

nascocom

**JOURNAL 6/80
7/80**

ZEITSCHRIFT FÜR BENUTZER DES NASCOM 1 ODER NASCOM 2

Herausgegeben von: M K - Systemtechnik, Michael Klein, Waldstraße 20
6728 Germersheim

Heftpreis: 4.- ; Abonnement für 1980 nur 39.-, sonst 48.- pro Jahr
für 12 Ausgaben

Redaktion: c/o M K - Systemtechnik, Waldstr. 20 6728 Germersheim

Liebe Leser,

das erste Jahr des NASCOM-JOURNALS geht nun mit einer Doppelausgabe zu Ende, von der wir hoffen, daß sie Ihren Beifall findet.

Die letzten drei bis vier Monate waren durch eine zunehmend schwieriger werdende Liefersituation bei NASCOM-Produkten gekennzeichnet. Nur die im Inland hergestellten Teile konnten noch problemlos geliefert werden. Nach Übernahme der Firma NASCO ist jedoch zu hoffen, daß die Liefersituation sich wieder verbessern wird. Bei den im Inland hergestellten Produkten hat vor allen Dingen unsere CLD - Floppy sehr gute Verkaufserfolge aufzuweisen. Offenbar scheinen das Konzept und der günstige Preis doch zahlreiche Anwender überzeugt zu haben. Auch die verschiedenen Typen von Graphik-Erweiterungen erfreuen sich großen Interesses.

Zahlreiche NASCOM-Anwender sind in den letzten drei Monaten auf APPLE-, EUROCOM- oder DAI-Systeme umgestiegen, sodaß es möglich war, die zurückgegebenen Gebrauchtsysteme anderen NASCOM-Interessenten anzubieten. Dies hat dazu beigetragen, daß sich die Liefersituation zumindest ein wenig entspannt hat.

Wir hoffen, daß das NASCOM-JOURNAL auch im nächsten Jahr mit gleichem Erfolg weitergeführt werden kann. Einen wesentlichen Beitrag zur Gestaltung des NASCOM-JOURNALS leisten nach wie vor die Leser des JOURNALS selber. Es ist zu hoffen, daß wir auch in Zukunft zahlreiche qualifizierte Beiträge von NASCOM-Anwendern erhalten werden.

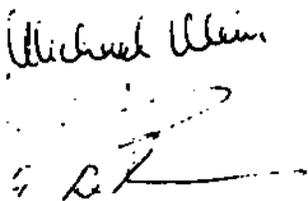
Da sich niemand besser mit den in den Beiträgen beschriebenen Problemen auskennt, als der Verfasser selber, möchten wir Sie bitten, Rückfragen immer direkt an die jeweiligen Autoren zu richten. Die Adressen

und gegebenenfalls auch Telefonnummern sind unter den Artikeln aufgeführt.

Nun noch eine Anmerkung zum Schluß: wir haben den Aufbau des JOURNALS leicht verändert: Die teilweise sehr langen Listings zu den einzelnen Beiträgen haben wir in dieser Ausgabe des JOURNALS auf die hinteren Ränge, sprich in den Anhang verbannt und durch Nummern gekennzeichnet. Für uns birgt diese Lösung vom Umbruch her viele Vorteile, aber wie ist das mit Ihnen? Wie gefällt Ihnen diese Lösung? Wir sind gespannt auf Ihre Reaktion; bitte schreiben Sie uns.

So, nun bleibt uns nur noch eines zu sagen: wir, d.h. der Mitarbeiterstab des NASCOM - JOURNALS wünschen Ihnen allen ein frohes Weihnachtstfest und alles Gute zum Neuen Jahr.

Die Herausgeber



Bezugsbedingungen

1. Das Jahresabonnement des NASCOM-JOURNAL kostet für 1980 39.-. Ab 1.1.1981 kostet 48.- pro Jahr.
2. Der Betrag ist nach Rechnungstellung unter Angabe der Rechnungsnummer (bitte auf keinen Fall vergessen !!) auf unser Sonderkonto 299 26 - 674 beim Postscheckamt Ludwigshafen/Rhein zu überweisen.
3. Ein Abonnement läuft für 1 Jahr. Wird es bis 6 Wochen vor Ablauf des Kalenderjahrs nicht gekündigt, dann läuft es auch im nächsten Jahr weiter.
4. Es erscheinen 1980 insgesamt 8 Ausgaben, ab 1981 dann 12 Ausgaben (bzw. 1 oder 2 Doppelausgaben).

Inhaltsverzeichnis

- 3-4 Systemtestprogramm für NASCOM 1
- 4 NASCOM 1 spielt Melodien
- 5 Neuer Disassembler für NASCOM 1 u. 2.
- 5 Verbesserung der "Simple Graphik"
- 6-7 Ein analoger Joystick für den NASCOM
- 8 Z-80 Befehle mit 10 Byte Länge ???
- 9 NMI Taste für den NASCOM 1
- 10 Verbesserungen zu INVASION
- 11-14 Kompensation des mech. Spieles
- 15 Formatierprogramm Auto-Space und Auto-NL für Modify
- 16 Renumberprogramm für BASIC
- 17 Ansteuerung eines 50 BAUD-Fernschreibers über UART
- 18 Telespiel mit Ton
- 19 Billigst-Speichererweiterung mit MK 4118
- 19-20 NASCOM 1 Graphik "Brutal"
- 20 2 nützliche Unterprogramme
- 21-23 Simulation eines Geldspielautomaten
- 24 Pfeilspiel
- 25-26 Druckausgabe für den NASCOM 1
- 27 Tabulator Programm
- 28 Einfach Disassembler für NASCOM
- 29-31 Graphik - Erweiterungskarte für NASCOM 1
- 31 Neue Schachversion unterstützt ROM-Graphik

Listings

- 32 Listing des Telespiel mit Ton (zu S. 18)
- 33-34 Unterprogramm zum Speichern von String-Feldern auf Cassette mit dem 8k BASIC
- 35 Reverse
- 36-44 Listing des Plotterprogrammes (zu 11-14)
- 45-47 Softwaretreiber für Kugelpopf-Drucker
- 48 Disassembler-Listing

Neue Produkte / Annoncen

- 49 Graphik-Zusatzkarte für den NASCOM 1
- 50 Neue Produkte
- 51-52 Kleinanzeigen

SYSTEMTEST FÜR DEN NASCOM 1
 =====

Ein Defekt in meinem NASCOM 1 System machte mir bewusst, wie hilflos man ohne aufwendige und teure Messtechnik diesem Problem gegenübersteht. Die Betriebssysteme helfen bei der Fehlersuche auch nicht weiter. Aus diesem Grund habe ich ein einfaches Testprogramm geschrieben. Es wird sinnvollerweise in einem EPROM (2708) abgelegt.

Das Programm besteht aus folgenden Teilen:

- 1) "drive LED" setzen
- 2) Test der statischen Speicher (0800-0FFF)
- 3) Interner PIO-Test
- 4) UART-Test

Beschreibung der einzelnen Programmteile

Test 1: Die "drive LED" wird eingeschaltet

Test 2: Die Speicherplätze der statischen Speicher werden nacheinander mit den Prüfmustern AA H und 55 H geladen. Nach jedem Prüfmuster wird eine Zeit von 10 Sekunden gestartet, und nach Ablauf der Zeit das Muster in allen Speicherzellen getestet. Wird ein Fehler festgestellt, so wird der "Test 2" wiederholt. Ist der Speichertest ohne Fehler ausgeführt, wird die "drive LED" ausgeschaltet und auf dem Bildschirm der Text/Speichertest O.K. ausgegeben. Testdauer etwa 30 Sek.

Test 3: Mit diesem Programmteil wird der Datenverkehr zwischen der CPU und dem PIO Baustein geprüft. Dazu werden nacheinander die Muster AA H und 55 H an die Register A und B ausgegeben. Danach werden

die Register des PIO zurückgelesen und mit dem vorher ausgegebenen Muster verglichen. Ist der Vergleich fehlerhaft, wird die "drive LED" eingeschaltet und der Programmteil "Test 3" wiederholt. Ist der Vergleich O.K., wird der Text /interner PIO-Test O.K./ auf dem Bildschirm ausgegeben.

Test 4: Bei diesem Test wird die serielle Schnittstelle des UART geprüft. Das ASCII-Zeichen "U" (55 H) wird 256 mal ausgegeben. Das Programm wartet nach jeder Zeichenausgabe auf "UART ready". Ist der UART nicht "ready", wird die "drive LED" eingeschaltet. Kommt das "ready" Signal, wird die "drive LED" ausgeschaltet. Dadurch blinkt die "drive LED" in der Frequenz des UART-Taktes und bildet somit einen Maßpunkt (trigger).

Ist das Zeichen "U" 256 mal ausgegeben, werden die Texte / UART-Test O.K./ und /NASCOM 1 SYSTEMTEST O.K./ auf dem Bildschirm ausgegeben. Danach geht die CPU in den "HALT"-Status.

Hinweise zur Fehlersuche

Mit dem Testprogramm, einem einfachen Logik-Tester (TTL / CMOS Prüfstift) und einigen Ersatz-ICs ist es möglich, "Hardwarefehler" herauszufinden.

Der erste und wichtigste Schritt bei der Fehlersuche ist die Kontrolle der Betriebsspannungen. Liegen die Spannungen richtig an, wird das System ausgeschaltet und das Betriebsprogramm herausgenommen. Danach wird das EPROM mit dem Testprogramm auf den Steckplatz 0000-03FF eingesteckt und das Programm durch die Betätigung der "reset"-Taste gestartet. Der Test ist beendet, wenn die "HALT"-LED leuchtet.

Das ausführliche Assemblerlisting des Testprogramms ist für 10 DM, das Assemblerlisting und das Testprogramm in einem EPROM (2708) für 54 DM per Nachnahme zu beziehen bei:

H.G. Schäfer

██████████
 ████████ Melsungen

```

000 3E 10 D3 00 3E AA 32 00
008 08 21 00 08 11 01 08 01
010 FF 07 ED B0 06 14 11 FF
018 FF 1D 20 FD 15 20 FA 16
020 F5 21 00 08 01 FF 07 ED
028 A1 20 17 EA 27 00 FE A0
030 28 08 FE 55 28 08 FE 20
038 20 0A 3E 55 18 C8 3E 20
040 10 C4 18 C8 AF D3 00 31
048 00 10 CD CE 00 21 D8 00
050 11 10 09 01 10 00 ED B0
058 3E 0F 06 FF D3 06 D3 07
060 10 FA 3E AA CD 7C 00 3E
068 55 CD 7C 00 CD CE 00 21
070 E8 00 11 50 09 01 17 00
078 ED B0 18 1B F5 D3 04 D3
080 85 C1 AF DB 04 B8 20 07
088 AF DB 05 B8 20 01 C9 3E
090 10 D3 00 DD E1 18 C1 06
098 FF 3E 55 D3 01 DB 02 87
0A0 FA A9 00 3E 10 D3 00 18
0A8 F4 AF D3 00 10 EB CD CE
0B0 00 21 00 01 11 90 09 01
0B8 0F 00 ED B0 CD CE 00 21
0C0 10 01 11 10 0A 01 17 00
0C8 ED B0 CD CE 00 76 11 FF
0D0 FF 1D 20 FD 15 20 FA C9
0D8 53 70 65 69 63 68 65 72
0E0 74 65 73 74 20 4F 2E 6B
0E8 69 6E 74 65 72 6E 65 72
0F0 20 50 49 4F 20 2D 20 54
0F8 45 53 54 20 4F 2E 6B 2E
100 55 41 52 54 20 2D 20 54
108 65 73 74 20 4F 2E 6B 2E
110 4E 41 53 43 4F 4D 20 31
118 20 53 59 53 54 45 4D 54
120 45 53 54 20 4F 2E 6B 2E
128 30 31 3A 32 36 2E 31 30
130 2E 38 30 20 FF FF FF FF
    
```

NASCOM 1 SPIELT MELODIEN
 =====

Dieses einfache Programm erzeugt Melodien, die durch ein schnell zusammengesetztes Interface an SK 1 Pin 13 hörbar gemacht werden können. OC57/58 enthält die Anfangsadresse für eine Tonhöhentabelle, OC5B/5C enthält den Beginn der Tabelle für die Tonlängen. In OC5E ist die Anzahl der Töne gespeichert. Das Programm startet bei OC50.

```

Oc50 3e 04 00 00 0e 00 21 76 Oc dd 21 00 Od 06 80 dd
Oc60 5e 00 56 d3 00 15 20 fb ed 51 1d c2 62 Oc 23 dd
Oc70 23 10 ec c3 56 Oc
    
```

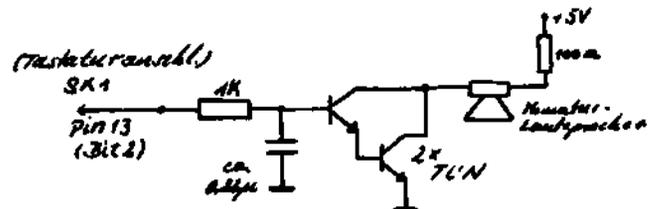
Im angegebenen Beispiel ist das Programm für 128 Töne ausgelegt, die Anfangsadresse der Tonhöhen liegt bei OC76, die Tonlängentabelle beginnt bei ODOO.

Mit folgenden Tabelle spielt Ihr NASCOM 1 das Präludium C-Dur von J.S. Bach.

```

Oc76 80 66 55 40 33 55 40 33 80 66 55 40 33 55 40 33
Oc86 80 71 4c 38 30 4c 38 30 80 71 4c 38 30 4c 38 30
Oc96 88 71 55 38 30 55 38 30 88 71 55 38 30 55 38 30
Oca6 80 66 55 40 33 55 40 33 80 66 55 40 33 55 40 33
Ocb6 80 66 4c 33 26 4c 33 26 80 66 4c 33 26 4c 33 26
Occ6 80 71 59 4c 38 59 4c 38 80 71 59 4c 38 59 4c 38
Ocd6 88 71 55 38 2a 55 38 2a 88 71 55 38 2a 55 38 2a
Oce6 88 80 66 55 40 66 55 40 88 80 66 55 40 66 55 40
Ocf6 2a ec 3a 3a e5 cc 3a 1b ae ec 50 64 78 a0 c8 78
Od06 a0 c8 50 64 78 a0 c8 78 a0 c8 50 5a 85 b4 d4 85
Od16 b4 d4 50 5a 85 b4 d4 85 b4 d4 4a 5a 78 b4 d4 78
Od26 b4 d4 4a 5a 78 b4 d4 78 b4 d4 50 64 78 a0 c8 78
Od36 a0 c8 50 64 78 a0 c8 78 a0 c8 50 64 85 c8 ff 85
Od46 c8 ff 50 64 85 c8 ff 85 c8 ff 50 5a 71 85 b4 71
Od56 85 b4 50 5a 71 85 b4 71 85 b4 4a 5a 78 b4 f0 78
Od66 b4 f0 4a 5a 78 b4 f0 78 b4 f0 4a 50 64 78 a0 64
Od76 78 a0 4a 50 64 78 a0 64 78 a0
    
```

Einfaches Interface, das im NASCOM-Gehäuse untergebracht werden kann und keinen zusätzlichen Verstärker benötigt.



Günter Böhm, ██████████
 Karlsruhe
 Tel.: ██████████

NEUER DISASSEMBLER

für NASCOM 1 und 2

Die Disassembler DIST4 und DISSYS sind speziell für den NASCOM 1 und 2 ausgelegt.

Die Version DIST4 läuft mit dem T4-Betriebssystem, DISSYS läuft unter NAS-SYS.

Beide Versionen sind sehr kompakte 3 Pass-Disassembler, die in 2 EPROMS 2708 Platz finden.

Der Dreipass-Disassembler ist in der Lage, Marken (Labels) zu setzen. Zur besseren Lesbarkeit sind die Labels nicht willkürlich durchnummeriert: die angesprochenen Adressen sind Bestandteil des Labelnamens.

Neben anderen Befehlen kennt dieser Disassembler auch einen Relocator-Befehl, er bietet daher die Möglichkeit, Objektprogramme zu verschieben. Die aus dem Maschinencode zurückübersetzten Quellprogramme können in ZEAP-kompatibler Form auf Kassette ausgegeben werden oder mit einem Drucker aufgelistet werden.

Die Anzeige ist je nach Betriebsart in Druck- oder Bildschirmseiten organisiert.

Die Betriebsarten des Disassemblers werden mittels Kommandobuchstaben aufgerufen.

Hier eine Übersicht über die einzelnen Kommandos:

- D disassembliert, Anzeige nur Bildschirm
- P disassembliert, Drucker und Bildschirm
- c erzeugt Quellfile auf Kassette
- Q setzt Verzögerung nach Zeilenende bei Kassette
- S definiert auszulassenden Speicherbereich (DEFS)
- R Relocator
- N Rückkehr in Monitor

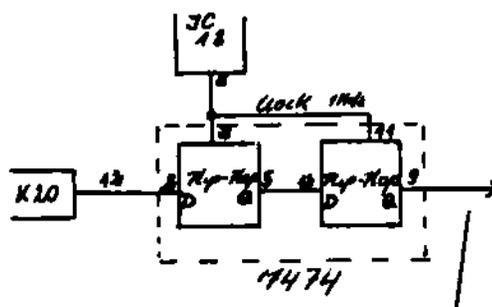
Außerdem gibt es noch zwei Befehle, mit deren Hilfe man den Disassembler um eigene Unterprogramme erweitern kann.

VERBESSERUNG ZUR "SIMPEL GRAPHIK"

(Heft 5/80)

Im Gegensatz zu Herrn Lotter bin ich mehr wohl der Ansicht, daß sich die Invertierung des davorstehenden Zeichens bei gesetztem Bit 7 störend auswirkt.

Einfacher als durch Nachschalten von 8 ExOr Gattern läßt sich das Problem durch eine Verzögerung mit einem 7474 lösen, was sich in meinem System schon seit einiger Zeit bewährt hat.



weiter in Heft 5/80

Günter Böhm,
Karlsruhe
Tel.:

LEERKASSETTEN

Speziell geeignet für Datenaufzeichnung. Hochwertiges BASF-Band. Kassette 5-fach verschraubt. Kassette 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
Waldstraße 20
6728 Germersheim/Rhein
Tel.: 07274/2756

Mit einem analogen Joystick ist es möglich, Bewegungen des Steuerknüppels in der X- Y- Ebene proportional dem Rechner mitzuteilen. Von dem verwendeten Vierquadranten - Potentiometer werden 2 Potis benutzt. Diese bestimmen die Impulsdauer zweier Monoflops. Die Digitalisierung erfolgt nach dem Zählverfahren.

