden ständig die Routinen des Betriebssystems benutzt; eine Umsetzung auf T2/T4 ist deshalb wohl recht mühselig. Das Relokator-Programm erzeugt einige Befehle erst während des Programmablaufs und ist deshalb nicht in ROM lauffähig. Beide Programme benutzen die letzten 16 Bytes (allerdings nicht vollständig) als Variablen-speicher. Dieser Bereich wird beim Start der Programme gelöscht.

RELOCATOR im TDL/NAS-SYS-Format

```

;0900000108AF061021600177230D
;090008010010FCEFOC494E502ED2
;0900100100412F452F540D002180
;09001801804D0122780CDF63DF49
;090020010879DF60226001ED5353
;09002801896201ED4364012266C5
;090030011201ED4368012A62018D
;0900380120ED5B6001AFED520601
;0900400100FF110800AF04ED52AC
;090048010830FB78326D0121083A
;09005001100ED5B660119EBDF04
;090058010054EF52454C4F433FA7
;090060010000E00DF7BF7FE0D2C
;09006801002810D630473E01D6F4
;09007001080810FC327700CBD91D
;090078010818E9DD2A6801DF55D1
;09008001003E0DF73E3BF7410E75
;0900880102003E09DF672A66014E
;0900900120ED5B6001AFED527C33
;0900980100DF677DDF673E01DF37
;0900A001046778DF672A66011686
;0900A80120083A6A01FE00202037
;0900B00100CB1038097EDF672343
;0900B801001520F5182ED55E2378
;0900C0010456E5EBED5B6001AFB4
;0900C80110ED52226B01E1D13A65
;0900D001816B01DF673EFF326A1A
;0900D801040115280F3A6C01DF47
;0900E001026723CB10AF326A0163
;0900E801001520C5792F3CDF67EA
;0900F00144226601DD226801DFF2
;0900F801084EDF6A216D0135C2D9
;09010001804E00DF55EFO3B00BC
;09010801000605AFDF6810FBDD04
;0901100140226801DF4EEFOD579A
;09011801002F4E3F00DF7BF7FED2
;09012001085728062A680118217C
;09012801000DF5DF5DD7242A2E
;09013001886401ED5B68017EF7B2
;0901380100237C9220F97D932043
;0901400100F5DF5DDF5FE5DF770B
;0901480100E1DF66DF5BDD7700F9
;0901500108DD23C9215901DF7109
;0901580100C9656F000000000000
;0000000000

```

Hex-Dump des Laders

```

0C80 AF 21 E2 OD 06 OE 77 23
0C88 10 FC EF OC 49 4E 50 2E
0C90 42 2F 52 45 4C OD 00 DF
0C98 63 DF 79 DF 60 22 E2 OD
OCA0 ED 53 E4 OD EF 43 2F 4D
OCA8 3F 00 CF FE 43 F5 CC CD
OCB0 OD F1 18 04 FE 4D 20 F1
OCB8 F7 CF FE OD 20 FB F7 CF
OCGO FE 3B 20 FB F7 OE 00 FD
OCG8 21 E2 OD DD 2A E4 OD ED
OCDO 5B E2 OD DD 19 CD 74 OD
OCD8 B7 CA C0 OD FD 77 05 CD
OCEO 74 OD 57 CD 74 OD 5F DD
OCE8 19 CD 74 OD FE 01 C2 9A
OCFO OD 00 CD 74 OD 47 FD 35
OCF8 05 CA 9A OD 3E 08 FD 77
OD00 06 FD 7E 07 B7 20 3D CB
OD08 10 38 14 CD 74 OD DD 77
OD10 00 DD 23 FD 35 05 28 52
OD18 FD 35 06 28 D5 18 E2 3E
OD20 FF FD 77 07 CD 74 OD FD
OD28 86 02 30 07 F5 3E 01 FD
OD30 77 08 F1 DD 77 00 DD 23
OD38 FD 35 05 28 2D FD 35 06
OD40 28 B0 18 BD CB 10 DA 9A
OD48 OD CD 74 OD FD 86 03 FD
OD50 86 08 DD 77 00 DD 23 AF
OD58 FD 77 08 FD 77 07 FD 35
OD60 05 28 07 FD 35 06 28 8A
OD68 18 97 CD D7 OD 81 C2 9A
OD70 OD C3 B9 OC 21 E6 OD CF
OD78 F7 D7 OD 77 CF F7 D7 08
OD80 ED 6F 7E F5 81 4F F1 C9
OD88 FE 3A 38 02 D6 07 D6 30
OD90 DA 9A OD FE 10 D2 9A OD
OD98 C9 00 DF 6B DD E5 E1 DF
ODAO 66 3A EB OD B7 28 0A DF
ODA8 5F DF 5D AF 32 EB OD DF
ODBO 5B EF OD 4E 5A 3F 00 DF
ODB8 7B FE 4E 20 F2 C3 B9 OC
ODCO 06 03 CD 74 OD B7 C2 9A
ODC8 OD 10 F7 18 D4 3E FF 32
ODDO EB OD DF 5F DF 5D C9 CD
ODD8 74 OD F5 57 79 92 4F F1
ODE0 C9 00 00 00 00 00 00 00
ODE8 00 00 00 00 00 00 00 00