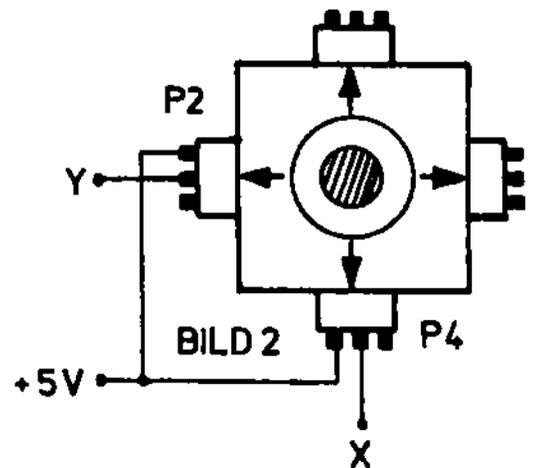
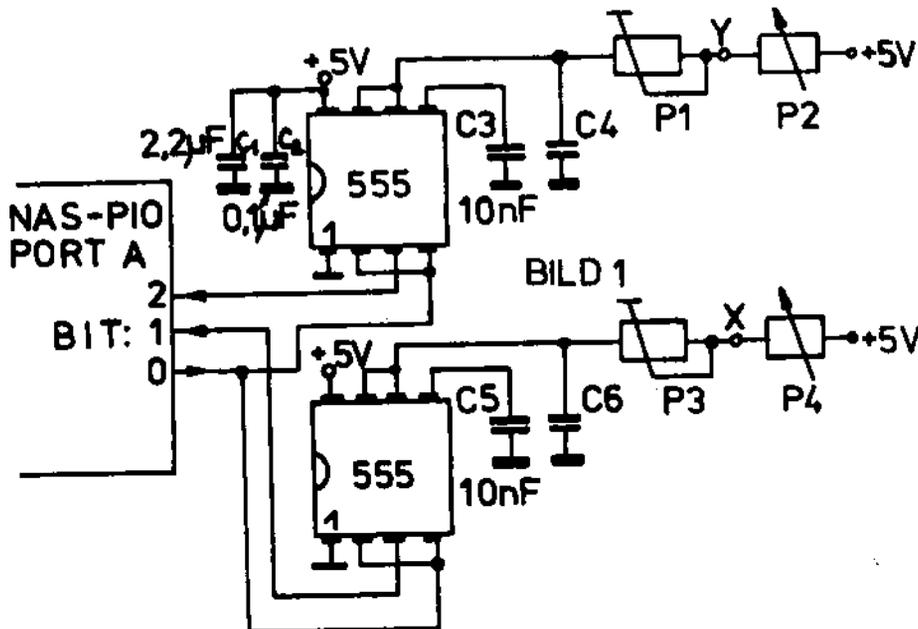
Der augenblicklichen Stellung des Joysticks wird direkt eine Bildschirmadresse zugeordnet. Im Gegensatz zu digitalen Joysticks kann hier nach einem "CALL JOYSTICK" dem Hauptprogramm die aktuelle Cursoradresse direkt im HL-Register übergeben werden.

Die Bilder 1 und 2 zeigen den Hardwareaufwand und den Anschluß an den NASCOM. Die relokatierbare Subroutine "JOYSTICK" ist ab 0000 H einzugeben. Nach dem Start des Testprogramms ab 0036 H sind die Trimmer P1 und P3 so abzugleichen, daß bei der Joystickbewegung:  der Cursor "⊙" exakt dieser Bewegung nachkommt. Bei Bauteiltoleranzen und Abweichungen in der Versorgungsspannung kann ein Feinabgleich von C4 und C6 nötig sein.

Eine mögliche Anwendung zeigt das Programm "Snake". Mit dem Joystick läßt sich dann eine "Schlange" aus max. 63 Elementen zeichnen, die sich bei einer Fortsetzung automatisch am Ende verkürzt.

Werden die Potis P2 und/oder P4 durch NTC-, LDR-Bauelemente ersetzt, ergibt sich ein breites Experimentierfeld.

U. Krüger



TESTPROGRAMM

```

ruft Subroutine Joystick auf
Start: 0036 H
Exit : RESET

EF 0C 00  RST 28H      } Bildsch.lösch.
CLS        } (NAS-SYS)
00

3E 20     LD A,20 H   A = Blank
21 00 00  LD HL,0000
E5        PUSH HL    save alte Curs.adr.
CD 00 00  CALL JOYSTICK HL = neue Curs.adr.
D1        POP DE     DE = alte Curs.adr.
B7        OR A
ED 52     SBC HL,DE   } alte = neue
19        ADD HL,DE   } Cursoradr. ?
28 F5     JRZ ①      ja
36 0F     LD (HL), "⊙" nein
12        LD (DE), A  vor. Zeich. lösch.
18 F0     JR ①
    
```

C2,C3,C5 Keramikkondensator
C4,C6 Folienkondensator (1)

Potis P2,P4 Kond.	20 kΩ	47 kΩ
C4	0,15µF + 27nF	47 nF + 10 nF
C6	0,15µF + 4,7nF	39 nF + 4,7nF
P1	4,7kΩ	10 kΩ
P3	10 kΩ	25 kΩ

Subroutine JOYSTICK

Start: 0000 H
 Exit: Cursoredresse in HL
 alle anderen Reg. unverändert

```

F5      PUSH AF
C5      PUSH BC
01 02 02 LD BC, 0202 H  B= Counter
                        C= AND Maske für die
                        Abfrage von Bit 1
                        an Port A
3E FF   LD A, FF H      } setze PID in CONTROL
03 06   OUT (06), A     } MODE
30      DEC A           } A= FE H = 1111 1110 B
03 06   OUT (06), A     } setze Bit 0 von Port A
                        als Ausgang, Bit 1-7
                        als Eingang

18 03   JR ②
6C      ① LD L,H
0E 04   LD C,04        C= AND Maske für die
                        Abfrage von Bit 2
                        an PortA.

AF      ② XOR A
67      LD H,A
03 04   OUT (04), A     } A= 00
                        H= 00
                        Reset Monoflops über
                        Bit 0
3E 20   LD A, 20 H     } Verzögerung zur
FF      RST 38 H       } sicheren Entladung von
                        C4 und C6
3C      INC A
03 04   OUT (04), A     } A= 01
24      ③ INC H         } Monoflop triggern
08 04   IN (04), A     } Zählchleife zur
A1      AND C           } Bestimmung der
20 FA   JRNZ ③         } Impulsdauer
10 EB   DJNZ ①         } 2. Durchgang falls
                        B ≠ 00

7C      LD A,H
1F      RRA
1F      RRA
1F      RRA
1F      RRA
1F      RRA
CB 10   RR L
1F      RRA
CB 10   RR L
E6 03   AND 03
06 08   ADD 08
67      LD H,A
C1      POP BC
F1      POP AF
C9      RET
  
```

"SNAKE"

ruft Subroutine Joystick auf
 Start: 0D36 H
 Exit : RESET

```

EF 0C 00 RST 28H      } Bildsch. löschen
                        CLS
                        00
                        } (NAS-SYS)
21 7F 00 LD HL,0D7F H }
06 80   LD B, 80 H    } Adressentabelle
23      ① INC HL       } von 0D80-0DFF
36 00   LD (HL),00    } löschen
10 FB   DJNZ ①
E5      ② PUSH HL     } save alte Curs.adr.
CD 00 00 ③ CALL JOYSTICK } HL = neue Curs.adr.
D1      POP DE       } reserve alte Curs.adr.
B7      OR A
ED 52   SBC HL,DE    } alte = neue
19      ADD HL,DE    } Cursoredresse ?
28 F5   JRZ ②        } ja
D9      EXX          } nein
21 82 00 LD HL,0D82 H }
11 80 00 LD DE,0D80 H } Adressentabelle
01 7F 00 LD BC,0D7F H } 2 mal shiften
ED 80   LDIR
D9      EXX
22 FE 00 LD(0D7E),HL } neue Curs.adr.eintr.
3E 20   LD A,20 H     } auf neuer Curs.adr.
BE      CP (HL)       } Blank ?
3E 0F   LD A,0F H     } A= "0"
28 02   JRZ ④        } ja
3E 0E   LD A,0E H     } nein, A= "0"
77      ④ LD (HL),A   } neues Zeichen drk.
E5      PUSH HL      } save Curs.adr.
2A 80 00 LD HL,(0D80) } lösche letztes
36 20   LD (HL),20 H } Zeichen der Schlange
18 04   JR ③
  
```

Hard- und Software sind für 2 MHz CPU
 Takt ausgelegt.
 NASBUG Monitorbenutzer haben den Befehl
 EF 0C 00 (Bildsch.löschen)
 durch die folgende Befehlsfolge zu ers.:

```

EF 1E 00 RST 28H      } Bildsch. löschen
                        CLS
                        00
                        } (NASBUG)
2A 18 0C LD HL,CURSOR } Promptzeichen
36 20   LD (HL),Blank } löschen
  
```

Z-80 Befehle mit 10 Byte Länge ???

Der Autor stellt die Behauptung auf, daß die Z-80 CPU nicht nur die max. 4 Byte langen Befehle (z.B. LD BC,(nn)) auszuführen imstande ist, sondern darüberhinaus auch mehrere 10 Byte lange Befehle korrekt abarbeiten kann!!

Maschinensprachen-Insider haben die Genugtuung, sich nicht den Beweis im nächsten NASCOM JOURNAL anschauen zu müssen.

Allen anderen sei zum Trost schon vorab erklärt, daß obige Insider garantiert nicht imstande sind, dieses Wissen nutzbringend anzuwenden!

U. Krüger

Z-80 Befehle mit 10 Byte Länge !?!

Hier der Beweis:

Führen Sie das folgende Programm im Single Step aus:

	0000 00		NOP
	0001 3E 02		LD ACCU , 02
→	0003 00 00 00 00 00 00 00 00 3E 04	←	
	0004 00		NOP
	000E 00		NOP

Bei jedem 00-Befehlsbyte wird für die Dauer des nachfolgenden Befehls das IX- anstelle des HL-Doppelregisters intern aktiviert. Gleiches gilt sinngemäß auch für das IV Register bei einem 00-Byte.

In diesem Beispiel wird anschließend 3E 04, also LD ACCU mit 04 ausgeführt. Auch andere Befehle die nicht die Register H oder L betreffen oder mit "CB", "ED" beginnen, wären anschließend normal ausgeführt worden.

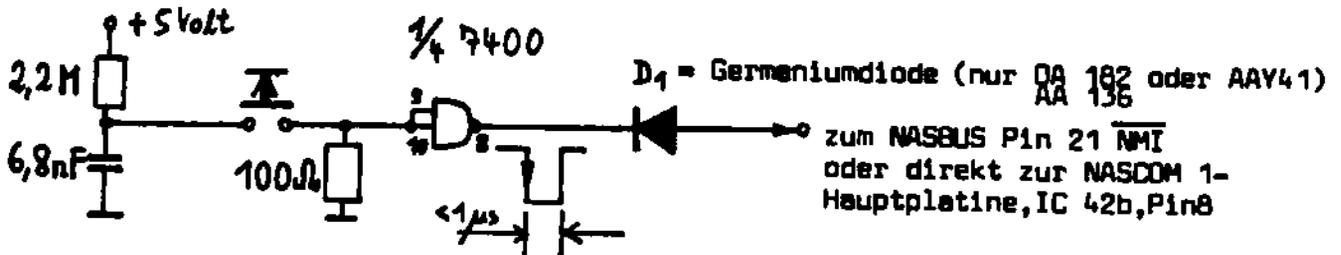
Nun wird auch deutlich, wie die neuen nützlichen Befehle funktionieren, die in der "ELEKTRONIK" (14/80 S.83 und 18/80 S.24) veröffentlicht wurden.

Ansonsten scheint die Z-80 CPU noch einige weitere Geheimnisse zu bergen: Die Gesetzmäßigkeit der Befehlsfolge "ED 70" bei verschiedenen vorausgegangenen 8 Bit arithmetischen Operationen konnte der Autor hieher noch nicht erkennen.

U.Krüger

NMI Taste für den NASCOM 1

Mit dem folgenden Schaltungsvorschlag kann auf einfache Weise ein Non Maskable Interrupt mittels externer Taste ausgelöst werden:



Als NANDgatter kann IC 3 der Tastaturplatte verwendet werden, welches zur Hälfte unbenutzt ist. Die Schaltung unterdrückt Kontaktprellen der Taste und beschränkt die Impulsdauer auf weniger als $1 \mu\text{sec}$.

Ein solcher Impuls wird lediglich bei einem andauernden WAIT- oder aktiven BUSREQUEST-Signal von der CPU ignoriert. Ansonsten wird jedes laufende Programm unterbrochen!

Die Adresse der nächsten Instruktion (des unterbrochenen Programms) wird als spätere RETURNadresse auf den Stack geladen. Dann führt die CPU über $\$NMI = 0C47H$ (NASbug T2, T4) bzw. $\$NMI = 0C70H$ (NAS-SYS 1) aus.

Dort steht der Sprungcode $C3H$ mit der nachfolgenden Adresse, die der Benutzer beliebig ändern kann. Ein Interrupt im direkten NASBUG-Monitorbetrieb ist nicht sinnvoll, da der interne Monitortack ab $0C33H$ zu klein ist und Fehler verursachen kann. Wird der NAS-SYS Monitor entsprechend modifiziert (s. NASCOM JOURNAL 4/80), so können auch unterbrochene Anwenderprogramme nach der Interruptroutine fehlerfrei weiterlaufen.

PS. Aufmerksame Betrachter werden feststellen, daß die Schaltung den Ausgang \bar{Q} von IC 42 b nach Masse kurzschließt. TI gibt jedoch im TTL DATA BOOK (76, S.6-56) für den 74LS74A einen für die Dauer von 1 sec max. zul. Kurzschlußstrom von -100 mA an, der jedoch durch den internen Collectorwiderstand (120Ω), der begrenzten "Sinkcapability" des 7400 und der Schwellenspannung an D1 bei weitem nicht erreicht wird!

Ulrich Krüger

Die unten aufgeführten Änderungen beziehen sich auf das Spiel "INVASION", veröffentlicht im JOURNAL 2/80. Der Spielablauf wird - meiner Meinung nach - wesentlich belebter und attraktiver. Falls Interesse : ausprobieren!
 (Die letzten drei Modifikationen wurden erst durch das originale kommentierte Listing von Graham Clarke ermöglicht.)

Adresse	orig.Listing	neu:	
OCAA	CD 69 00	CD D2 DF	CALL OFD2
OCF6	D2 CA 0C	D2 AA 0C	JPNC OCAA
OD2D	D2 CD 0D	D2 AA 0C	JPNC OCAA
OFD2	-- -- --	21 02 0C	LD HL,0C02
OFD5	-- --	C8 A6	RES 4,(HL)
OFD7	--	23	INC HL
OFD8	-- --	C8 A6	RES 4,(HL)
OFDA	-- -- --	C3 69 0D	JP K8D
ODEC	DD 36 41 20	DD 36 41 19	LD (IX+41H)," $\frac{1}{2}$ "
			ermöglicht wiederholtes Laden einer Rakete aus einem unzerstörten Magazin
OD44	3E 04	3E 01	LD A,01
			verringert die Nachladeverzögerung einer Rakete
DE34	3E 50	3E 40 (z.B.)	LD A,40H
			Erhöhung der Bombenabwurfrate

Kompensation des mechanischen Spieles

Wie schon im Nascom-Journal 3/80 erwähnt, herrscht bei dem Getriebe, das den Schlitten zieht, ein Spiel von 0,5 mm (das sind 18 Schritte). Um diesen Fehler zu beseitigen, kann man vor den Aufrufen des Unterprogrammes ABPLOT das folgende Korrekturprogramm einschieben. Dies geschieht so, daß an Stelle von CF5B_h einfach 0C80_h als Adresse von ABPLOT verwendet wird. Das Programm überprüft zunächst, ob der Zeichenstift nach rechts oder links bewegt werden soll und ob diese Richtung mit der vorhergehenden übereinstimmt. Bei Gleichheit ist keine Korrektur notwendig und der eigentliche Zeichenvorgang erfolgt sofort. Sind die beiden Richtungen jedoch verschieden, so werden in der neuen Richtung 18 zusätzliche Schritte ausgeführt, bevor der Zeichenvorgang beginnt. Dadurch bewegen sich zunächst nur der Motor und das Getriebe; erst bei den Schritten, die über 18 hinausgehen und die schon zum Zeichenvorgang gehören, wird auch der Zeichenstift mitbewegt.

```

                                0010 ,zusatz zum
                                0020 ,plotter-ansteuerungs-programm
                                0030 ,korrektur des mechanischen spieles
                                0040 ,peter bentz,1980
                                0050 ,
cf5b                            0060 abplot equ hcf5b
0c4f                            0070 ram    equ h0c4f
ce0f                            0080 abs    equ hce0f
ce28                            0090 neghl  equ hce28
cf14                            0100 xdriфт equ hcf14,(binaeraufruf)
                                0110 ,
0c80                            0120 '      org h0c80 ,(frei verschiebbar)
0c80 2a540c                      0130 start ld    hl,(5ram)
0c83 ed5b560c                   0140      ld    de,(7ram)
0c87 e5                          0150      push hl
0c88 d5                          0160      push de
0c89 ed5b500c                   0170      ld    de,(1ram)
0c8d b7                          0180      or    a
0c8e ed52                       0190      sbc  hl,de
0c90 3a5a0c                      0200      ld    a,(11ram)
0c93 f5                          0210      push af
0c94 cd0fce                      0220      call abs
0c97 f1                          0230      pop  af
0c98 215a0c                      0240      ld    hl,11ram
0c9b be                          0250      cp   (hl)
0c9c 280b                        0260      jr   z l1
0c9e fe23                        0270      cp   h23
0ca0 211200                      0280      ld    hl,18
0ca3 cc28ce                      0290      call z neghl
0ca6 cd14cf                      0300      call xdriфт
0ca9 d1                          0310 l1   pop  de
0caa e1                          0320      pop  hl
0cab 22540c                      0330      ld   (5ram),hl
0cae ed53560c                   0340      ld   (7ram),de
0cb2 c35bcf                      0350      jp  abplot

```

Prinzip der Ansteuerung für einen Schrittmotor

Die zwei Spulen eines Schrittmotors werden, wie im Nascom-Journal 3/80 beschrieben im Viertaktbetrieb angesteuert und die Stromrichtung in jeder Spule durch eine 1 bzw. 0 am Ausgang der PIO bestimmt. Bei einer fortlaufenden Weiterschaltung des Motors in eine Richtung müssen folgende Bit-Kombinationen an den entsprechenden Ausgängen der PIO vorliegen:

Schritt 1: 11
 Schritt 2: 10
 Schritt 3: 00
 Schritt 4: 01

Dieses Muster erhält man, wenn man das 8-bit-Wort $33_{10} = 00110011$ durch einen RRC- oder RLC-Befehl laufend nach rechts oder links rotiert und dabei nach jedem Schritt die zwei rechten Bits beobachtet. Deren Wert gibt man an die PIO weiter.

00110011	Schritt 1
01100110	Schritt 2
11001100	Schritt 3
10011001	Schritt 4
00110011	Schritt 5 = Schritt 1

Da bei dieser Ansteuerungsart die absolute Positionierung nur über einen Bereich von vier Schritten erfolgen kann, muß man gleichzeitig mit jedem Schritt nach rechts oder links einen Positionszähler hoch- oder herunterzählen. So hat man im Mikrocomputer ständig eine Angabe über die absolute Position des Schrittmotors. Selbstverständlich kann der Mikroprozessor die einzelnen Bitmuster so schnell berechnen, daß der Schrittmotor nicht mehr mitkommt. Deshalb muß nach jedem Schritt eine Pause eingelegt werden, damit der Motor in die neue Position schwingen kann. Mit dieser Pause hat man die Möglichkeit, die Drehzahl des Schrittmotors ohne die Hilfe eines mechanischen Getriebes quasi stufenlos zu verändern und hat doch eine reproduzierbare, quarzgenaue Geschwindigkeit.

Beispiel für die Ansteuerung eines Schrittmotors

Wir nehmen an, der Schrittmotor sei über Bit 0 und 1 des Ports 05 der PIO angeschlossen und die anderen Ausgänge des Ports seien unbenutzt. Man gibt EC50 bzw. EC70, gefolgt von einem Leerraum und der gewünschten hexadezimalen Schrittzahl ein. Nach "New Line" läuft der Motor je nach gewählter Startadresse die eingegebene Schrittzahl nach rechts oder links und bleibt dann wieder stehen.

```

0010 ,beispiel zur ansteuerung
0020 ,eines schrittmotors
0030 ,
0286 parse equ h0286
0c0e arg2 equ h0c0e
0035 kdel equ h0035
0060 ,
0c50 0070 org hc50
0c50 3e0f 0080 start1 ld a,h0f ,pio
0c52 d307 0090 out (07),a ,aktivieren
0c54 2a0e0c 0100 ld hl,(arg2),schrittzahl einlesen
0c57 7c 0110 l1 ld a,h ,wenn null
0c58 b5 0120 or l ,dann
0c59 ca8602 0130 jp z parse ,ruecksprung
0c5c 2b 0140 dec hl
0c5d 3a8f0c 0150 ld a,(sr)
0c60 0f 0160 rrca , rechts-shift
0c61 328f0c 0170 ld {sr},a
0c64 d305 0180 out (05),a ,ausgabe
0c66 0601 0190 ld b,1 ,variable
0c68 cd3500 0200 l2 call kdel ,pause
0c6b 10fb 0210 djnz l2
0c6d 18e8 0220 jr l1
0c6f 00 0230 nop
0240 ,
0c70 3e0f 0250 start2 ld a,h0f
0c72 d307 0260 out (07),a
0c74 2a0e0c 0270 ld hl,(arg2)
0c77 7c 0280 l3 ld a,h
0c78 b5 0290 or l
0c79 ca8602 0300 jp z parse
0c7c 2b 0310 dec hl
0c7d 3a8f0c 0320 ld a,(sr)
0c80 07 0330 rlca , links-shift
0c81 328f0c 0340 ld {sr},a
0c84 d305 0350 out (05),a
0c86 0601 0360 ld b,1
0c88 cd3500 0370 l4 call kdel
0c8b 10fb 0380 djnz l4
0c8d 18e8 0390 jr l3
0c8f 33 0400 sr defb h33,(schieberegister)

```

Diese beiden Programme sind bis auf den RRCA bzw. RLCA-Befehl identisch. Deshalb könnte man per Programm eine Fallunterscheidung machen und je nach Drehrichtung in einem Programm einen RRCA oder RLCA-Befehl einfügen. Das ist zwar eine etwas unschöne Methode, spart aber viel Speicherplatz und wird deshalb in dem Plotter-Ansteuerungs-Programm angewendet.

Erweiterung auf zwei Schrittmotoren

Für zwei Motoren, die unabhängig voneinander angesteuert werden sollen, muß das beschriebene Konzept beträchtlich erweitert werden. Wenn man nur waagerechte oder senkrechte Linien mit dem Plotter zeichnen will, genügt es ja, wenn sich immer nur ein Schrittmotor dreht. Um aber beliebige schräge Linien zeichnen zu können, wird eine etwas umfangreichere Arithmetik benötigt, die mindestens so schnell ist wie die maximale Frequenz der Motoren.

Dazu müssen zuerst aus den Absolutkoordinaten von Start und Ziel der zu zeichnenden Linie die Relativbewegung in X- und Y-Richtung bestimmt werden und je nach Vorzeichen die Drehrichtung der Motoren festgelegt werden. Ist die Bewegung für einen Motor gleich Null, wird nur der andere Motor um die entsprechende Schrittzahl bewegt. Dann wird durch eine Division die Steigung der zu zeichnenden Linie berechnet. Nun wird Schritt für Schritt eine Treppenkurve gezeichnet, die der gedachten exakten Linie möglichst nahe kommt. Dies geschieht so, daß das Programm zunächst einen Schritt waagrecht zeichnet und darauf so lange in der senkrechten Richtung zeichnet, bis die Treppenkurve die theoretische Linie berührt oder schneidet. Dann wird wieder ein waagrechtter Schritt durchgeführt und dieser Algorithmus wiederholt sich so lange, bis der Zielpunkt erreicht ist.

Peter Bentz

■ ■ ■ Mülheim/Ruhr

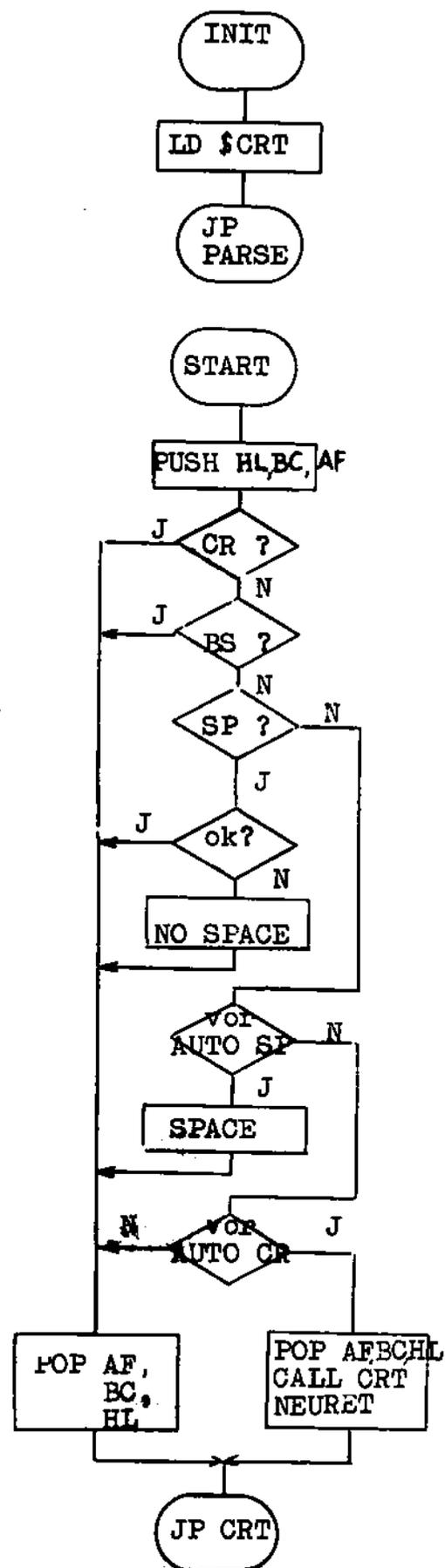
Formatierprogramm
- Auto-Space und Auto-NL für Modify -

F90	21	99	OF	INIT	LD HL, sSTART
F93	22	4B	OC		LD \$CRT, HL
F96	C3	86	O2		JP PARSE
F99	E5	C5	F5	START	PUSH HL, BC, AF
F9C	FE	1F			CP CR
F9E	28	38			JRZ ENDE 1
FA0	FE	1D			CP BS
FA2	28	34			JRZ ENDE 1
FA4	FE	20			CP SP
FA6	3A	18	OC		LD A CURSCHADR L
FA9	20	0A			JRNZ FORMAT
FAB	FE	8E			CP MON SF ADR
FAD	28	29			JRZ ENDE 1
FAF	F1	3E	OO	NOSPACE	POP AF, LD A, OO
FB2	F5				PUSH AF
FB3	18	23			JR ENDE 1
FB5	21	DE	OF	FORMAT	LD HL, SPTAB
FB8	01	07	OO		LD BC, LÄNGE
FBB	ED	B1			CPIR
FBD	28	14			JRZ SPACE
FBF	FE	A8			CP AUTO CR ADR
FC1	20	15			JRNZ ENDE
FC3	F1	C1	E1	CR	POP AF, BC, HL
FC6	CD	3B	O1		CALL CRT
FC9	D1	D1			POP DE, POP DE
FCB	11	BA	O1		LD DE, CRT-RETADR
FCE	D5				PUSH DE
FCF	3E	1F			LD A, CR
FD1	18	08			JR ENDE 2
FD3	3E	20		SPACE	LD A, SPACE
FD5	CD	3B	O1		CALL CRT
FD8	F1	C1	E1	ENDE 1	POP AF, BC, HL
FDB	C3	3B	O1	ENDE 2	JP CRT
FDE	94	97	9A	SPTAB	
FD1	9D	AO	A3		
FE4	A6				
FE5					

Das Programm ist als Hilfe bei der Eingabe langer Programme in Maschinensprache gedacht. (Wurde nur mit T4 getestet.) Es wird mit E F90 gestartet. Anschließend kann normal M aufgerufen werden. Nach jedem Byte wird ein SPACE gesetzt, nach dem achten Byte ein NL. Das Formatierprogramm nach beendeter Eingabe mit Reset verlassen, da es bei anderen Befehlen stören würde!! Zur eventuellen Veränderung der Adresslage sind die Adressen in F91/F92 und in FB6/FH7 entsprechend anzupassen.

P.S. Es ist wahrscheinlich, daß T4-Spezialisten auch einfachere Lösungen finden bzw. kennen.

Herbert Grasl,
 Nonnenhorn



8k BASIC - Zeilennummernprogramm

Das folgende Programm ändert die Zeilennummerierung eines BASIC-Programms. Es ist voll verschiebbar, kann also irgendwo im Speicher stehen.

Wichtig ist nur, daß man nach jedem Kaltstart die Anfangsadresse in USRLOC einträgt, beim ROM-BASIC in 1004 H, 1005 H.

Das Programm wird mit Hilfe der USR(x) - Funktion aufgerufen, wobei als Argument x die neue erste Zeilennummer angegeben wird.

Nun kann man nicht einfach "USR(x)" als Befehl eingeben - der Erfolg wäre ein SN-Error. Die folgenden Möglichkeiten sind erlaubt:

```
PRINT USR(x)
A = USR(x)
```

Die neuen Zeilennummern werden - ausgehend von der ersten - jeweils um 10 erhöht. Die Schrittweite 10 ist in Zeile 70 festgelegt.

Was das Programm nicht kann, es prüft nicht, ob die Zeilennummern noch erlaubt sind (d.h. kleiner als 65529), und es ändert auch nicht die Nummern nach GOTO, GOSUB.

Falls sich jemand an diese Aufgabe setzen will, hier ein paar Erklärungen zum Aufbau des Zeilenbuffers:

Der Zeilenbuffer beginnt beim ROM-BASIC auf 10FA. Die einzelnen Zeilen sind (nach Zeilen-NR. sortiert) im folgenden Format abgelegt:

```
2 byte Adresse der nächsten Zeile
2 byte Zeilen-Nr. (HEX)
n byte Zeileninhalt: Ziffern,Space in ASCII;alles andere als "Token"
```

```
0010 ; RENUMBER (8k BASIC - ROM)
0020 ; Startadresse in 1004 eintragen
0030 ; Aufruf mit USR(neue 1.Zeilennr.)
0040 ;
0E00 0050      ORG #E00
0E00 CD8BE9 0060      CALL #E98B
0E03 010A00 0070      LD BC,10
0E06 3AFB10 0080 ; Test, ob Zeilen vorhanden
0E09 B7      0090      LD A,(#10FB)
0E0A C8      0100      OR A
0E0A C8      0110      RET Z ; nein, leer
0E0B ED53FC10 0120 ; Ja:
0E0F 2AFA10 0130      LD (#10FC),DE ; 1.Zeilennr.
0E0F 2AFA10 0140      LD HL,(#10FA) ; Adr. 2.Zeile
0E12 23      0150 ;
0E12 23      0160 LOOP INC HL
0E13 23      0170      INC HL ; Adr. fertig
0E14 7E      0180      LD A,(HL)
0E15 23      0190      INC HL
0E16 B6      0200      OR (HL)
0E17 C8      0210      RET Z ; Ende wenn dort 0000
0E18 EB      0220 ; berechne Zeilennummer:
0E18 EB      0230      EX DE,HL
0E19 09      0240      ADD HL,BC
0E1A EB      0250      EX DE,HL
0E1B 72      0260 ; trage Znr. ein:
0E1B 72      0270      LD (HL),D
0E1C 2B      0280      DEC HL
0E1D 73      0290      LD (HL),E
0E1E 2B      0300 ; HL = Adresse(naechste Znr.) :
0E1E 2B      0310      DEC HL
0E1F 7E      0320      LD A,(HL)
0E20 2B      0330      DEC HL
0E21 6E      0340      LD L,(HL)
0E22 67      0350      LD H,A
0E23 18ED    0360      JR LOOP
```

1 byte 00 als Endkennung

Nach "NEW" werden die Speicherstellen 10FA, 10FB gelöscht, der Rest bleibt unverändert.

Die Token sind 1 byte-Kürzel, für GOTO steht 00, für GOSUB 8C usw..

Das Ende des Zeilenbuffers ist mit 0000 gekennzeichnet.

ANSTEUERUNG EINES 50 BAUD FERNSCHREIBERS
 ÜBER UART

Üblicherweise wird ein Fernschreiber über den IN/OUT Port vom NASCOM 1 angesteuert. Wer sich diesen Port aber für andere Zwecke freihalten möchte, kann den serienmäßigen UART verwenden, wenn einige kleine Änderungen an der Hardware vorgenommen werden. Um den Takt für den Fernschreiber bereitzustellen, kann über LK 4 ein externer Taktgenerator mit 800 Hz angeschlossen werden, (die der UART intern auf 50 Hz teilt). Einfacher ist es meiner Meinung nach, durch Nachschalten eines Teilers 74 LS 191 an den eingebauten IC 33 die benötigte Frequenz zu erzeugen. Hier kann es nötig werden, den Kondensator C12 zu verändern. (Bei meinem System genügte das Parallelschalten eines 4,4n Kondensators. Die Frequenz kann dann am Wendeltrimmer genau eingestellt werden).

Man benötigt keinen Frequenzzähler, wenn man einfach das Programm laufen läßt und dabei solange reguliert, bis der Fernschreiber richtig ausdrückt.

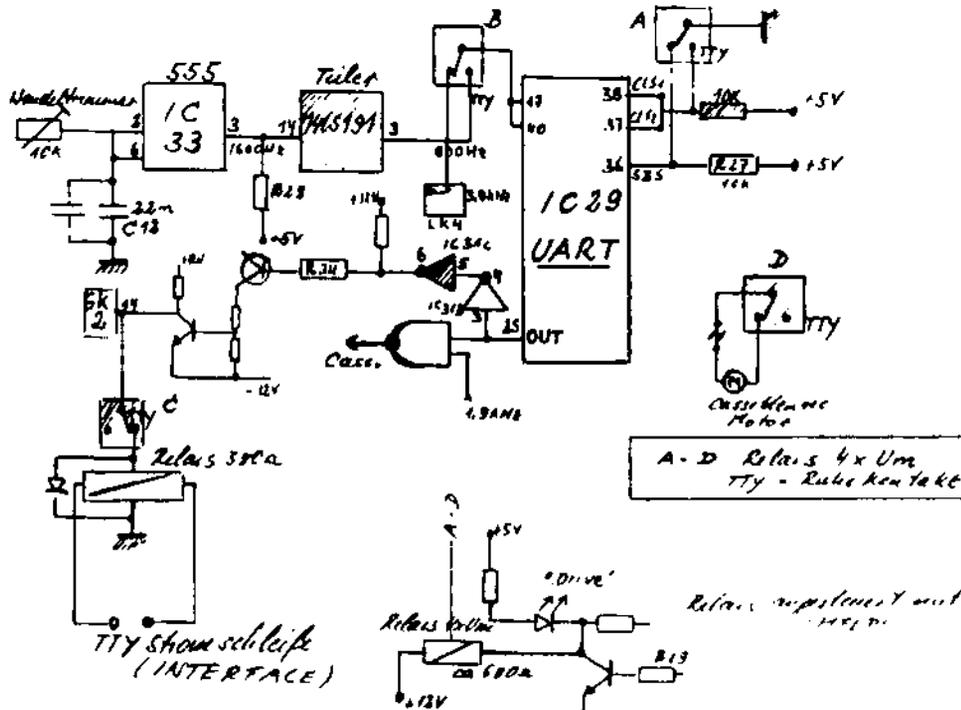
Einfacher ist die Umstellung auf 5 Bit und die Einstellung des Stop Bits. Pin 37 und 38 des UART werden herausgebogen und über einen 10k Widerstand mit +5V verbunden. Für TTY-Betrieb werden sie über einen Schalter an Masse gelegt. Die Brücke an LK 2 ist so zu verändern, daß Pin 36 des UART bei TTY-Betrieb über R27 an +5V liegt, im Cassettenbetrieb über Schalter an Masse geschlossen wird.

Die Stromschleife des Fernschreibers wird durch ein Relais unterbrochen, das durch -12V von Pin 14 an SK 2 betrieben wird.