```

Nie heißt das nun eigentlich  
nicht zu  
relozierbar, relocating,  
relocatierbar, relokalisierbar  
Bierbar, relocable,  
relokabel, relokabel,  
verschiebbar, verschieblich  
verschiebel, verschiebsam,  
Verschiebung  
verschiekdel,  
ortungebunden,  
ortlos oder  
vielortig ???  
Am besten einigen  
wir uns auf  
"verschortatibel"!

Günter Kreidl  
Straele  
Tel. \_\_\_\_\_



# JAHRESINHALTSVERZEICHNIS

## A. NASBUG T2

<b>Spiele:</b>	Mondlandung		0/2	
	Tongenerator	U. Krüger	0/4	
	Sprachsynthesizer	U. Krüger	0/4	
	LIFE	P. Bentz	1/4	
	INVASION		2/5	
	Änderung zu INVASION		6/10	
	8 Damen Problem	L. Bayer	3/14	
	Real-Time Clock	R. Böttchers	4/8	
	Reaktionstest	E. v. d. Vaart	5/8	
	Melodienerzeugung	G. Böhm	6/4, 18	
	Telespiel m. Ton	G. Böhm	6/18, 32	
	<b>Utilities:</b>	Speichertest	U. Wurditsch	0/7
		Stichwortsuche	J. Altmann	2/7
		Ladeprogramm.	J. Bezold	2/14
		Scheduler	R. Böttchers	4/6
		PLOTTER Software	P. Bentz	5/2; 6/36
		Mech. Spiel	"	6/11
		Schrittmot.	"	6/12
		Mehr Adr. mit Tabul.	E. v. d. Vaart	5/8
		Sortierprogramm	Th. Schreiner	5/12
Systemtest		H. G. Schäfer	6/4	
Nützl. Unterprogramm		M. Bach	6/20	
Intell. Tabulator		E. v. d. Vaart	6/27	
Disassembler		G. Böhm	6/28, 48	
Schreibmasch. Steuer		U. Krüger	0/6	
Kugelpopftreiber		M. Bach	6/45	
Formatierprogramm		H. Grasl	6/15	

## B. NASBUG T4

## C. NASSYS

<b>Spiele:</b>	PIRANHAS		1/8	
	Software Uhr		2/4	
	NIMM	G. Möse	4/5	
	Hohe Hausnummer	G. Möse	4/5	
	<b>Utilities:</b>	DARTS	G. Brust	6/24
		Interrupts	B. Ploss	4/16
		Sporadische Speicherfehler	M. Klein	0/7
		IMP Drucker		1/2
		Z 80 ASSEMBLER	M. Klein	1/11; 2/12
		NASPEN	H. J. Dietmann	2/2
		Disass.+ZEAP	P. Bentz	3/12
		Hobby Elektr. 80	M. Klein	4/3
		Schreibmasch. Interface		4/13
		Hochauflösende Grafik		4/14
	<b>D. Beschreibungen</b>	Bookshop		4/19
		Umlaute mit NASCOM		5/7
		Neuer Disassembler	U. Krüger	6/5
		10 Byte Z80 Befehle?	"	6/8
		Grafik Erweiterung	W. Hentschel	6/29
		Grafik Zusatzkarte		6/49