Damit die Stromschleife im Ruhezustand geschlossen bleibt, muß der Ausgang des UART nochmals invertiert werden. Dies kann sehr einfach geschehen, indem man das unbenutzte Gatter IC 31 c verwendet. Hier werden die Pins 4, 5 und 6 herausgebogen und entsprechend verbunden.

Die Umschaltung von Cassetten Modus auf Fernschreiber kann mit Umschaltern per Hand erfolgen. Eleganter ist die Lösung, ein Relais zu verwenden, das durch den Befehl "MFLOP" den Cassettenmotor startet, das TTY-Relais ausschaltet und den UART in Cassettenmodus bringt.

Günter Böhm,
 Karlsruhe
 Tel.:



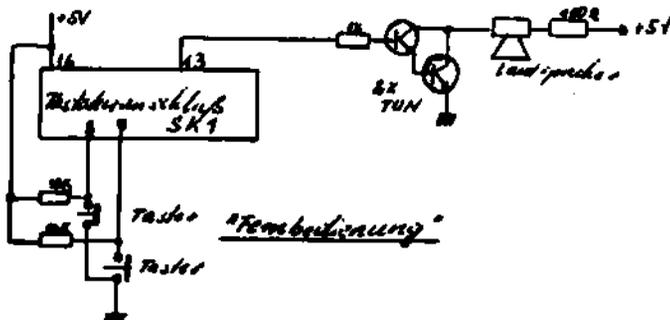
TELESPIEL MIT TON

Dieses Telespiel bildet auf der rechten Hälfte des Bildschirms eine "Mauer" von Klötchen ab, von denen sich nach einem Zufallsprogramm jeweils eines löst, sich nach links bewegt und mit einem "Schläger" abgeschossen werden soll. Ein Zähler zählt dabei bei Treffern aufwärts, bei Fehlschlägen abwärts. Erreicht der Zähler Null, ist das Spiel verloren, wobei ein Trauermarsch ertönt, erreicht er zehn, erklingt ein Tusch, und das Spiel wird als gewonnen angezeigt. Treffer und Fehlschläge werden ebenfalls hörbar gemacht, auch das Berühren des oberen oder unteren Bildrandes durch den Schläger erzeugt ein unterschiedliches akustisches Signal. Am Ende des Spiels kann ein neues Spiel mit veränderter Geschwindigkeit gewählt werden.

Die Tonerzeugung erfolgt über ein Interface, wie z.B. in Heft 8 des NASCOM-JOURNALS, wobei allerdings Bit 2 vom Port 8 verwendet wird (d.h. Pin 13 des Tastaturanschlusses).

Die Bewegung des Schlägers erfolgt über eine einfache Fernbedienung, die ebenfalls am Tastaturanschluss abgezapft wird. (Siehe Zeichnung)

Man startet das Spiel bei 8D38. Am Ende kann durch Drücken der Zifferntasten eine neue Geschwindigkeit eingegeben werden, wobei 1 die schnellste, z.B. 4 eine sehr langsame Geschwindigkeit ergibt. Ein neues Spiel wird durch Drücken der NEW LINE - Taste gestartet.



SO KÖNNEN SIE SELBST MELODIEN PROGRAMMIEREN

Die Übersicht gibt in Hex. die Werte an, die Sie zur Erzeugung der tiefsten Oktave benötigen. Die höheren Oktaven ergeben sich durch jeweiliges Halbieren der Werte. Beispiel: C = FF, c = 80, c' = 40, c'' = 20 etc.

Daneben finden Sie die Werte für die zugehörige Tonlänge, umgerechnet in vier verschiedene Notenwerte. (Leider benötigt jede Tonhöhe einen anderen Faktor, was die Melodienprogrammierung etwas umständlich macht).

Ton	♩	♪	♫	♮	Tonhöhe
C	0A	14	28	50	FF
D	0B	16	2C	5A	E3
E	0C	19	32	64	CC
F	0D	1A	35	6A	BF
G	0F	1E	3C	78	AA
A	10	21	42	85	9A
H	12	25	4A	95	88

Es lassen sich auch Zwischentöne programmieren. Diese Werte können Sie z.T. experimentell selbst ermitteln

Günter Böhm,
 Karlsruhe
 Tel.: [redacted]

SPRÜCHE

der mensch steht im mittelpunkt,
 und somit allem im wege.

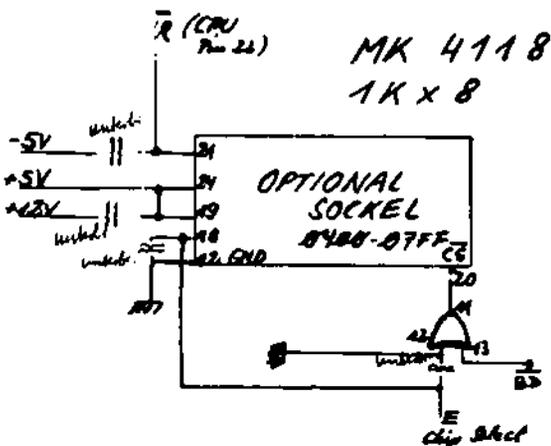
wo wir sind klappt nichts --
 aber wir koennen auch nicht ueberall sein.

wir wissen zwar nicht, was wir wollen --
 aber das mit ganzer kraft.

jeder macht was er will --
 keiner macht was er soll --
 aber alle machen mit.

BILLIGST-SPEICHERERWEITERUNG MIT MK 4118

Schon für ca. DM 70,- kann man den NASCOM 1 um 1k Speicherplatz erweitern, wenn man den freien EPROM-Sockel (IC 39) benutzt. Es sind nur sehr wenige Leitungsänderungen vorzunehmen. Installiert man entsprechende Umschalter, so ist das System auch weiterhin mit zusätzlichen EPROMs zu benutzen, man kann entsprechende Programme vor dem "Einbrennen" in EPROM sogar am "richtigen Platz" testen (0400 bis 07FF), was für nicht verschlebbliche Programme ein ungeheurer Vorteil ist. Aber auch bei fester Verdrahtung bietet ein zusätzlicher Bereich von 1k schon große Vorteile, vor allem für diejenigen, die sich zunächst vor der Ausgabe einiger Hundert Mark für eine normale Erweiterung scheuen. (Zumal kein verstärktes Netzteil notwendig ist!)



Am einfachsten lassen sich die notwendigen Anschlüsse folgendermaßen bewerkstelligen:

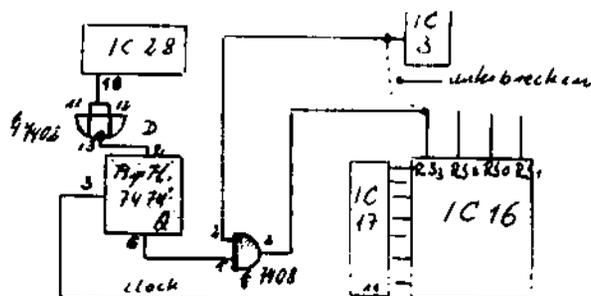
Zunächst nimmt man IC 44 aus der Fassung. Man biegt Pin 12 heraus und lötet ihn mit einer kurzen Leitung an Masse. Am MK 4118 (erhältlich bei MK-Systemtechnik) biegt man Pin 18, 19 und 21 heraus.

Am MK 4118 werden Pin 19 und 24 mit einem kurzen Drahtstück verbunden. Pin 18 verbindet man mit dem freigewordenen Anschluß am Sockel des IC 44. Nun muß nur noch Pin 21 mit dem \overline{WR} Ausgang der CPU verbunden werden, und der Zusatzspeicher ist einsatzbereit.

NASCOM 1 GRAPHIK "BRUTAL"

Um mit dem NASCOM 1 einigermaßen interessante Graphik zu erzeugen, ohne den Character Generator auszuwechseln zu müssen, ist es notwendig, a) die Zeilenabstände zu entfernen und b) die Punktmatrix zu verändern. Mit geringen Veränderungen an der Hardware ist das folgendermaßen möglich.

- a) Unterbricht man die Taktleitung von IC 3 zum RS 3 Eingang des Char. Generators und legt diesen auf Masse, so werden die Zeichen zweimal übereinander abgebildet, und die Leerzeile verschwindet. Man könnte hier einen manuellen Schalter einbauen; der Effekt läßt sich aber auch per Software steuern. Im Beispiel übernimmt diese Funktion das unbenutzte Bit 7 des VDU RAMs.



- b) Folgender Kniff verändert die Form der ASCII-Zeichen und bildet die verschiedensten graphischen Formen ab: Der LD-Takt von IC 18 zum Schieberegister IC 16 wird unterbrochen und durch den 4MHz Takt von Link 6 ersetzt. Anstelle eines Schalters kann auch hier durch Software gesteuert werden, z.B. mit Bit 5 von Port 0. Ein einfacher Befehl, wie 3E 00 oder 3E 20 D3 00, schaltet dann von gewohnten ASCII-Zeichen auf Graphik oder umgekehrt.

Günter Böhm,
Karlsruhe
Tel.: [REDACTED]


```

5 REM DIESES SPIEL IST BESONDERS GUT FUER
6 REM NAS - SYS GEEIGNET.
50 CLS
60 PRINT'SIMULATION EINES GELDSPIELAUTOMATEN'
70 PRINT'-----'
80 PRINT:PRINT:PRINT
100 PRINT'DIESES SPIEL SIMULIERT EINEN GELDSPIEL-'
102 PRINT'AUTOMATEN DER VERGANGENEN JAHRE'
105 INPUT'BITTE DEN HOECHSTBETRAG DES EINATZES FESTLEGEN:':H
108 CLS
110 PRINT:PRINT:PRINT:PRINT
120 PRINT' SIE KOENNEN NUN EINEN BETRAG VON'
125 PRINT' DM 1.- BIS DM'H'.- EINSETZEN.'
132 PRINT
140 FOR A=1 TO 2000
142 NEXT A
145 CLS
150 P=0
155 PRINT
160 INPUT'WIEVIEL DM WOLLEN SIE EINSETZEN':M
180 IF M>H THEN 860
185 PRINT
190 IF M<1 THEN 880
200 M=INT(M)
210 GOSUB 1270
230 X=INT(6*RNDRND(1)+1)
240 Y=INT(6*RNDRND(1)+1)
250 Z=INT(6*RNDRND(1)+1)
260 PRINT
270 IF X=1 THEN 910
280 IF X=2 THEN 930
290 IF X=3 THEN 950
300 IF X=4 THEN 970
310 IF X=5 THEN 990
320 IF X=6 THEN 1010
330 IF Y=1 THEN 1030
340 IF Y=2 THEN 1050
350 IF Y=3 THEN 1070
360 IF Y=4 THEN 1090
370 IF Y=5 THEN 1110
380 IF Y=6 THEN 1130
390 IF Z=1 THEN 1150
400 IF Z=2 THEN 1170
410 IF Z=3 THEN 1190
420 IF Z=4 THEN 1210
430 IF Z=5 THEN 1230
440 IF Z=6 THEN 1250
450 IF X=Y THEN 600
460 IF X=Z THEN 630
470 IF Y=Z THEN 650
480 PRINT:PRINT:PRINT
490 PRINT'SIE HABEN VERLOREN.'
500 P=P-M
510 PRINT'IHR STAND BETRAEGT IM MOMENT:P'DM'
515 PRINT
520 INPUT'NEUER START JA - NEIN':AD
540 IF AD='N' THEN 550
544 CLS
545 GOTO 160
550 PRINT

```

```

560 IF P=0 THEN 670
570 IF P=0 THEN 690
580 IF P)0 THEN 710
590 GOTO 1350
600 IF Y=Z THEN 730
610 IF Y=1 THEN 820
620 GOTO 1341
630 IF Z=1 THEN 820
640 GOTO 470
650 IF Z=1 THEN 820
660 GOTO 1341
670 PRINT'BISHER HATTEN SIE LEIDER NUR VERLUST'
680 GOTO 1350
690 PRINT'SIE HATTEN WEDER GEWINN NOCH VERLUST.'
700 GOTO 1350
710 PRINT'HOLEN SIE DEN GEWINN BEI DER KASSE AB.'
720 GOTO 1350
730 IF Z=1 THEN 780
740 PRINT:PRINT:PRINT'00 DEN ZEHNFACHEN BETRAG 00'
750 PRINT'GEWONNEN.'
760 P=(((100*M)+M)+P)
770 GOTO 510
780 PRINT:PRINT:PRINT'000 JACKPOT 000'
790 PRINT'GEWONNEN.'
800 P=(((1000*M)+M)+P)
810 GOTO 510
820 PRINT:PRINT:PRINT'0 FUENFFACHER BETRAG 0'
830 PRINT'IHR GEWINN.'
840 P=(((500*M)+M)+P)
850 GOTO 510
860 PRINT'DER HOECHSTEINSATZ BETRAEHT MAX.'H'.- DM.'
870 GOTO 160
880 PRINT'DER MINDESTESATZ BETRAEHT MIN. 1.- DM.'
890 GOTO 160
900 GOTO 220
910 PRINT' A ':GOSUB 1310
920 GOTO 330
930 PRINT' B ':GOSUB 1310
940 GOTO 330
950 PRINT' C ':GOSUB 1310
960 GOTO 330
970 PRINT' D ':GOSUB 1310
980 GOTO 330
990 PRINT' E ':GOSUB 1310
1000 GOTO 330
1010 PRINT' F ':GOSUB 1310
1020 GOTO 330
1030 PRINT' A ':GOSUB 1310
1040 GOTO 390
1050 PRINT' B ':GOSUB 1310
1060 GOTO 390
1070 PRINT' C ':GOSUB 1310
1080 GOTO 390
1090 PRINT' D ':GOSUB 1310
1100 GOTO 390
1110 PRINT' E ':GOSUB 1310
1120 GOTO 390
1130 PRINT' F ':GOSUB 1310
1140 GOTO 390
1150 PRINT' A ':GOSUB 1270

```

```

1160 GOTO 450
1170 PRINT ' B':GOSUB 1270
1180 GOTO 450
1190 PRINT ' C':GOSUB 1270
1200 GOTO 450
1210 PRINT ' D':GOSUB 1270
1220 GOTO 450
1230 PRINT ' E':GOSUB 1270
1240 GOTO 450
1250 PRINT ' F':GOSUB 1270
1260 GOTO 450
1270 FOR Q=1 TO 25
1280 PRINT CHR(42):
1290 NEXT Q
1300 RETURN
1310 FOR T=1 TO 5
1320 PRINT CHR(45):
1330 NEXT T
1340 RETURN
1341 PRINT:PRINT:PRINT'DOPPELTER GEWINN..'
1343 P=((20M)+M)+P
1344 GOTO 510
1350 FOR I=1 TO 3500
1360 NEXT
1370 PRINT'WOLLEN SIE NOCHMAL S SPIELEN?'
1380 INPUT'JA ODER NEIN':AD
1390 IF AD='N' THEN CLS:GOTO1395
1392 GOTO 50
1395 PRINT:PRINT:PRINT
1400 PRINT'TSCHUESS - AUFWIEDERSEHEN, KOMMEN SIE':
1410 PRINT' WIEDER.'
1420 GOTO 1490
OK

```

ACHTUNG:

=====

DIE FERN-SCHREIBMASCHINE KANN NICHT ALLE ZEICHEN WIEDERGEBEN DIE IN BASIC VORKOMMEN, DAFUER WERDEN AEBNLIICHE ODER ANDERE SYMBOLE VERWENDET.

HIER LIEGT DIE ZUSAMMENSTELLUNG DIE STETS ZU BEACHTEN IST:

BASIC - ZEICHEN

FERN-SCHREIBZEICHEN

MULTIPLIKATION (STERNCHEN).....	*
EXPONENT (PFEIL NACH OBEN).....	^
SPIITZE KLAMMER (GROSSER).....)
SPIITZE KLAMMER (KLEINER).....	(
STRICHPUKKT (GLEICH DOPPELPUKKT).....	:
STRINGZEICHEN (DOLLARZEICHEN).....	\$
ANFORDERUNGSZEICHEN ENTSPR. AUSLASSEZEICHEN:	'
PROZENTZEICHEN.....	%
ADDERUNGSZEICHEN ENTSPRICHT PUNKT.....	+

ALLE WENN NUR VORKOMMEN ZEICHEN IN BASIC SIND IDENTISCH.