## E. BASIC Programm

8 Damen Problem	M.Klein	0/10
Entfernung mit QTH	W.Widmann	3/8
17 + 4	P.Szymanski	3/13
Zahlenraten		4/11
Änderung der Zeilennummer	U.Wurditsch	6/16
Geldspielautomat	H.Auge	6/21
Stringfelder auf Cassette	P.Szymanski	6/33
Kalender		1/7

## F. Hardware

Interface Schreibmaschine	U.Krüger	0/6
Streifenkiller	M.Klein	0/9
Snowplough	K.Poschmann	4/9
Verbessertes Cassetteninterface	H.Gundermann	2/4
Recorder Fernsteuer.mit Optokoppler	G.Kreidl	4/9
Interface Kugelkopfdrucker	M.Bach	6/25
TTY über UART	G.Böhm	6/17
Hell- Dunkelumschaltung		3/10
Invers Darstellung	K.Poschmann	4/8
Simpelgrafik	C.Lotter	5/15
	G.Böhm	6/5
Hochauflösende Grafik	A.Schunck	5/9
Grafik "Brutal"	G.Böhm	6/19
Speicherschutz	A.Schunck	2/4
BasicRom auf normaler Speichererweiterung	R.Böttchers	3/11
Billigst Speichererweiterung	G.Böhm	6/19
2 Betriebssysteme für NASCOM 2	B.Ploss	4/9
Analoger Joystick	U.Krüger	6/6
NMI Taste für NASCOM 1	U.Krüger	6/9
PLOTTER für NASCOM	P.Bentz	3/3
Autoreset	C.Lotter	5/14
TV als Videomonitor	U.v.Mulert	5/6



# NEUES VOM RELOCATOR...

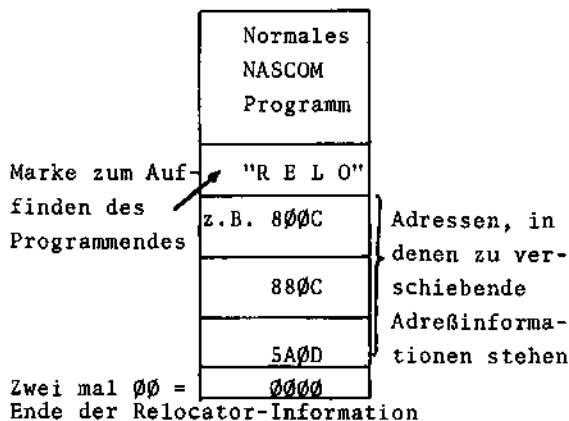
Das weiter oben beschriebene Relocator-Programm eignet sich ausgezeichnet, wenn bereits mehrere Programme im TDL-Format vorliegen. Dies ist für den NASCOM leider nicht der Fall. Außerdem ergeben sich noch einige weitere Nachteile:

- Das Umschreiben alter Programme scheint doch recht aufwendig.
- Eine Zeile im TDL-Format ist bedeutend länger als im NASCOM R- oder L-Format und zudem schwer lesbar, da Zwischenräume fehlen.
- Falsche Eingabe bei Relocator zerstört Programmfile ohne Error-Meldung.
- Komplizierte Verbesserung falscher Zahlenwerte (Berechnung des Komplements mit ASCII-Tabelle)
- Nicht in EPROM speicherbar.
- Augenblicklich nur mit NASSYS verwendbar.

Um obige Nachteile zu vermeiden, habe ich einen kleinen Relocator entwickelt, der sehr einfach anzuwenden ist. Seine Vorteile liegen darin, daß er

- a) jedes alte NASCOM-Programm verarbeitet,
- b) sowohl mit R- als auch L-Format arbeitet,
- c) relocalisierbare Programme auf Cassette speichern kann,
- d) oder aber (und das ist sein größter Vorteil) jedes fertige Programm durch einfaches Anhängen der Relocator-Informationen auf das relocalisierbare Format bringt.