Heinrich Auge

Künzelsau

PFEILSPIEL

von: Günter Brust

Baden-Baden

----- D A R T S -----

Unter Verwendung von NAS-SYS1. Startadresse :

ØD3Ø

0c80	21	00	08	11	01	08	01	bf	02	36	20	ed	b0	21	0a	08
0c90	11	40	00	3e	2f	06	06	c6	01	77	19	05	20	f9	06	05
0ca0	d6	01	77	19	05	20	f9	21	39	08	36	0d	22	2a	0d	3e
0cb0	80	32	26	0d	3e	ff	32	27	0d	3e	0b	32	28	0d	3a	29
0cc0	0d	fe	00	ca	f5	0c	3a	27	0d	3d	20	23	11	40	00	dd
0cd0	21	0a	08	fd	21	4a	08	dd	7e	00	06	0a	fd	4e	00	dd
0ce0	71	00	fd	19	dd	19	05	20	f3	32	8a	0a	3a	b5	0c	32
0cf0	27	0d	cd	3f	0e	3a	26	0d	3d	20	21	2a	2a	0d	3a	28
0d00	0d	11	40	00	36	20	3d	28	03	19	18	05	21	39	08	3e
0d10	0a	36	0d	22	2a	0d	32	28	0d	3a	b0	0c	32	26	0d	cd
0d20	3c	0e	c3	be	0c	00	5a	58	08	0f	f9	08	00	51	49	00
0d30	21	00	08	11	01	08	01	fe	03	36	20	ed	b0	21	c5	0e
0d40	11	dd	0e	0f	09	00	ed	b0	21	ce	0e	11	dc	08	0f	0b
0d50	00	ed	b0	cd	6d	0d	30	fb	fe	59	20	04	06	0f	18	06
0d60	fe	4e	20	ef	06	00	12	78	32	29	0d	18	0c	c5	d5	e5
0d70	00	df	6f	00	e1	d1	c1	c9	00	21	ef	0e	11	d7	0a	0f
0d80	15	00	ed	b0	21	04	0f	11	0c	0b	0f	08	00	ed	b0	21
0d90	0c	0f	11	2f	0b	0f	11	8c	0b	0f	07	00	ed	b0	cd	80
0da0	cd	ca	0d	21	14	0f	11	8c	0b	0f	11	af	0b	0f	07	00
0db0	0c	cd	6b	0e	cd	ca	0d	21	1b	0f	11	af	0b	0f	07	00
0dc0	ed	b0	cd	80	0c	cd	6b	0e	18	d6	11	4f	0b	21	2d	0d
0dd0	3e	30	ed	6f	12	13	ed	6f	12	ed	6f	11	72	0b	23	ed
0de0	6f	12	13	ed	6f	12	ed	6f	3a	2d	0d	fe	51	28	08	3a
0df0	2e	0d	fe	51	28	06	c9	cd	6f	0f	18	03	cd	74	0f	d5
0e00	21	22	0f	0f	07	00	ed	b0	cd	1a	0e	62	6b	23	0f	08
0e10	00	ed	b8	cd	1a	0e	d1	18	e6	00	0e	40	ff	00	00	cd
0e20	6d	0d	3e	04	0d	20	f5	c9	fe	52	20	05	e1	e1	c3	30
0e30	0d	fe	59	c0	e1	e1	c3	79	0d	00	00	00	ff	00	00	cd
0e40	6d	0d	d0	fe	52	20	05	e1	d1	c3	30	0d	21	92	0b	be
0e50	20	02	e1	c9	21	b5	0b	be	c0	e1	c9	00	00	df	62	00
0e60	fe	53	c8	fe	52	20	f6	e1	c3	30	0d	2a	2a	0d	0e	2e
0e70	36	20	2b	36	0d	06	02	ff	00	00	05	20	fa	0d	20	f0
0e80	2b	7e	e6	0f	f5	21	29	0f	01	07	00	fe	00	28	04	09
0e90	3d	20	fc	11	de	09	ed	b0	3a	92	0b	fe	51	20	05	21
0ea0	2d	0d	18	03	21	2e	0d	f1	86	27	fe	52	30	04	77	c3
0eb0	5b	0f	21	53	0f	11	de	09	01	08	00	ed	b0	c3	5b	0f
0ec0	00	00	00	00	00	2d	2d	44	41	52	54	53	2d	2d	48	61
0ed0	72	64	20	47	61	6d	65	3f	20	53	20	3d	20	53	74	61
0ee0	72	74	20	20	52	20	3d	20	52	65	53	74	61	72	74	50
0ef0	6c	61	79	69	6e	67	20	74	6f	20	35	31	20	45	78	61
0f00	63	74	6c	79	50	6c	61	79	65	72	20	31	50	6c	61	79
0f10	65	72	20	32	50	72	65	73	73	20	51	50	72	65	73	73
0f20	20	50	20	20	57	49	4e	53	20	52	55	42	42	49	53	48
0f30	4d	41	44	45	20	49	54	42	45	54	54	45	52	20	4e	4f
0f40	54	20	42	41	44	43	4c	4f	53	45	20	20	2d	53	48	4f
0f50	54	2d	20	54	4f	4f	20	4d	55	43	48	06	80	ff	00	00
0f60	05	20	fd	21	8a	0b	11	8b	0b	0f	30	00	ed	b0	c9	21
0f70	04	0f	18	03	21	0c	0f	11	db	09	0f	08	00	ed	b0	21
0f80	d9	0e	1f	57	0a	0f	16	00	ed	b0	1f	e3	09	c9	÷	

DRUCKAUSGABE FÜR DEN NASCOM 1
 =====

Im Folgenden beschreibe ich den Anschluß eines IBM-Kugelkopfdruckers 3982 an ein NASCOM-System.

Inwiefern das auf Kugelkopfschreibmaschinen Übertragbar ist, kann ich nicht sagen, es soll aber einen Umbausatz (Magnete) zum Nachrüsten geben. Auch sonst kenne ich mich mit den Dingen nicht aus und mußte das meiste durch Probieren ermitteln.

Die Bewegungen werden durch Magnete ausgelöst: für den Kugelkopf 7 Stück und je einer für M-O (?), rotes Farbband, schwarzes Farbband, SP(ace), BS (Backspace), CR (neue Zeile + Kopf zum Zeilenanfang), TAB(ulator), Index (?), UC (upper case = Großschreibung) und LC (lower case). Die Relais werden mit 48 V angetrieben und müssen für ca. 10 ms (CR 50 ms) angezogen bleiben, um den entsprechenden Vorgang auszulösen. Drucken eines Zeichens dauert ca. 60 ms, CR je nach Zeilenlänge.

7 Relais für den Kugelkopf sind natürlich redundant für die 44 Zeichen des Kugelkopfes (ohne Shift), man braucht aber leider alle. Ich habe den Kode durch Ausprobieren ermittelt und bei mehreren Möglichkeiten die ausgewählt, bei der weniger Relais aktiviert werden. Um trotzdem mit 8 Steuerleitungen auszukommen, wird mit Bit 7 zwischen druckbaren Zeichen und Sonderzeichen umgeschaltet. Dafür ist im Drucker ein Interface eingebaut mit 4 IC's CD 4001 und Darlington-Transistoren zur Ansteuerung der Relais (Freilaufdioden waren schon vorhanden). Die offenen Eingänge ziehen die 10k-Widerstände hoch, damit sind die Relais ausgeschaltet. Lo aktiviert die Eingänge. Damit beim Einschalten des Rechners der Drucker nicht losrattert, habe ich ein 8212 als O-Port benutzt und CLR an Reset gelegt. Jede Leitung wird durch 4049-Buffer noch mal invertiert und alles stimmt (man könnte auch ein PIO nehmen).

Am Besten wär's ja gewesen, den Drucker mit einer seriellen Schnittstelle aus-

zustatten, es entstünde jedoch ein ziemlicher Aufwand insbesondere wegen Klein-Großumschaltung. Ich habe stattdessen die Alternative eines Software-Treibers gewählt. Ein Unterprogramm (SELPRI), das relokatable ist, erfüllt folgende Aufgaben: es erkennt Sonderzeichen (SP, BS, NL) und führt sie aus. Druckbare Zeichen werden in den Seletrik (=Kugelkopf-)Kode umgewandelt. Dann wird überprüft, in welchem Shift-Zustand der Kopf steht und ggf. wird er gedreht, dann das Zeichen gedruckt. Bestimmte Zeichen, die der Kugelkopf nicht hat, die aber dringend nötig sind, wie z.B. \$, £, *, † (soll 'kleiner' bedeuten) und ‡ werden durch Doppeldruck dargestellt: 1. Zeichen, BS, 2. Zeichen.

Als Merker für den Shift-Zustand dient Bit 7 in ARGS von NASBUG. Wenn ein anderes Bit ungleich 0 ist, wird der Shift-Zustand und ggf. das PIO initialisiert. Da ARGS beim Starten des Hauptprogrammes benutzt wird, initialisiert das Programm den Drucker beim ersten Aufruf.

Zunächst hatte ich feste Zeitkonstanten programmiert, das gab aber bei NL unnötig lange Verzögerungen (man könnte natürlich auch einen Buchstabenzähler einbauen und die Zeit berücksichtigen, die der Kopf zur Rückreise braucht), aber man kann insgesamt etwas an Geschwindigkeit gewinnen, wenn eine Rückmeldung benutzt wird.

Das ist in der Maschine auch schon vorgesehen, es wird aber noch ein Eingang benötigt. Ich habe Bit 7 der Tastaturschnittstelle genommen.

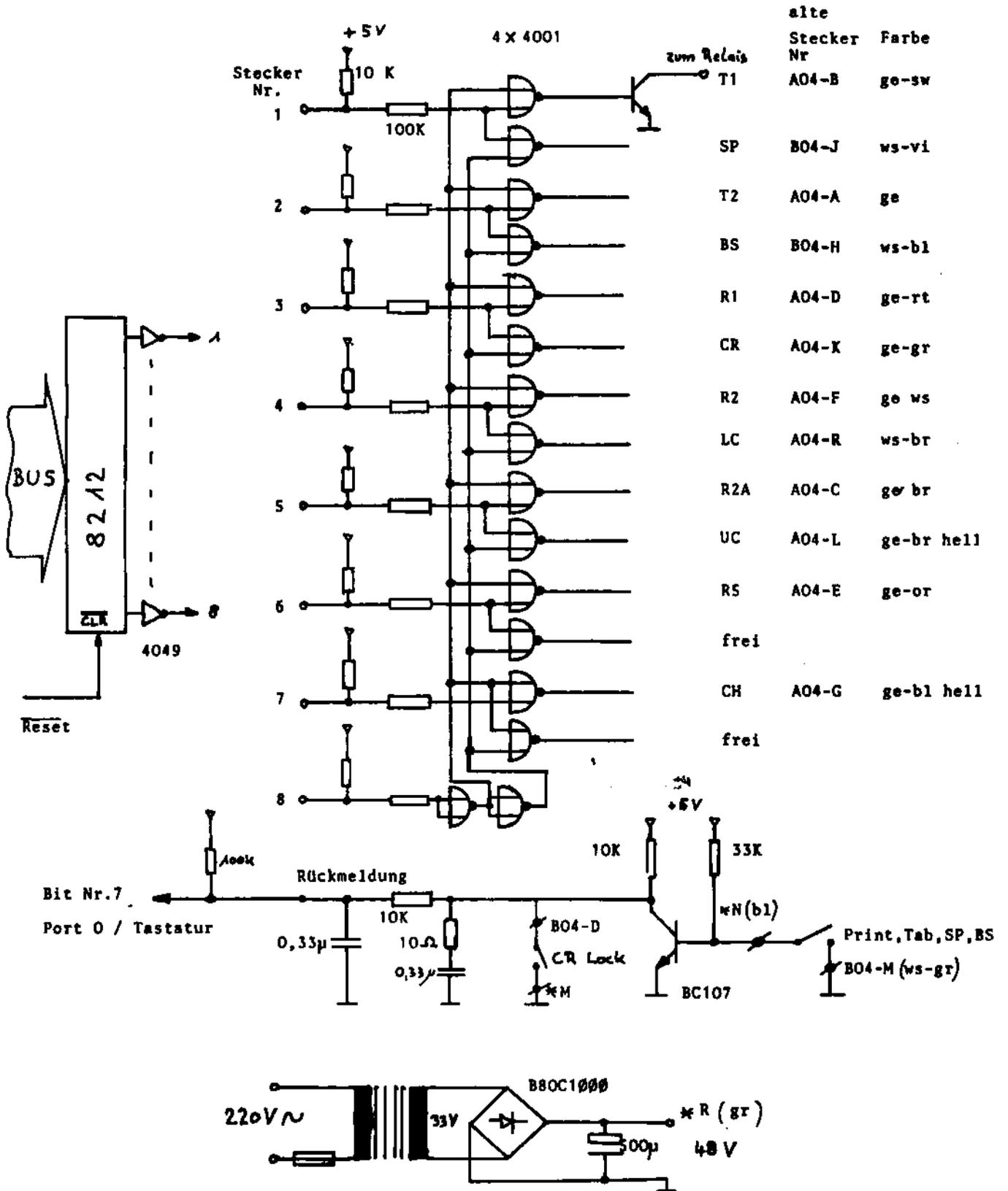
Näheres geht aus dem schwach kommentierten Quellcode hervor. Die Kode-Tabelle gilt nur für eine bestimmte Kugelkopftype, (Courier 12), es gibt leider mehrere Ausführungen davon.

Michael Bach

INTERFACE ZUR ANSTEUERUNG EINES KUGELKOPF-DRUCKERS

RECHNER

DRUCKER



EINFACH-DISASSEMBLER FÜR NASCOM
 =====

"Einfach" deshalb, weil der Disassembler nicht alle 280-Befehle erkennt. So fehlen z.B. die "Bit"-Befehle und die, welche mit den Index-Registern zu tun haben. Dafür kann das Programm relative Sprünge ausrechnen und druckt beim Auftauchen des Restart Befehls "EF" das zugehörige Datenstring aus. Mehr war beim besten Willen nicht auf einem Speicherplatz von 1k unterzubringen, zumal noch eine Subroutine enthalten ist, die ASCII in Baudot Code für Fernschreiber umsetzt.

Dieses Fernschreiberprogramm beginnt bei 0757 und läßt sich für Systeme, die keinen Fernschreiber über UART anschließen können leicht folgendermaßen umschreiben:

```
0757 CD 3B 01 C9 Call CRT Return
```

Zusätzlich muß eine Programmstelle umgeschrieben werden, die für den Fernschreiber Carriage Return und Line Feed ausgibt:

```
048D CD 40 02 Call "Scroll"
0490 00 00 00 00 NOP
0494 00 00 00
```

07F9 muß von C7 (Restart 0) auf 76 (Halt) geändert werden.

Damit ist der Disassembler für jedes NASCOM 1 System verwendbar.

Das Programm benötigt den Speicherbereich 0400 bis 07FF. Es kann in einem EPROM 2708 im freien EPROM Sockel gespeichert sein, oder man benutzt ein 1k Static RAM MK 4118 als Speichererweiterung, wie an anderer Stelle beschrieben. Gestartet wird das Programm mit E07E3. Dann drückt man die "A" Taste und gibt wie gewohnt Anfangs- und Endadresse des Programms ein, das disassembliert werden soll. Dann New Line. Vorsicht bei Bildschirmausgabe: Es können hier nur max. 15 Programmzeilen sichtbar gemacht werden. Erkennt das Programm einen Befehl nicht, wird bei TTY Betrieb ausgedruckt, auf dem Bildschirm erscheint das \$ Zeichen.

Hier als kleine Kostprobe den Anfang des Disassemblers, auf einem alten Lorenz Fernschreiber ausgegeben.

```
0400 push hl
0401 ld a,b
0402 call 04be
0405 ld a,c
0406 call 04be
0409 ld l,04
040b ld a,20
040d call 04cf
0410 dec l
0411 jp nz,040b
0414 ld hl,055a
```

Günter Böhm, XXXXXXXXXXXXXXXXXXXX
 Karlsruhe XXXX
 Tel.: XXXXXXXXXX

TIP - TIP
 =====

Löschen des RAM-Speichers (T2, T4, SYS)

nach dem Einschalten oder Reset: CC00
 C01 4400

Die 4400 gelten für 16k, beim N1 ohne Erweiterung: 400

Funktionsweise:

Da nach Reset ~~an~~ C00 00 steht, wird durch den Kipierbefehl der gesamte Speicherbereich auf 00 gesetzt. Anschließend führt der Monitor automatisch E 0000 aus.

Anwendung:

wichtig beim Arbeiten mit Assembler, wenn dort DEFS verwendet wird.

Der Assembler überspringt nämlich den mit DEFS ausgesparten Speicherbereich, d.h. falls dort vom Einschalten her noch zufällige Bytes stehen, so bleiben sie boshafterweise auch erhalten und führen später zum Absturz des assemblierten Programmes.

GRAPHIK-ERWEITERUNGSKARTE FÜR NASCOM I *****

Ab sofort ist eine Graphik-Erweiterungskarte für den NASCOM I erhältlich, die dem NASCOM-Benutzer eine Vielzahl neuer Möglichkeiten eröffnet:

- a) Schachgraphik (in Verb. mit dem Schachprogramm)
- b) allgemeine Graphik (ähnlich PET, M280 etc.)
- c) inverse Darstellung von Schrift und Graphik
- d) Benutzung der Set-Funktion aus dem Microsoft-Basic
- e) volle Kompatibilität mit NASCOM II möglich.

Die Hauptattraktion dabei ist natürlich die Schachgraphik, die zusammen mit dem überarbeiteten und verbesserten Schachprogramm das Spiel wesentlich übersichtlicher und interessanter macht. Mit dieser Erweiterung kann sich der NASCOM in Bezug auf Schach durchaus mit anderen Herstellern (auch höherer Preisklassen) messen. Demonstrationen werden in der nächsten Ausgabe des Journals vorgestellt.

Die allgemeine Graphik besteht aus Symbolen und Mustern, die man bei sehr vielen anderen Modellen auch finden kann. Da die Graphiksymbole in einem Standard-EPROM gespeichert sind, ist es sehr einfach, sich einen Graphik-Satz zusammenzustellen. Z.B. könnte man ganz spezielle Spiele programmieren, für die dann ein eigener Graphik-Satz zur Verfügung stünde, ganz nach Wunsch!

Eine zweite Möglichkeit ist durch eine umschaltbare Hardware realisiert worden. Die inverse Darstellung des kompletten Bildschirms oder auch einzelner Zeichen. Bisher war es für NASCOM I - Benutzer nicht möglich, die Set-Funktion aus dem Microsoft-Basic zu benutzen, mit der auf dem Bildschirm 4608 verschiedene Punkte einzeln gesetzt oder gelöscht werden können. Dies geschieht in einer 96 x 48 Matrix und ist besonders interessant für Anwender, die bevorzugt mit Basic programmieren wollen.

Nicht zuletzt ist es mit dieser Erweiterung auch möglich, Programme, die für den NASCOM II geschrieben wurden, auf dem NASCOM I laufen zu lassen (mit NASSYS).

Das Funktionsprinzip ist ganz einfach. Der Rechner benutzt intern einen 8 Bit breiten Datenbus. Die normalen Symbole des NASCOM werden im ASCII-Code angesprochen, der aber nur ein 7 Bit Code ist. Somit bleibt ein Bit zur freien Verwendung übrig.

Dieses Bit wird nun zur Unterscheidung benutzt, ob ein Graphik- oder ein Schriftsymbol auf dem Schirm erscheinen soll. Ist Bit 7, = 0, wird je nach Schalterstellung entweder der Bildschirm invers dargestellt, oder es erscheint ein Graphiksymbol.

Die Karte kann sowohl mit einem EPROM 2708, als auch mit einem EPROM 2716 bestückt werden. Für den Fall, daß ein 2708 Verwendung findet, läßt sich der EPROM-Inhalt gleichzeitig invers und normal ansprechen.

Die Erweiterungskarte hat das Format einer halben EUROPAKARTE und wird über vorbereitete Flachbandkabel an den NASCOM angeschlossen. Der Charactergenerator und das Schieberegister 74LS165 müssen hierzu auf die Erweiterungskarte umgesetzt werden. Die Flachbandkabel werden dann in die freigewordenen Sockel gesteckt. Zusätzlich müssen noch drei Drahtbrücken eingelötet werden. Die normale Funktion des NASCOM wird dadurch in keiner Weise beeinträchtigt.