## Format:



## Arbeitsweise:

Sie laden ein Programm wie gewohnt, oder Sie tippen es manuell ein, wobei Sie es schon in den Bereich legen sollten, auf dem es bei Ihnen laufen soll. Das geladene Programm muß jedenfalls immer in den gewünschten Bereich kopiert werden.

Nun hängen Sie an das Programm 52h 45h 4Ch und 4Fh an ("RELO"), um den Beginn der Relocate-Information zu kennzeichnen. Daran müssen sich die Adressen anschließen, deren Inhalte zu verschieben sind.

Beispiel: Im Originallisting steht bei 0C80 CD 48 0D

Sie müssen nun 810C ans Programm anhängen, weil dort die zu verschiebende Adresse beginnt.

Dieses Verfahren führen Sie fort, bis alle direkten Sprünge, Calls und Adreßladebefehle angehängt sind. Dann wird durch 00 00 das Ende kenntlich gemacht.

Sie starten den Relocator, der übrigens voll verschieblich und auch in EPROM speicherbar ist, im abgedruckten Beispiel bei C80, wobei Sie als Argumente die Originalstartadresse und die Zieladresse eingeben.

Beispiel: Ein Originalprogramm beginnt bei 0DE0. Es soll bei Ihnen auf 1000 beginnen.

Sie geben ein: EC80 0DE0 1000 NL

Der Relocator errechnet nun aus Start- und Zieladresse das Displacement. (Es kann sowohl nach oben als auch nach unten verschoben werden).

```

0c80          0010          org    8c80
0c0e          0020 arg2    equ    8c0e
0c10          0030 arg3    equ    8c10
0c80 2a0e0c  0040 displ   ld    hl,(arg2)
0c83 ed5b100c 0050          ld    de,(arg3)
0c87 d5      0060          push  de
0c88 dde1    0070          pop   ix
0c8a e5      0080          push  hl
0c8b a7      0090          and   a
0c8c ed52    0100          sbc   hl,de
0c8e 300a    0110          jr    nc,neg
0c90 e1      0120 pos     pop   hl
0c91 0601    0130          ld    b,1
0c93 eb      0140          ex    de,hl
0c94 a7      0150          and   a
0c95 ed52    0160          sbc   hl,de
0c97 eb      0170          ex    de,hl
0c98 1804    0180          jr    reloc
0c9a eb      0190 neg     ex    de,hl
0c9b e1      0200          pop   hl
0c9c 0600    0210          ld    b,0
                0220 reloc
    
```

Dann sucht der Relocator die Information "RELO", rechnet die folgenden Adressen auf die physikalische Lage des Programms um und addiert (oder subtrahiert) das Displacement zu ihren Inhalten.