Die Karte ist als Bausatz oder als Fertigungskarte lieferbar und zwar vorläufig mit folgenden verschiedenen EPROM-Typen:

- 1) NASCOM II kompatibel
- 2) mit Schachgraphik und Schachprogramm (2708)
- 3) mit Schachgraphik, Set-Funktion und Schachprogramm
- 4) Sonderwünsche nach Kundenspezifikation

- Flexibler Einbau durch konfektioniertes Flachbandkabel
- Wahlweiser Einsatz von 8K oder 16K- EPROM'S
- Invertieren der Charaktere auf dem Bildschirm
- Darstellung von eigenen Graphik - Entwürfen auf dem Bildschirm (Programmieren von EPROM'S erforderlich)
- Keine zusätzliche Stromversorgung notwendig
- Karte im Europa - Format

Der Charaktergenerator des NASCOM beinhaltet im wesentlichen Buchstaben. Setzt man an diese Stelle des NASCOM-Rechners (IC 16) eine zusätzliche Logikschaltung, sind zusätzliche Zeichen (Charaktere) auf einem anzuschließendem Bildschirm darstellbar.

AUFBAU- ANLEITUNG

Das gelieferte Material entspricht einem hohen Qualitätsstandard auf das wir Garantie geben. Dennoch muß man einige Hinweise beachten, um Schäden an Bauteilen zu vermeiden.

- 1) Bevor gelötet wird sind empfindliche Bauteile (z.B. der Charaktergenerator) aus ihren Fassungen zu nehmen, die Anschlußbeine möglichst nicht berühren.
 - 2) Lötunkte nicht überhitzen. (Braune Verfärbung der Leiterplatte)
 - 3) Nach dem Löten: Kontrolle benachbarter Lötunkte. Lötzinnbrücken sind zu entfernen.
- Beachtet man die wenigen Hinweise, kann eigentlich nichts den Erfolg beeinträchtigen.

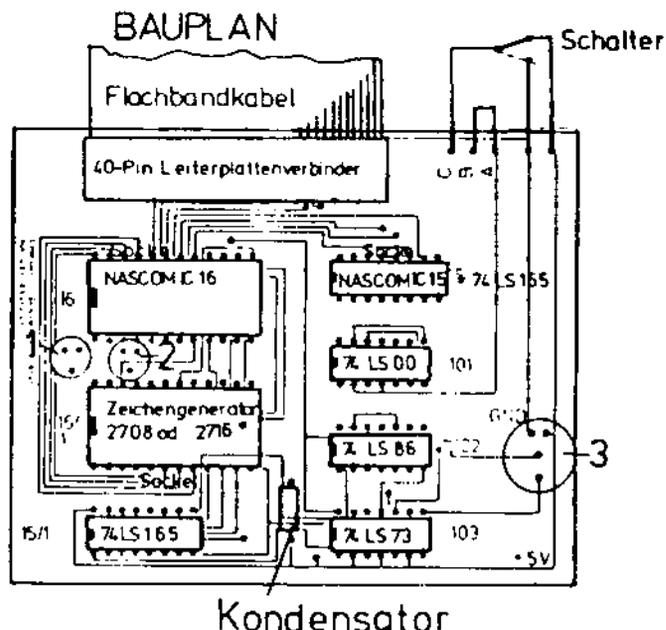
- Die Ic's 101; 102; 103; und 15/1 werden gemäß Bauplan eingelötet (Position beachten!)
- An die Stellen 15; 16; 16/1 werden Sockel eingelötet
- Das Flachbandkabel ist mit seinem 40- Pin Leiterplattenverbinder von oben auf die Extender- Karte zu drücken (siehe Bauplan), so kann man anschließend die 40 Pins auf der Rückseite verlöten.
- der Kondensator wird entsprechender Markierung auf dem Bauplan eingelötet.

Der Schalter.

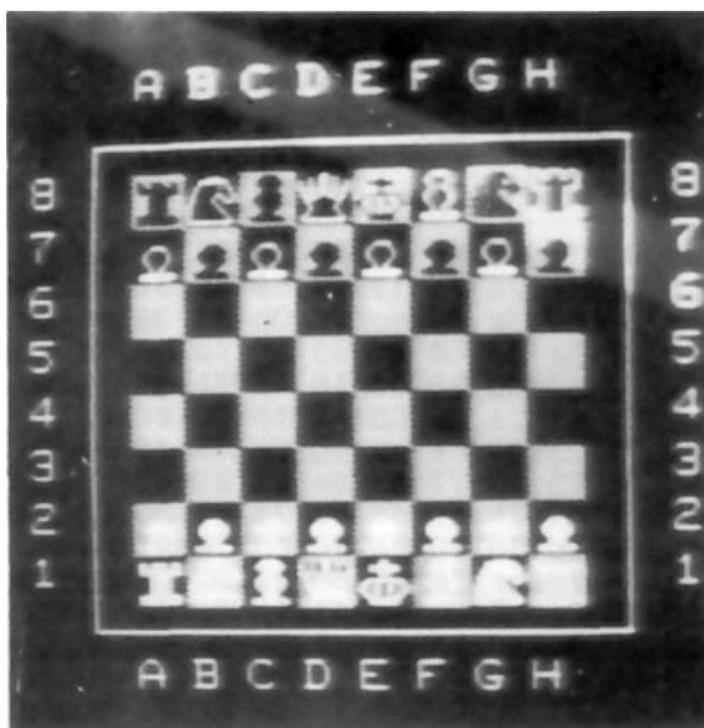
Durch ihn lassen sich die Buchstaben (kommend vom NASCOM- Generator) auf dem Bildschirm invertieren. Man sollte ihn dementsprechend leicht zugänglich montieren und der Bauplanzeichnung entsprechend anschließen. Dabei ist zu beachten: Zwischen A und B auf der Extenderkarte ist eine Brücke zu legen.

Desweiteren sind noch drei Brücken zu löten. Dazu ist von Bedeutung, welches EPROM als Erweiterungs-generator eingesetzt wird:

Einzusetzendes EPROM als Erweiterungs-generator	Drahtbrücken		
	1	2	3
8 k-PROM z.B.: INTEL 2708 oder andere (5V, 12V und -5V Versorgung)	V	V	I
16 k-PROM z.B.: INTEL 2716 TEXAS 2516 (Single 5V Versorgung)	/	/	I V II *



* entweder I: Die Charaktere, kommend aus dem Erweiterungs-generator werden nicht invertiert



Wolfgang Hentschel

Heidelberg-

Neue Schachversion unterstützt ROM-Grafik

Das beliebte 9k Schachprogramm ist inzwischen weiter verbessert worden. Es existieren jetzt sowohl eine NASBUG T4 - Version, als auch eine Version für NAS-SYS 1 (NASCOM 1 oder NASCOM 2).

Außer den reinen Programmverbesserungen enthält die neue Schachversion nun auch eine Grafik-Unterstützung. Mit einem zusätzlichen Zeichengenerator-ROM (bei MK-Systemtechnik erhältlich), kann man beim NASCOM 2 ohne jegliche Änderungen - nur durch Vertauschen des Zeichengenerator-ROM mit dem SCHACH-ROM - Schach mit voller Grafik laufen lassen (siehe Bild oben).

Für den NASCOM 1 ist noch die im vorhergehenden Text vorgestellte Grafik-Zusatzkarte erforderlich. Sie ist in einer Arbeitsstunde aufgebaut und wahlweise als Bausatz oder als Fertigergerät erhältlich.

Folgende Einführungspreise gelten bis zum 15. Januar 1981:

SCHACHPROGRAMM für NASCOM 1 oder NASCOM 2: Version "G" (mit Grafik)	DM 128.-
SCHACHPROGRAMM für NASCOM 1 oder NASCOM 2: Version "K" (ohne Grafik)	DM 98.-
Grafik-Zusatz-ROM (1k)	DM 78.-
Grafik-Zusatz-ROM (2k) (Schachgrafik und Klötzchengrafik).	DM 98.-
Grafik-Zusatzkarte NASCOM 1	DM 198.-

Alle Preise verstehen sich incl. MWSt.

Wir haben 25 Stück Grafik-Karten auf Lager. Greifen Sie zu, bevor die erste Serie vergriffen ist !


```

1 REM LEIDER ERLAUBT DAS NASCOM 8K - BASIC
2 REM ZWAR DAS ABSPEICHERN VON NUM. FELDERN
3 REM MIT DEM BEFEHL CSAVE*X, ABER NICHT DAS
4 REM DAS SPEICHERN VON STRINGFELDERN AUF
5 REM CASSETTE. MAN KANN JEDOCH DIE STRINGS
6 REM BEI EINGESCHALTETER SER. SCHNITTSTELLE
7 REM AUSGEREN UND SO MIT DEM RECORDER
8 REM AUFZEICHNEN. DIE EINGABE KANN DANN
9 REM MIT INPUT ERFOLGEN.
10 REM **** UNTERPROGRAMME ZUM SPEICHERN
11 REM VON STRING- FELDERN AUF CASSETTE
12 REM COPYRIGHT (C) 1980 PETER SZYMANSKI
13 REM ES WIRD DAS $- FELD B$
14 REM UND DAS NUM. FELD I
15 REM GESPEICHERT
16 REM BEDINGUNGEN:
17 REM DAS FELD I MUSS MIT DIMI(1)
18 REM DIMENSIONIERT WERDEN.
19 REM DAS FELD B$ MUSS NACH BEDARF
20 REM DIMENSIONIERT WERDEN.
21 REM BEI AUFRUF DES UNTERPROGRAMMS ZUM
22 REM SPEICHERN AUF BAND MUSS I DIE ANZAHL
23 REM DER ELEMENTE VON B$ ENTHALTEN.
24 REM DAS FOLGENDE PROGRAMM ZEIGT DIE
25 REM VERWENDUNG DER UNTERPROGRAMME.
26 REM IM UNTERPROGRAMM 2000 SIND EINIGE
27 REM WARTESCHLEIFEN EINGEFUEGT, DIE JE
28 REM NACH BAUD-RATE UND CPU-TAKT ANGE-
29 REM PASST WERDEN MUESSEN.
30 DIM I(1):CLEAR 2000:DIMB$(100)
31 B$="ABCDEFGHIJ":I=20
32 FORJ=0TOI:B$(J)=B$:NEXT:GOSUB 2000
33 PRINT"JETZT WIRD DAS BAND UEBERPRUEFT."
34 PRINT"BITTE SPULEN SIE ZURUECK!"
35 PRINT"DANN DRUECKEN SIE 'ENTER'."
36 INPUTUS:GOSUB8000
37 PRINT"NUN KOENNEN WIR DAS BAND WIEDER"
38 PRINT"EINLESEN. BITTE SPULEN SIE ZURUECK"
39 REM WENN IN ZEILE 620 DER BEFEHL 'I=0'
40 REM WEGGELASSEN WIRD, WERDEN DIE STRINGS
41 REM AN DAS SCHON BESTEHENDE FELD B$
42 REM ANGEHAENGT. DANN MUSS NATUERLICH
43 REM B$ ENTSPRECHEND DIMENSIONIERT SEIN
44 REM UND MIT CLEAR nnnn GENUEGEND $- SPEICHER
45 REM BEI DER INITIALISIERUNG ZUR VERFUEGUNG
46 REM GESTELLT WORDEN SEIN.
47 PRINT"UND DRUECKEN 'ENTER'."
48 I=0:GOSUB3000:END
49 REM SPEICHERN AUF BAND
50 PRINT:PRINT:PRINT"RECORDER STARTEN,"
51 PRINT"      DANN 'ENTER' DRUECKEN!"
52 INPUT US:I(1)=I:CSAVE*I
53 GOSUB10000:FORJ=1TO3000:NEXT
54 PRINT"FILE DEMO"
55 FORJ=1TO1000:NEXT:FORJ=0TOI
56 PRINTB$(J):FORA=1TO300:NEXTA
57 FORA=1TO500:NEXTA:PRINT"*****":FORA=1TO500

```

```
2110 NEXTA:NEXTJ:GOSUB10100:RETURN
3000 REM EINLESEN VOM BAND
3010 PRINT"RECORDER STARTEN!"
3020 CLOAD*I:GOSUB10000
3040 INPUTA$
3050 IFRIGHT$(A$,9)<>"FILE DEMO"THENGOTO3040
3060 PRINTRIGHT$(A$,9)" GEFUNDEN!"
3070 FORJ=ITOI(1)+I:PRINTJ$:INPUTB$(J)
3085 INPUTA$:IFRIGHT$(A$,2)="**"THEN3090
3086 GOTO3085
3090 NEXT:I=I+I(1)
3100 PRINT"FILE GELADEN ="I"ELEMENTE INSGESAMT!"
3110 GOSUB10100:RETURN
8000 REM UEBERPRUEFEN BAND
8010 PRINT"RECORDER STARTEN!"
8020 CLOAD*I:GOSUB10000
8040 INPUTA$
8050 IFRIGHT$(A$,9)<>"FILE DEMO"THENGOTO8040
8060 PRINTRIGHT$(A$,9)" GEFUNDEN!"
8070 FORJ=0TOI(1)
8080 INPUTB$
8085 INPUTA$:IFRIGHT$(A$,2)="**"THEN8090
8086 GOTO8085
8090 NEXT
8100 PRINT"FILE GEPRUEFT":GOSUB10100:RETURN
10000 REM EINSCHALTEN VON X0 (SER. SCHNITTST.)
10001 DOKE3189,1925:DOKE3187,1917:POKE3112,0
10010 RETURN
10100 REM ABSCHALTEN VON X0
10101 DOKE3189,1921:DOKE3187,1919:RETURN
```

```

10 REM **** R E V E R S E
20 REM **** COPYRIGHT (c) 1980 PETER SZYMANSKI
30 GOSUB500
40 CLS:PRINT"Brauchen Sie die Spielanleitung (J/N)";
50 INPUTA$:IFLEFT$(A$,1)<>"J"THEN190
60 PRINT"Das Spiel heisst 'REVERSE'. Auf dem Bild-"
70 PRINT"schirm wird eine Zahl angezeigt, deren"
80 PRINT"Ziffern in zufaelliger Reihenfolge sind."
90 PRINT"Sie muessen versuchen, die Ziffern wieder"
100 PRINT"zu ordnen. Es muss also wieder"
110 PRINT"                                12345...."
120 PRINT"angezeigt werden.Sie werden aufgefordert,"
130 PRINT"anzusehen, wieviele Stellen der Zahl (von"
140 PRINT"links) umgedreht werden sollen. Beispiel:"
150 PRINT"Die zu ordnende Zahl ist 3421. Sie geben"
160 PRINT"'2' ein (=4321) und dann '4', und schon"
170 PRINT"ist das Problem geloeset."
180 PRINT"Also dann...          VIEL SPASS!":PRINT
190 CLEAR200:REM **** SETZE STRINGBEREICH
200 PRINT"Mit wieviel Ziffern wollen sie spielen";
210 INPUTZ:IFZ>9ORZ<2THEN190
220 CLS:PRINT"          r e v e r s e / e s r e v e r"
230 REM **** ERZEUGEN DER LOESUNGSAHLE IN W$
240 DIMD$(Z):FORI=1TOZ:D$(I)=STR$(I)
250 W$=W$+RIGHT$(D$(I),1):NEXT
260 REM **** ERZEUGEN DER ANFANGSAHLE
270 A$="":FORI=1TOZ
280 A=INT(RND(1)*(Z+1)):IFD$(A)=" "THEN280
290 A$=A$+RIGHT$(D$(A),1):D$(A)=" ":NEXT
300 A=0:PRINTTAB(25)A$
310 INPUT"                REVERSE";I
320 REM **** UMDREHEN DER LINKEN I ZIFFERN
330 IFI<1ORI>ZTHEN310
340 A=A+1
350 B$=LEFT$(A$,I):A$=RIGHT$(A$,LEN(A$)-I)
360 C$="":FORJ=IT01STEP-1:C$=C$+MID$(B$,J,1)
370 NEXT:A$=C$+A$:PRINTTAB(25)A$
380 IFA$=W$THEN400:REM **** LOESUNG GEFUNDEN ?
390 GOTO310
400 PRINT"Sie haben die Loesung im"A"ten Versuch"
410 PRINT"                                gefunden!"
420 PRINT:PRINT"Wollen Sie es noch mal versuchen ";
430 INPUT"(J/N)";J$:IFLEFT$(J$,1)="J"THEN40
440 PRINT:PRINT:PRINT"Das ist aber schade."
450 PRINT"Es hat Spass gemacht, mit Ihnen zu";
460 PRINT" spielen."
470 PRINT:PRINT:PRINT:PRINT"                auf bald"
480 PRINT:PRINT:PRINT
490 END
500 REM **** SETZEN DES ZUFALLSAHLENGENERATORS
510 FORZ=0TO7:READY(Z):NEXT
520 DATA 237,95,6,0,205,242,240,201
530 Z=0:FORJ=3328TO3335:FOKEJ,Y(Z):Z=Z+1:NEXT
540 DOKE4100,3328:A=USR(0):Z=RND(-1/(A+1))
550 RETURN

```

```

0010 ,plotterprogramm
0020 ;
0030 ;
0040 ;
0050 ,im ram-bereich werden 48 speicher-
0060 ,plaetze von 0c50 bis 0c7f benoetigt
0070 ,(ram1 bis ram48)
0c4f 0080 ram equ h0c4f
cc00 0090 start org hcc00
cc00 211fcc 0100 ld hl,ramber
cc03 11580c 0110 ld de,9ram
cc06 010e00 0120 ld bc,14
cc09 edb0 0130 ldir
cc0b cd2dcc 0140 call zero
cc0e 3e0f 0150 ld a,h0f
cc10 d307 0160 out (07),a
cc12 3eff 0170 ld a,hff
cc14 d305 0180 out (05),a
cc16 21decf 0190 restar ld hl,ktab
cc19 22450c 0200 ld (h0c45),hl
cc1c c38602 0210 jp h286
cc1f 33 0220 ramber defb h33
cc20 33 0230 defb h33
cc21 23 0240 inc hl
cc22 c9 0250 ret
cc23 23 0260 inc hl
cc24 c9 0270 ret
cc25 cb06 0280 rlc (hl)
cc27 c9 0290 ret
cc28 cb06 0300 rlc (hl)
cc2a c9 0310 ret
cc2b 08 0320 defb h08
cc2c 90 0330 defb h90
cc2d af 0340 zero xor a
cc2e 21500c 0350 ld hl,1ram
cc31 0608 0360 ld b,8
cc33 77 0370 ld (hl),a
cc34 23 0380 inc hl
cc35 10fc 0390 djnz B-2
cc37 cd4ecd 0400 call anzeig
cc3a 2a540c 0410 dezanx ld hl,(5ram)
cc3d 11e00b 0420 ld de,h0be0
cc40 cdf5cc 0430 call bindez
cc43 2a560c 0440 ld hl,(7ram)
cc46 c3f5cc 0450 jz bindez
cc49 1a 0460 dezein ld a,(de)
cc4a fe20 0470 cp h20
cc4c 13 0480 inc de
cc4d 28fa 0490 jr z dezein
cc4f 1b 0500 dec de
cc50 1a 0510 ld a,(de)
cc51 fe2d 0520 cp '-'
cc53 2003 0530 jr nz B+5
cc55 13 0540 inc de
cc56 1807 0550 jr B+9
cc58 fe2b 0560 cp '+'
cc5a 2001 0570 jr nz B+3
cc5c 13 0580 inc de
cc5d 3e2b 0590 ld a,'+'
cc5f 326c0c 0600 ld (29ram),a