Nach Auffinden von  $\emptyset\emptyset\emptyset$  meldet der Relocator "Ready" und springt zum Monitor zurück. Das Programm ist nun an gewünschter Stelle lauffähig. Sie können es von dort auch auf Cassette abspeichern (inklusive der Relocator-Informationen). Als Ausgangspunkt für weitere Verschiebung muß aber immer die Anfangsadresse verwendet werden, für die das Programm ursprünglich geschrieben wurde.

```

0c9a      0010      org      #c9a
0c9a dde5      0020      reloc   push   ix
0ca0 e1        0030      pop     hl
0ca1 2b       0040      dec     hl
0ca2 23       0050      such   inc     hl
0ca3 3e52     0060      ld      a,#52
0ca5 be       0070      cp     (hl)
0ca6 20fa     0080      jr     nz,such
0ca8 23       0090      inc     hl
0ca9 3e45     0100      ld      a,#45
0cab be       0110      cp     (hl)
0cac 20f4     0120      jr     nz,such
0cae 23       0130      inc     hl
0caf 3e4c     0140      ld      a,#4c
0cb1 be       0150      cp     (hl)
0cb2 20ee     0160      jr     nz,such
0cb4 23       0170      inc     hl
0cb5 3e4f     0180      ld      a,#4f
0cb7 be       0190      cp     (hl)
0cb8 20e8     0200      jr     nz,such

13be 23       0210      inc     hl
13bf e5       0220      push   hl
13c0 dde1     0230      pop     ix
13c2 c5       0240      gefu   push   bc
13c3 4e       0250      ld      c,(hl)
13c4 23       0260      inc     hl
13c5 46       0270      ld      b,(hl)
13c6 c5       0280      push   bc
13c7 e1       0290      pop     hl
13c8 c1       0300      pop     bc
13c9 3e00     0310      ld      a,0
13cb b0       0320      or     b
13cc 2803     0330      jr     z,sub1
13ce 19       0340      add    hl,de
13cf 1803     0350      jr     aender
13d1 a7       0360      sub1   and    a
13d2 ed52     0370      sbc    hl,de
13d4 e5       0380      aender push   hl
13d5 c5       0390      push   bc
13d6 4e       0400      ld      c,(hl)
13d7 23       0410      inc     hl
13d8 46       0420      ld      b,(hl)
13d9 c5       0430      push   bc
13da e1       0440      pop     hl
13db c1       0450      pop     bc
13dc 3e00     0460      ld      a,0
13de b0       0470      or     b
13df 2803     0480      jr     z,sub2
13e1 19       0420      add    hl,de
13e2 1803     0500      jr     einfue
13e4 a7       0510      sub2   and    a
13e5 ed52     0520      sbc    hl,de
13e7 e5       0530      einfue push   hl
13e8 fde1     0540      pop     iy
13ea e1       0550      pop     hl
13eb c5       0560      push   bc
13ec fde5     0570      push   iy
13ee c1       0580      pop     bc
13ef 71       0590      ld      (hl),c

```

```

13f0 23       0600      inc     hl
13f1 70       0610      ld      (hl),b
13f2 c1       0620      endtes pop   bc
13f3 dd23     0630      inc     ix
13f5 dd23     0640      inc     ix
13f7 3e00     0650      ld      a,0
13f9 ddb0     0660      cp     (ix+0)
13fc 2009     0670      jr     nz,weiter
13fe dd23     0680      inc     ix
1400 ddb0     0690      cp     (ix+0)
1403 2807     0700      jr     z,ende
1405 dd2b     0710      dec     ix
1407 dde5     0720      weiter push  ix
1409 e1       0730      pop     hl
140a 18b6     0740      jr     gefu
140c ef       0750      ende   defb  #ef
140d 52       0760      defm  'ready'
1412 00       0770      defb  0
1413 c35903   0780      jp     #359

```

Dieses Programm läßt sich natürlich nur dann problemlos durchführen, wenn die Ursprungsprogramme durchsichtig sind und nicht zu viele direkte Sprünge enthalten. (Obwohl das Programmformat immer noch kleiner bleibt als bei TDL).

Deshalb ein Aufruf an alle Leser, die Programme veröffentlichen:

1. Vermeiden Sie möglichst direkte Sprünge! (Falls die Distanz für relative Sprünge zu groß wird, kann man Programmtteile einfügen, die die Distanz verlängern. z.B. ....JR MARK 5  
: MARK 5 JR ENDE etc.)
2. Unterstreichen Sie auf jeden Fall alle Adressen, die bei Verschiebung verändert werden müssen. (Wird ja zum größten Teil schon gemacht.)
3. Komfortabelste Lösung: Hängen Sie die Relocator-Information an Ihre Programme an. Für Anwender eines Assemblers: Ans Programm anhängen DEFM "RELO" und dann die Marken +Displacement, die zu verschiebende Adressen enthalten. Normalerweise ist das Displacement "+1" z.B. bei CALL CD Adresse JP C3 Adresse LD HL21 Adresse Nur bei Verwendung der Index-Register wird es "+2" z.B. bei LD IX DD21 Adresse Vergessen Sie nicht, die "Ende"-Information  $\emptyset\emptyset\emptyset$  ! (Assembler: DEFW  $\emptyset$ )

Hier als Abschluß das Hex-Dump des Relocators. (Wie gesagt: klein, voll verschieblich und einfach)

0c80 2a 0e 0c ed 5b 10 0c d5  
 0c88 dd e1 a5 a7 ed 32 30 0a  
 0c90 e1 06 01 eb a7 ed 52 eb  
 0c98 19 04 eb e1 06 00 dd e5  
 0ca0 e1 2b 23 3e 52 be 20 fa  
 0ca8 23 3e 45 be 20 f4 23 3e  
 0cb0 4c be 20 ee 23 3e 4f be  
 0cb8 20 e8 23 e5 dd e1 c5 4e  
 0cc0 23 46 c5 e1 c1 3e 00 b0  
 0cc8 28 03 19 18 03 a7 ed 52  
 0cd0 e5 c5 4e 23 46 c5 e1 c1  
 0cd8 3e 00 b0 28 03 19 18 03  
 0ce0 a7 ed 52 e5 fd e1 e1 c5  
 0ce8 fd e5 c1 71 23 70 c1 dd  
 0cf0 23 dd 23 3e 00 dd be 00  
 0cf8 20 09 dd 23 dd be 00 28  
 0d00 07 dd 2b dd e5 e1 18 b6  
 0d08 ef 52 45 41 44 59 00 c3  
 0d10 59 03

Günter Böhm

## kleinanzeigen

Jeder Abonnent kann kostenlose Kleinanzeigen bis 40 Wörter aufgeben!

### VERKAUFE

NASBUG T4 und NAS-SYS i (interruptfähige Version) jeweils in einem 2716 (5V) für DM 40,-

Johannes C. Lotter, [REDACTED]

Tel. [REDACTED]

### SUCHE

Tauschpartner für Programme auf Disketten

### VERKAUFE

NAS-DIS EPROMS mit Unterlagen 90,-  
 Graphik-ROM 90,-  
 EPROM-BOARD mit 12k Basic 250,-

Wolfgang Mayer-Gürr

Tel. [REDACTED]

### LEERKASSETTEN



Speziell geeignet für Datenaufzeichnung. Hochwertiges BASF-Band. Cassette 5-fach verschraubt. Cassette C10, d.h. 10 Minuten spieldauer, daher besonders geeignet für Mikrorechnerprogramme.

10 Stk 19.80     Jede Kassette mit selbst-  
 20 Stk 36.00     klebendem Aufkleber zum  
 50 Stk 87.50     Beschriften.  
 100 Stk 160.00

Bei: M K - Systemtechnik

## PREISAUSSCHREIBEN

### Gewinnen Sie ein Schachprogramm

Von nun an wollen wir in regelmäßigen Abständen Preisausschreiben für Programme unter verschiedenen Themen starten. Fürs erste Mal haben wir das Thema "Computerspiele" ausgewählt. Das Programm, das von der Jury (Herausgeber und Redaktion) als das beste ausgewählt wird, wird mit einem SCHACHPROGRAMM NACH WUNSCH prämiert! Ein lohnender Preis!

Kramen Sie in Ihrer Schublade nach Spielprogrammen; sicher finden Sie etwas, was für eine Teilnahme geeignet ist.

Das eingesandte Programm wird nach folgenden Bedingungen ausgewählt:

1. Wichtig ist der "Unterhaltungswert".
2. Es soll einfach bedienbar sein.
3. Es soll möglichst wenig Speicherplatz belegen.
4. Es soll möglichst originell sein. (Kein weiteres Nimm oder Mastermind)

Einsendeschluß ist der 31.7.81

Die Entscheidung der Jury ist endgültig.

Einsendungen bitte unter dem Stichwort "Preisausschreiben"

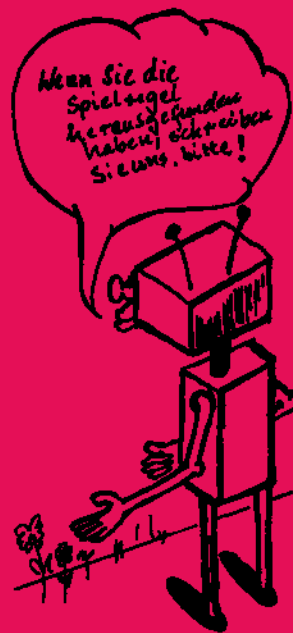
Und nun viel Glück!