```

cc62	1a	0610		ld	a,(de)
cc63	fe20	0620		cp	h20
cc65	13	0630		inc	de
cc66	28fa	0640		jr	z \boxminus -4
cc68	1b	0650		dec	de
cc69	af	0660		xor	a
cc6a	21660c	0670		ld	hl,23ram
cc6d	0606	0680		ld	b,6
cc6f	77	0690		ld	(hl),a
cc70	23	0700		inc	hl
cc71	10fc	0710		djnz	\boxminus -2
cc73	1a	0720		ld	a,(de)
cc74	d630	0730		sub	h30
cc76	f8	0740		ret	m
cc77	fe0a	0750		cp	10
cc79	f0	0760		ret	p
cc7a	d5	0770		push	de
cc7b	216a0c	0780		ld	hl,27ram
cc7e	116b0c	0790		ld	de,28ram
cc81	010400	0800		ld	bc,4
cc84	edb8	0810		laddr	
cc86	12	0820		ld	(de),a
cc87	34	0830		inc	(hl)
cc88	d1	0840		pop	de
cc89	13	0850		inc	de
cc8a	18e7	0860		jr	\boxminus -23
cc8c	2600	0870	muladd	ld	h,0
cc8e	cd9acc	0880		call	mulrip
cc91	ed5b6d0c	0890		ld	de,(30ram)
cc95	19	0900		add	hl,de
cc96	226d0c	0910		ld	(30ram),hl
cc99	c9	0920		ret	
cc9a	f5	0930	mulrip	push	af
cc9b	c5	0940		push	bc
cc9c	0610	0950		ld	b,16
cc9e	4a	0960		ld	c,d
cc9f	7b	0970		ld	a,e
cca0	eb	0980		ex	de,hl
cca1	210000	0990		ld	hl,0
cca4	cb39	1000		srl	c
cca6	1f	1010		rra	
cca7	3001	1020		jr	nc \boxplus 3
cca9	19	1030		add	hl,de
ccaa	eb	1040		ex	de,hl
ccab	29	1050		add	hl,hl
ccac	eb	1060		ex	de,hl
ccad	10f5	1070		djnz	\boxminus -9
ccaf	c1	1080		pop	bc
ccb0	f1	1090		pop	af
ccb1	c9	1100		ret	
ccb2	d5	1110	dezbin	push	de
ccb3	2600	1120		ld	h,0
ccb5	3a670c	1130		ld	a,(24ram)
ccb8	6f	1140		ld	l,a
ccb9	226d0c	1150		ld	(30ram),hl
ccbc	3a680c	1160		ld	a,(25ram)
ccb1	6f	1170		ld	l,a
ccc0	110a00	1180		ld	de,10
ccc3	cd8ccc	1190		call	muladd
ccc6	3a690c	1200		ld	a,(26ram)

ccc9	6f	1210	ld	l,a
ccca	116400	1220	ld	de,100
cccd	cd8ccc	1230	call	muladd
ccd0	3a6a0c	1240	ld	a,(27ram)
ccd3	6f	1250	ld	l,a
ccd4	11e803	1260	ld	de,1000
ccd7	cd8ccc	1270	call	muladd
ccda	3a6b0c	1280	ld	a,(28ram)
ccdd	6f	1290	ld	l,a
ccde	111027	1300	ld	de 10000
cce1	cd8ccc	1310	call	muladd
cce4	d1	1320	pop	de
cce5	3a6c0c	1330	ld	a,(29ram)
cce8	fe2d	1340	cp	'-
ccea	c0	1350	ret	nz
cceb	2a6d0c	1360	ld	hl,(30ram)
ccee	cd28ce	1370	call	neghl
ccf1	226d0c	1380	ld	(30ram),hl
ccf4	c9	1390	ret	
ccf5	f5	1400	push	af
ccf6	c5	1410	push	bc
ccf7	e5	1420	push	hl
ccf8	d5	1430	push	de
ccf9	cb7c	1440	bit	7,h
ccfb	3e20	1450	ld	a,h20
ccfd	2805	1460	jr	z B+7
ccff	cd28ce	1470	call	neghl
cd02	3e2d	1480	ld	a,'-
cd04	12	1490	ld	(de),a
cd05	13	1500	inc	de
cd06	011027	1510	ld	bc,10000
cd09	cd41cd	1520	call	gdiv
cd0c	12	1530	ld	(de),a
cd0d	13	1540	inc	de
cd0e	01e803	1550	ld	bc,1000
cd11	cd41cd	1560	call	gdiv
cd14	12	1570	ld	(de),a
cd15	13	1580	inc	de
cd16	016400	1590	ld	bc,100
cd19	cd41cd	1600	call	gdiv
cd1c	12	1610	ld	(de),a
cd1d	13	1620	inc	de
cd1e	010a00	1630	ld	bc,10
cd21	cd41cd	1640	call	gdiv
cd24	12	1650	ld	(de),a
cd25	13	1660	inc	de
cd26	7d	1670	ld	a,l
cd27	c630	1680	add	a,h30
cd29	12	1690	ld	(de),a
cd2a	13	1700	inc	de
cd2b	13	1710	inc	de
cd2c	13	1720	inc	de
cd2d	d9	1730	exx	
cd2e	d1	1740	pop	de
cd2f	0604	1750	ld	b,4
cd31	13	1760	inc	de
cd32	1a	1770	ld	a,(de)
cd33	fe30	1780	cp	'0
cd35	2005	1790	jr	nz B+7
cd37	3e20	1800	ld	a,h20

bindez

cd39	12	1810		ld	(de),a
cd3a	10f5	1820		djnz	B-9
cd3c	d9	1830		exx	
cd3d	e1	1840		pop	hl
cd3e	c1	1850		pop	bc
cd3f	f1	1860		pop	af
cd40	c9	1870		ret	
cd41	af	1880	gdiv	xor	a
cd42	3c	1890		inc	a
cd43	b7	1900		or	a
cd44	ed42	1910		sbc	hl,bc
cd46	f242cd	1920		jp	p gdiv+1
cd49	09	1930		add	hl,bc
cd4a	3d	1940		dec	a
cd4b	c630	1950		add	a,h30
cd4d	c9	1960		ret	
cd4e	11d00b	1970	anzeig	ld	de,h0bd0
cd51	2a500c	1980		ld	hl,(1ram)
cd54	cd5bcd	1990		call	anz
cd57	2a520c	2000		ld	hl,(3ram)
cd5a	13	2010		inc	de
cd5b	7c	2020	anz	ld	a,h
cd5c	cd60cd	2030		call	ahex
cd5f	7d	2040		ld	a,l
cd60	f5	2050	ahex	push	af
cd61	1f	2060		rra	
cd62	1f	2070		rra	
cd63	1f	2080		rra	
cd64	1f	2090		rra	
cd65	cd69cd	2100		call	ahex1
cd68	f1	2110		pop	af
cd69	e60f	2120	ahex1	and	h0f
cd6b	c630	2130		add	a,h30
cd6d	fe3a	2140		cp	h3a
cd6f	3802	2150		jr	c B+4
cd71	c607	2160		add	a,7
cd73	12	2170		ld	(de),a
cd74	13	2180		inc	de
cd75	c9	2190		ret	
cd76	db05	2200	ausg	in	a,(05)
cd78	47	2210		ld	b,a
cd79	3ef0	2220	aus1	ld	a,hf0
cd7b	a0	2230		and	b
cd7c	47	2240		ld	b,a
cd7d	21580c	2250		ld	hl,9ram
cd80	3e03	2260		ld	a,h03
cd82	a6	2270		and	(hl)
cd83	b0	2280		or	b
cd84	47	2290		ld	b,a
cd85	23	2300		inc	hl
cd86	3e0c	2310		ld	a,h0c
cd88	a6	2320		and	(hl)
cd89	b0	2330		or	b
cd8a	d305	2340		out	(05),a
cd8c	3a640c	2350		ld	a,(2ram)
cd8f	f5	2360		push	af
cd90	f1	2370		pop	af
cd91	f5	2380		push	af
cd92	f1	2390		pop	af
cd93	3d	2400		dec	a

cd94	20f9	2410		jr	nz B-5
cd96	c9	2420		ret	
cd97	cba0	2430	heben	res	4,b
cd99	cd79cd	2440		call	aus1
cd9c	3e20	2450		ld	a,h20
cd9e	1807	2460		jr	B+9
cda0	cbe0	2470	senken	set	4,b
cda2	cd79cd	2480		call	aus1
cda5	3e40	2490		ld	a,h40
cda7	f5	2500		push	af
cda8	af	2510		xor	a
cda9	f5	2520		push	af
cdaa	f1	2530		pop	af
cdab	f5	2540		push	af
cdac	f1	2550		pop	af
cdad	3d	2560		dec	a
cdae	20f9	2570		jr	nz B-5
cdb0	f1	2580		pop	af
cdb1	3d	2590		dec	a
cdb2	20f3	2600		jr	nz B-11
cdb4	c9	2610		ret	
cdb5	21580c	2620	xricht	ld	hl,9ram
cdb8	cd5e0c	2630		call	15ram
cdbb	2a700c	2640		ld	hl,(33ram)
cdbe	2b	2650		dec	hl
cdbf	22700c	2660		ld	(33ram),hl
cdc2	2a500c	2670		ld	hl,(1ram)
cdc5	cd5a0c	2680		call	11ram
cdc8	22500c	2690		ld	(1ram),hl
cdcb	cd4ecd	2700		call	anzeig
cdce	c376cd	2710		jp	ausg
cdd1	21590c	2720	yricht	ld	hl,10ram
cdd4	cd610c	2730		call	18ram
cdd7	2a720c	2740		ld	hl,(35ram)
cdda	2b	2750		dec	hl
cddb	22720c	2760		ld	(35ram),hl
cdde	2a520c	2770		ld	hl,(3ram)
cde1	cd5c0c	2780		call	13ram
cde4	22520c	2790		ld	(3ram),hl
cde7	13e1	2800		jr	B-29
cde9	2a720c	2810	dx0	ld	hl,(35ram)
cdec	7c	2820		ld	a,h
cded	b5	2830		or	l
cdee	c8	2840		ret	z
cdef	cdd1cd	2850		call	yricht
cdf2	3a650c	2860		ld	a,(22ram)
cdf5	f5	2870		push	af
cdf6	f1	2880		pop	af
cdf7	3d	2890		dec	a
cdf8	20fb	2900		jr	nz B-3
cdfa	18ed	2910		jr	B-17
cdfc	2a700c	2920	dy0	ld	hl,(33ram)
cdfd	7c	2930		ld	a,h
ce00	b5	2940		or	l
ce01	c8	2950		ret	z
ce02	cdb5cd	2960		call	xricht
ce05	3a650c	2970		ld	a,(22ram)
ce08	f5	2980		push	af
ce09	f1	2990		pop	af
ce0a	3d	3000		dec	a

ce0b	20fb	3010		jr	nz B-3
ce0d	18ed	3020		jr	B-17
ce0f	cb7c	3030	abs	bit	7,h
ce11	200b	3040		jr	nz B+13
ce13	3e23	3050		ld	a,h23
ce15	325a0c	3060		ld	(11ram),a
ce18	3e06	3070		ld	a,h06
ce1a	325f0c	3080		ld	(16ram),a
ce1d	c9	3090		ret	
ce1e	3e2b	3100		ld	a,h2b
ce20	325a0c	3110		ld	(11ram),a
ce23	3e0e	3120		ld	a,h0e
ce25	325f0c	3130		ld	(16ram),a
ce28	7d	3140	neghl	ld	a,l
ce29	2f	3150		cpl	
ce2a	6f	3160		ld	l,a
ce2b	7c	3170		ld	a,h
ce2c	2f	3180		cpl	
ce2d	67	3190		ld	h,a
ce2e	23	3200		inc	hl
ce2f	c9	3210		ret	
ce30	061f	3220	mult	ld	b,31
ce32	210000	3230		ld	hl,0
ce35	227c0c	3240		ld	(45ram),hl
ce38	21780c	3250		ld	hl,41ram
ce3b	cd68ce	3260		call	shift
ce3e	3014	3270		jr	nc B+22
ce40	ed5b700c	3280		ld	de,(33ram)
ce44	2a7c0c	3290		ld	hl,(45ram)
ce47	19	3300		add	hl,de
ce48	227c0c	3310		ld	(45ram),hl
ce4b	3007	3320		jr	nc B+9
ce4d	2a7e0c	3330		ld	hl,(47ram)
ce50	23	3340		inc	hl
ce51	227e0c	3350		ld	(47ram),hl
ce54	217c0c	3360		ld	hl,45ram
ce57	cd68ce	3370		call	shift
ce5a	10dc	3380		djnz	B-34
ce5c	217d0c	3390		ld	hl,46ram
ce5f	cb7e	3400		bit	7,(hl)
ce61	2a7e0c	3410		ld	hl,(47ram)
ce64	2801	3420		jr	z B+3
ce66	23	3430		inc	hl
ce67	c9	3440		ret	
ce68	cb26	3450	shift	sla	(hl)
ce6a	23	3460		inc	hl
ce6b	cb16	3470		rl	(hl)
ce6d	23	3480		inc	hl
ce6e	cb16	3490		rl	(hl)
ce70	23	3500		inc	hl
ce71	cb16	3510		rl	(hl)
ce73	c9	3520		ret	
ce74	f5	3530	div	push	af
ce75	c5	3540		push	bc
ce76	af	3550		xor	a
ce77	32740c	3560		ld	(37ram),a
ce7a	32750c	3570		ld	(38ram),a
ce7d	0e11	3580		ld	c,17
ce7f	cb72	3590		bit	6,d
ce81	2007	3600		jr	nz B+9

ce83	cb23	3610		sla	e
ce85	cb12	3620		rl	d
ce87	0c	3630		inc	c
ce88	18f5	3640		jr	B-9
ce8a	b7	3650		or	a
ce8b	ed52	3660		sbc	hl,de
ce8d	f297ce	3670		jp	p pos
ce90	19	3680		add	hl,de
ce91	e5	3690		push	hl
ce92	21740c	3700		ld	hl,37ram
ce95	1806	3710		jr	B+8
ce97	e5	3720	pos	push	hl
ce98	21740c	3730		ld	hl,37ram
ce9b	cbc6	3740		set	0,(hl)
ce9d	0d	3750		dec	c
ce9e	2807	3760		jr	z B+9
cea0	cd68ce	3770		call	shift
cea3	e1	3780		pop	hl
cea4	29	3790		add	hl,hl
cea5	18e3	3800		jr	B-27
cea7	e1	3810		pop	hl
cea8	2a740c	3820		ld	hl,(37ram)
ceab	ed5b760c	3830		ld	de,(39ram)
ceaf	c1	3840		pop	bc
ceb0	f1	3850		pop	af
ceb1	c9	3860		ret	
ceb2	114c0b	3870	einanf	ld	de,h0b4c
ceb5	cd49cc	3880		call	dezein
ceb8	3a660c	3890		ld	a,(23ram)
cebb	b7	3900		or	a
cebc	c9	3910		ret	
cebd	cdb2ce	3920	rdplot	call	einanf
cec0	200b	3930		jr	nz B+13
cec2	2a540c	3940		ld	hl,(5ram)
cec5	22700c	3950		ld	(33ram),hl
cec8	2a560c	3960		ld	hl,(7ram)
cecb	1812	3970		jr	B+20
cecd	cdb2cc	3980		call	dezbin
ced0	22700c	3990		ld	(33ram),hl
ced3	22540c	4000		ld	(5ram),hl
ced6	cd49cc	4010		call	dezein
ced9	cdb2cc	4020		call	dezbin
cedc	22560c	4030		ld	(7ram),hl
cedf	22720c	4040		ld	(35ram),hl
cee2	c375cf	4050		jp	rbplot
cee5	cd28ce	4060	links	call	neghl
cee8	22700c	4070	rechts	ld	(33ram),hl
ceeb	22540c	4080		ld	(5ram),hl
ceee	210000	4090		ld	hl,0
cef1	18e9	4100		jr	B-21
cef3	cd28ce	4110	unten	call	neghl
cef6	22720c	4120	oben	ld	(35ram),hl
cef9	22560c	4130		ld	(7ram),hl
cefc	210000	4140		ld	hl,0
ceff	22540c	4150		ld	(5ram),hl
cf02	22700c	4160		ld	(33ram),hl
cf05	18db	4170		jr	B-35
cf07	cdb2ce	4180	xdrift	call	einanf
cf0a	2005	4190		jr	nz B+7
cf0c	2a540c	4200		ld	hl,(5ram)