# othello

```

Oc50 21 e7 0f 3e 08 cd 7a 04
Oc58 c6 30 32 54 0b 3e 08 cd
Oc60 7a 04 c6 30 32 55 0b c9
Oc68 20 31 32 33 34 35 36 37
Oc70 38 20 31 2e 2e 2e 2e 2e
Oc78 2e 2e 2e 31 32 2e 2e 2e
Oc80 2e 2e 2e 2e 2e 32 33 2e
Oc88 2e ff 2e 2e 2e 2e 2e 33
Oc90 34 2e 3d ff ff 00 2e 2e
Oc98 2e 34 35 2e 2e ff ff ff
Oca0 2e 2e 2e 35 36 2e 2e 3d
Oca8 2e 3d 2e 2e 2e 36 37 2e
Ocb0 2e 2e 2e 2e 2e 2e 2e 37
Ocb8 38 2e 2e 2e 2e 2e 2e 2e
Occ0 2e 38 20 31 32 33 34 35
Occ8 36 37 38 20 4d 6f 76 65
Ocd0 20 4e 6f 20 20 34 20 33
Ocd8 20 63 68 6f 69 63 65 73
Oce0 20 00 20 74 6f 20 6d 6f
Oce8 76 65 89 0c 02 01 00 00
Ocf0 01 09 0a 0b f5 f6 f7 ff
Ocf8 00 58 08 95 00 00 ff 00
Od00 cd c8 0d 21 d4 0c cd 80
Od08 0d 00 00 00 21 e1 0c 7e
Od10 2f 77 cd 45 0e 01 0d 0a
Od18 11 68 0c cd b0 0d 21 d6
Od20 0c 06 02 cd 94 0d 20 1c
Od28 cd 90 0f 21 ff 0c 34 7e
Od30 3d ca 03 0d ef 20 47 11
Od38 6d 65 20 4f 76 65 72 21
Od40 00 c3 88 0f af c3 0d 0f
Od48 00 ef 20 52 6f 77 20 43
Od50 6f 6c 3f 00 cd db 01 21
Od58 54 0b cd 70 0d da 9e 0e
Od60 21 68 0c 06 00 09 cd 80
Od68 0e ca 9e 0e c3 03 0d 00
Od70 cd a0 0d d8 cd 8c 0d cd
Od78 a0 0d d8 81 4f c9 00 00
Od80 3e 39 23 34 be d0 36 30
Od88 2b c3 88 0e 17 4f 17 17
Od90 81 4f c9 00 af b7 c3 90
Od98 0e 00 23 c6 d0 10 f6 c9
Oda0 7e 23 fe 20 28 fa d6 31
Oda8 d8 fe 08 3f 3c c9 00 00
Odb0 cd b2 0e 2a 19 0c c5 1a
Odb8 13 77 23 36 20 23 10 f7
Odc0 0e 2c 09 c1 0d 20 ef c9
Odc8 21 fe 0c 7e 2f 77 32 d0
Odd0 0e 00 21 73 0c 01 2e 08
Odd8 3e 08 71 23 3d 20 fb 23
Ode0 23 10 f5 21 20 30 22 d4
Ode8 0c 22 d6 0c af 32 ff 0c
Odf0 67 00 32 e1 0c 2f 6f 22
Odf8 94 0c 6c 67 22 9e 0c c9
Oe00 af 32 fc 0c 11 f0 0c 1a
Oe08 13 b7 20 05 3a fc 0c b7
Oe10 c9 4f 17 9f 47 e5 cd 20
Oe18 0e cc 30 0e e1 18 e8 00
Oe20 09 3a e1 0c 2f be c0 09
Oe28 be 28 fc 2f be c9 00 00
Oe30 22 fb 0c 3a fd 0c b7 c0
Oe38 3a e1 0c 2f b7 ed 42 be
Oe40 2f 77 28 f7 c9 af 2f 32
Oe48 fd 0c 21 20 30 22 d6 0c
Oe50 21 73 0c 0e 08 06 08 c5
Oe58 01 3d 2e 7e b8 28 04 b9
Oe60 20 10 70 cd 00 0e 28 0a
Oe68 36 3d e5 21 d6 0c cd 80
Oe70 0d e1 23 c1 10 e1 23 23
Oe78 0d 20 da af 32 fd 0c c9
Oe80 3e 3d be ca 00 0e bf c9
Oe88 34 7e fe 21 c0 36 3f c9

```



```

Oe90 cd 8c 0d 7e fe 20 20 02
Oe98 c6 10 81 c3 9a 0d af 1e
Oea0 20 45 72 72 6f 72 21 00
Oea8 06 80 cd 35 00 10 fb c3
Oeb0 15 0d 21 d0 0e cb fe 00
Oeb8 ef 1e 00 21 ca 0b 22 18
Oec0 0c ef 2a 20 4f 54 48 45
Oec8 4c 4c 4f 20 2a 20 43 3d
Oed0 ff 20 20 53 63 6f 72 65
Oed8 3a 20 20 ff 20 20 30 20
Oee0 20 20 20 80 20 20 30 1f
Oee8 00 c5 d5 06 64 1a 13 fe
Oef0 ff 20 06 21 e5 0b cd 80
Oef8 0d fe 00 20 06 21 ed 0b
Of00 cd 80 0d 10 e8 d1 c1 3e
Of08 20 32 4a 0b c9 21 ff 0c
Of10 77 3a fe 0c 4f 3a e1 0c
Of18 b9 c2 49 0d cd 90 0f 21
Of20 dd 0f 14 a0 0f 46 23 e5
Of28 af 32 ec 0c 26 0c 1a fe
Of30 00 20 01 76 13 6f 7e fe
Of38 3d 20 0e af 32 ed 0c d5
Of40 11 f0 0c 1a 13 b7 20 11
Of48 d1 10 e3 e1 3a ec 0c fe
Of50 00 28 d2 2a ea 0c c3 66
Of58 0d c5 4f 17 9f 47 e5 09
Of60 7e e1 c1 fe 00 28 04 fe
Of68 ff 20 d8 e5 21 ed 0c 34
Of70 7e 00 2b be 38 07 77 e1
Of78 22 ea 0c 18 01 e1 18 c3
Of80 00 00 00 00 00 00 00 00
Of88 cd 98 0f c3 00 0d 00 00
Of90 cd 3e 00 fe 20 20 f9 c9
Of98 cd 3e 00 fe 2f 20 f9 c9
Ofa0 73 7a c0 b9 75 8e be a5
Ofa8 87 bb ac 78 98 bc 9b 91
Ofb0 76 a2 77 bd 89 aa 8c a7
Ofb8 8b 96 a8 a9 9d a0 8a 93
Ofc0 8d b4 ab b1 a6 7f 82 88
Ofc8 80 a1 92 9c b2 97 81 b3
Ofd0 bf ba 74 7d 79 84 b6 af
Ofd8 7e b0 b5 83 00 04 08 08
Ofe0 04 08 10 08 04 00 00 00

```

Obiges Programm wurde von Richard Beal erstellt.

Es läuft mit NASBUG T2 und T4 und startet bei ØDØØ. Die NASCOM-Besitzer, die das freie Bit 7 des Charactergenerators für Graphik verwenden, werden allerdings ihre Freude daran haben, denn das Programm verwendet FF, um das ASCII Zeichen 7F darzustellen. Auf meinem Schirm sieht das etwas seltsam aus. Vielleicht können Sie die Unschönheit selbst beseitigen.

Eigentlich sollten Sie zum Spielen von Othello die Spielregeln kennen; aber die habe ich selbst noch nicht herausgefunden. Das Programm hilft Ihnen allerdings, denn es zeigt Ihnen durch "ERROR", wann Sie mogeln, und gibt an, wann Sie am Zug sind.

Das Spiel gegen den Computer wird jeweils durch Drücken der Leertaste fortgesetzt. Wenn Sie gegen einen "menschlichen" Partner spielen wollen, ändern Sie einfach ØF19 von C2 zu C3. Dann sind beide Partner immer abwechselnd am Zug. "/" beginnt neues Spiel.

G.Böhm