cf0f	1806	4210		jr	B+8
cf11	cdb2cc	4220		call	dezbin
cf14	22540c	4230		ld	(5ram),hl
cf17	22700c	4240		ld	(33ram),hl
cf1a	ed5b500c	4250		ld	de,(1ram)
cf1e	eb	4260		ex	de,hl
cf1f	b7	4270		or	a
cf20	ed52	4280		sbc	hl,de
cf22	22500c	4290		ld	(1ram),hl
cf25	18c7	4300		jr	B-55
cf27	cdb2ce	4310	ydrift	call	einanf
cf2a	2005	4320		jr	nz B+7
cf2c	2a560c	4330		ld	hl,(7ram)
cf2f	1806	4340		jr	B+8
cf31	cdb2cc	4350		call	dezbin
cf34	22560c	4360		ld	(7ram),hl
cf37	22720c	4370		ld	(35ram),hl
cf3a	ed5b520c	4380		ld	de,(3ram)
cf3e	eb	4390		ex	de,hl
cf3f	b7	4400		or	a
cf40	ed52	4410		sbc	hl,de
cf42	22520c	4420		ld	(3ram),hl
cf45	18b5	4430		jr	B-73
cf47	cdb2ce	4440	adplot	call	einanf
cf4a	280f	4450		jr	z B+17
cf4c	cdb2cc	4460		call	dezbin
cf4f	22540c	4470		ld	(5ram),hl
cf52	cd49cc	4480		call	dezein
cf55	cdb2cc	4490		call	dezbin
cf58	22560c	4500		ld	(7ram),hl
cf5b	2a540c	4510	abplot	ld	hl,(5ram)
cf5e	ed5b500c	4520		ld	de,(1ram)
cf62	b7	4530		or	a
cf63	ed52	4540		sbc	hl,de
cf65	22700c	4550		ld	(33ram),hl
cf68	2a560c	4560		ld	hl,(7ram)
cf6b	ed5b520c	4570		ld	de,(3ram)
cf6f	b7	4580		or	a
cf70	ed52	4590		sbc	hl,de
cf72	22720c	4600		ld	(35ram),hl
cf75	2a720c	4610	rbplot	ld	hl,(35ram)
cf78	cd0fce	4620		call	abs
cf7b	3a5a0c	4630		ld	a,(11ram)
cf7e	325c0c	4640		ld	(13ram),a
cf81	3a5f0c	4650		ld	a,(16ram)
cf84	32620c	4660		ld	(19ram),a
cf87	22720c	4670		ld	(35ram),hl
cf8a	2a700c	4680		ld	hl,(33ram)
cf8d	cd0fce	4690		call	abs
cf90	22700c	4700		ld	(33ram),hl
cf93	cd3acc	4710		call	dezan
cf96	2a700c	4720		ld	hl,(33ram)
cf99	7c	4730		ld	a,h
cf9a	b5	4740		or	l
cf9b	cae9cd	4750		jp	z dx0
cf9e	2a720c	4760		ld	hl,(35ram)
cfa1	7c	4770		ld	a,h
cfa2	b5	4780		or	l
cfa3	cafccd	4790		jp	z dy0
cfa6	2a720c	4800		ld	hl,(35ram)

cfa9	ed5b700c	4810	ld	de,(33ram)
cfad	cd74ce	4820	call	div
cfb0	cdb5cd	4830	call	xricht
cfb3	21740c	4840	ld	hl,37ram
cfb6	11780c	4850	ld	de,41ram
cfb9	010400	4860	ld	bc,4
cfbc	edb0	4870	ldir	
cfbe	cd30ce	4880	call	mult
cfc1	ed5b720c	4890	ld	de,(35ram)
cfc5	b7	4900	or	a
cfc6	ed52	4910	sbc	hl,de
cfc8	2808	4920	jr	z @+10
cfca	f4b5cd	4930	call	p xricht
cfcd	fed1cd	4940	call	m yricht
cfdd	18e1	4950	jr	@-29
dfd2	21700c	4960	ld	hl,33ram
dfd5	7e	4970	ld	a,(hl)
dfd6	23	4980	inc	hl
dfd7	b6	4990	or	(hl)
dfd8	c8	5000	ret	z
dfd9	c4b5cd	5010	call	nz xricht
dfdc	18d5	5020	jr	@-41
dfde	59	5030	defb	'y
dfdf	27cf	5040	defw	ydrift
dfe1	58	5050	defb	'x
dfe2	07cf	5060	defw	xdrift
dfe4	48	5070	defb	'h
dfe5	97cd	5080	defw	heben
dfe7	53	5090	defb	's
dfe8	a0cd	5100	defw	senken
dfea	41	5110	defb	'a
dfeb	47cf	5120	defw	adplot
dfed	52	5130	defb	'r
dfee	bdce	5140	defw	rdplot
dff0	5a	5150	defb	'z
dff1	2dcc	5160	defw	zero
dff3	4d	5170	defb	'm
dff4	ad01	5180	defw	h01ad
dff6	54	5190	defb	't
dff7	fc01	5200	defw	h01fc
dff9	45	5210	defb	'e
dfffa	d002	5220	defw	h02d0
dfffc	00	5230	nop	
dfffd	3d00	5240	defw	h003d
dffff	00	5250	nop	

ktab

```

0010 ; SELPRI 15.06.80
0020 ; =====
0030 ;
0040 ;Softwaretreiber für Kugelkopf-Drucker
0050 ;-----
0060 ;
0070 ;Autor: Michael Bach, Stegen
0080 ;
0090 ;ASCII-Zeichen in A wird in SELECTRIC-
0100 ;Kode gewandelt und auf der Schreib-
0110 ;maschine mit Rückmeldung ausgegeben,
0120 ;einige Zeichen durch Doppeldruck,
0130 ;
0140 ;Länge des Programms genau 1/4 KByte.
0150 ;
0160 ;Das Programm ist relokativ!
0170 ;
0180 ;Betriebssystem: MASRUC T4,
0190 ;für MAS-SYS die Sonderzeichen ändern.
0200 ;
0210 RCAL EQU B10;relativer CALL
0220 RDEL EQU B88;Verzögerung
0230 ARG1 EQU B00A;Shiftmarker (ARG1)
0240 ;
0250 ORG B500
0260 SELPRI PUSH AF
0270 LD A,(ARG1)
0280 AND B7F
0290 OR A;Initialisierung fällig?
0300 JP Z SELPRI-1
0310 ;Initialisierung des Shift-Zustandes
0320 XOR A
0330 LD (ARG1),A
0340 LD A,B88;Unshift-Kode
0350 RST RCAL
0360 DEFB SELDHT-$-1
0370 POP AF
0380 ;Überprüfung des Kode-Bereiches
0390 SELPRI CP B7F;Zu gross?
0400 RET NC
0410 CP B1C;zu klein?
0420 RET C
0430 PUSH AF
0440 PUSH BC
0450 PUSH HL
0460 PUSH AF
0470 CP B21;Sonderzeichen?
0480 JR NC ZEICH
0490 LD HL,SOZAR-B10-$-5
0500 RST RCAL
0510 DEFB ACHLPC-$-1
0520 LD C,A
0530 LD B,0
0540 ADD HL,BC
0550 LD A,(HL)
0560 JR CSELOW
0E00
0E00 F5
0E01 3A0A0C
0E04 E67F
0E06 B7
0E07 2808
0E09 AF
0E0A 320A0C
0E0D 3E88
0E0F D7
0E10 5F
0E11 F1
0E12 FE7F
0E14 D0
0E15 FE1C
0E17 D8
0E18 F5
0E19 C5
0E1A E5
0E1B F5
0E1C FE21
0E1E 300C
0E20 215000
0E23 07
0E24 47
0E25 4F
0E26 0600
0E28 09
0E29 7E
0E2A 1821

```

```

oE2C 214400      o570 ;Normalzeichen
oE2F D7          o580 ZEICH LD HL,NOTAB-B21-S-5
oE30 3B          o590 RST RCAL
oE31 4F          o600 DEFB ADHLPC-S-1
oE32 0600        o610 LD C,A
oE34 09          o620 LD B,0
oE35 7E          o630 ADD HL,BC;Kode-Adresse
oE36 4F          o640 LD A,(HL)
oE37 E680        o650 LD C,A
oE39 210A0C      o660 AND B0;Maske für SHIFT-Bit
oE3C BE          o670 LD HL,ARG1
oE3D 280B        o680 CP (HL);SHIFT-Umschaltung nötig?
oE3F 77          o690 JR Z M14
oE40 3E88        o700 LD (HL),A
oE42 CB79        o710 LD A,B88;unshift-Kode
oE44 2802        o720 PIT 7,C
oE46 3E90        o730 JR Z M15
oE48 D7          o740 LD A,B90;shift-Kode
oE49 26          o750 M15 RST RCAL
oE4A 79          o760 DEFB SELOUT-S-1
oE4B 7BRF        o770 M14 LD A,C
oE4D D7          o780 RES 7,A
oE4E 21          o790 CSELOU RST RCAL
oE4F F1          o800 DEFB SELOUT-S-1
oE50 210F00      o810 ;Test auf Doppeldruck
oE53 07          o820 POP AF
oE54 17          o830 LD HL,NOTAB1-S-5
oE55 010C00      o840 RST RCAL
oE58 EDB1        o850 DEFB ADHLPC-S-1
oE5A 200C        o860 LD BC,6;6 Zeichen sind zu doppeln
oE5C 3E82        o870 CPIB
oE5E D7          o880 JR NZ ENDE;kein Doppeldruck
oE5F 10          o890 LD A,B82;vor 2. Zeichen Backspace
oE60 010500      o900 RST RCAL
oE63 09          o910 DEFB SELOUT-S-1
oE64 7E          o920 LD BC,NOTAB2-DOTAB1-1;2. Zeichen
oE65 F5          o930 ADD HL,BC
oE66 1BC4        o940 LD A,(HL)
oE68 E1          o950 PUSH AF
oE69 C1          o960 JR ZEICH
oE6A F1          o970 :
oE6B C9          o980 ENDE POP HL
oE6C C1          o990 POP BC
oE6D C5          1000 POP AF
oE6E 09          1010 RET
oE6F C9          1020 ;
                  1030 ;UP PC zu HL addieren
                  1040 ; damit Programm relokativel wird
oE6C C1          1050 ADHLPC POP BC
oE6D C5          1060 PUSH BC
oE6E 09          1070 ADD HL,BC
oE6F C9          1080 RET
                  1090 ;

```

```

1100 ;UP Ausgabe Zeichen in A
oE70 C5 1110 SELOUT PUSH BC
oE71 4F 1120 LD C,A
oE72 o600 1130 LD B,o
oE74 DB00 1140 M1 IN A,(o);vorheriges Zeichen fertig?
oE76 CB7F 1150 BIT 7,A
oE78 2004 1160 JP NZ M2
oE7A AF 1170 XOR A
oE7B FF 1180 RST REEL;Warten
oE7C 10F6 1190 DJNZ M1;maximal 255x7 mSec
oE7E 79 1200 M2 LD A,C
oE7F D308 1210 OUT (E),A
oE81 o600 1220 LD B,o
oE83 AF 1230 M3 XOR A
oE84 FF 1240 RST REEL
oE85 DB00 1250 IN A,(o);Druckvorgang begonnen?
oE87 CB7F 1260 BIT 7,A
oE89 2802 1270 JR Z M4
oE8B 10F6 1280 DJNZ M3;Warten bis es druckt
oE8D AF 1290 M4 XOR A
oE8F D308 1300 OUT (E),A;Ausgang löschen
oE90 C1 1310 POP BC
oE91 C9 1320 RET
1330 ;
1340 ;Sonderzeichen-Tabelle in SELECTRIC,
1350 ;Reihenfolge in ASCII: BS FF NL SP
oE92 02 1360 NOTAB DEFB 5E2 534 5P4 5P1;55 CR CR SP
1370 ;
1380 ;Normalzeichen-Tabelle
1390 ;" Ausrufez. $ % & %
oE96 00 1400 NOTAB DEFB 509 508 526 525 5A7 507
1410 ;( bis -
oE9C 34 1420 DEFB 534 538 537 53F 507 508 540 510
1430 ;/ bis 6
oEA4 87 1440 DEFB 5E7 525 53F 513 533 527 517 503
1450 ;7 bis grosser
oEAC 37 1460 DEFB 537 52B 543 5A4 597 598 533 597
1470 ;5 bis F
oEB4 A8 1480 DEFB 5A9 503 5A9 5C2 5AA 586 50C 588
1490 ;G bis N
oEBC BC 1500 DEFB 58C 59C 580 5A0 53A 5AC 580 50A
1510 ;0 bis V
oEC4 A5 1520 DEFB 525 504 588 505 505 5A2 53A 590
1530 ;W bis ^
oECC C1 1540 DEFB 5C1 53E 524 5A3 505 508 5A1 534
1550 ;' bis f
oED4 C0 1560 DEFB 5C0 534 520 542 52A 53C 51C 528
1570 ;o bis n
oE9C 3C 1580 DEFB 53C 506 508 520 50A 52C 537 51A
1590 ;o bis v
oEE4 25 1600 DEFB 525 514 508 535 505 522 53A 530
1610 ;w bis ^
oEEC 41 1620 DEFB 541 53E 504 523 515 518 521 524
1630 ;
1640 ;Donnelzeichen-Tabelle
oEF4 5E 1650 DOTAB1 DEFM " ^$%&("
oEF9 3E 1660 DEFB 53E
1670 ;Zweitzeichen-Tabelle
oEFA 60 1680 DOTAB2 DEFM " ^-1+--"

```

Listing des "Einfach" Disassemblers

```

0400 e5 78 cd be 04 79 cd be 04 2e 04 3e 20 cd cf 04
0410 2d c2 0b 04 21 5a 05 7e 23 b7 f2 17 04 0a ae 23
0420 a6 23 c2 17 04 7e e5 11 86 04 d5 e6 7f fe 20 d2
0430 cf 04 3d ca b0 04 3d ca b5 04 3d ca b1 04 3d ca
0440 a6 04 11 da 04 d5 11 07 03 21 fa 04 3d c8 21 15
0450 05 3d 00 2e 15 3d c8 2e 25 3d c8 2e 3f 3d c8 16
0460 09 3d c8 11 03 04 2e 4a 3d c8 2e 52 3d c8 00 00
0470 3d ca df 06 3d ca 16 07 3d 00 00 00 d1 0a 0f 0f
0480 0f e6 37 c3 cf 04 e1 7e 23 b7 f2 25 04 3e 0d cd
0490 cf 04 3e 0a cd cf 04 d1 d5 03 7a 2f 67 7b 2f 6f
04a0 09 d2 01 04 e1 c9 3e 20 cd cf 04 3e 20 c3 cf 04
04b0 03 0a c3 be 04 03 03 0a cd be 04 0b 0a 03 f5 1f
04c0 1f 1f 1f cd c7 04 f1 e6 0f c6 90 27 ce 40 27 e5
04d0 d5 c5 00 cd 57 07 c1 d1 e1 c9 0a 1f 15 c2 db 04
04e0 a3 3c 57 7e 23 b7 f2 e3 04 15 c2 e3 04 7e e6 7f
04f0 cd cf 04 7e 23 b7 f2 ed 04 c9 80 41 44 c4 41 44
0500 c3 53 55 c2 53 42 c3 41 4e c4 58 4f d2 4f 52 a0
0510 43 50 a0 00 00 a0 4e da 5a a0 4e c3 43 a0 50 cf
0520 50 c5 50 a0 4d a0 52 4c 43 c1 52 52 43 c1 52 4c
0530 c1 52 52 c1 44 41 c1 43 50 cc 53 43 c6 43 43 c6
0540 c2 c3 c4 c5 c8 cc 28 48 4c a9 c1 42 c3 44 c5 48
0550 cc 53 d0 42 c3 44 c5 48 cc 41 c6 00 ff 4e 4f d0
0560 01 cf 4c 44 20 04 0b 2c 82 03 cf 49 4e 43 04 8b
0570 09 cf 41 44 44 04 48 4c 2c 8b 0b cf 44 45 43 04
0580 8b 04 c7 49 4e 43 04 89 05 c7 44 45 43 04 89 06
0590 c7 4c 44 20 04 09 2c 81 07 c7 88 02 ff 4c 44 20
05a0 04 28 42 43 29 2c c1 12 ff 4c 44 20 04 28 44 45
05b0 29 2c c1 22 ff 4c 44 20 04 28 02 29 2c 48 cc 32
05c0 ff 4c 44 20 04 28 02 29 2c c1 0a ff 4c 44 20 04
05d0 41 2c 28 42 43 a9 1a ff 4c 44 20 04 41 2c 28 44
05e0 45 a9 2a ff 4c 44 20 04 48 4c 2c 28 02 a9 3a ff
05f0 4c 44 20 04 41 2c 28 02 a9 76 ff 48 41 4c d4 40
0600 c0 4c 44 20 04 09 2c 8a 80 c0 05 04 41 2c 8a c0
0610 c7 52 45 54 04 87 c2 c7 4a 50 20 04 07 2c 82 c6
0620 c7 05 04 41 2c 81 c4 c7 43 41 4c 4c 20 07 2c 82
0630 ef ff 50 52 49 4e 54 3a 8d c7 c7 52 53 54 04 90
0640 c1 cf 50 4f 50 04 8c c5 cf 50 55 53 48 20 8c c3
0650 ff 4a 50 20 04 82 d3 ff 4f 55 54 04 28 01 29 2c
0660 c1 e3 ff 45 58 20 04 28 53 50 29 2c 48 cc f3 ff
0670 44 c9 c9 ff 52 45 d4 e9 ff 4a 50 20 04 28 48 4c
0680 a9 f9 ff 4c 44 20 04 53 50 2c 48 cc db ff 49 4e
0690 20 04 41 2c 28 01 a9 eb ff 45 58 20 04 44 45 2c
06a0 48 cc fb ff 45 c9 cd ff 43 41 4c 4c 20 82 10 ff
06b0 44 4a 4e 5a 20 8e 18 ff 4a 52 20 04 8e 3e ff 4a
06c0 52 43 04 8e 30 ff 4a 52 4e 43 20 8e 28 ff 4a 52
06d0 5a 04 8e 20 ff 4a 52 4e 5a 20 8e 00 00 24 83 d1
06e0 16 10 3e 0d cd cf 04 3e 0a cd cf 04 1e 09 3e 20
06f0 1d 28 05 cd cf 04 18 f6 03 0a fe 00 28 08 cd cf
0700 04 15 28 dc 18 f2 3e 0d cd cf 04 3e 0a cd cf 04
0710 e1 e1 e1 c3 00 04 d1 03 0a b7 f5 f2 2a 07 3e 2d
0720 cd cf 04 ff ed 44 d6 02 18 08 3e 2b cd cf 04 ff
0730 c6 02 cd be 04 00 3e 20 cd cf 04 3e 48 cd cf 04
0740 3e 45 cd cf 04 3e 58 cd cf 04 d1 d1 c3 8d 04 00
0750 00 00 00 00 00 00 00 d9 57 e6 70 ca 77 07 d6 40
0760 fa 8a 07 78 b7 c2 71 07 3e 1f cd 5d 00 06 01 00
0770 00 21 61 07 c3 7a 07 21 d5 07 7a e6 7f 85 6f d2
0780 83 07 24 7e cd 5d 00 00 d9 c9 3e 20 ba ca 9c 07
0790 78 b7 ca 9c 07 3e 1b cd 5d 00 06 00 21 9c 07 c3
07a0 7a 07 03 19 0e 09 01 0d 1a 14 06 0b 0f 12 1c 0c
07b0 18 16 17 0a 05 10 07 1e 13 1d 15 11 04 1c 05 09
07c0 0d 04 1a 05 0f 12 04 11 0c 03 1c 1d 16 17 13 01
07d0 0a 10 15 07 06 18 0e 1e 04 1e 04 19 0b 04 04 02
07e0 02 04 08 21 ec 07 22 45 0c cd 86 02 41 ef 07 ed
07f0 4b 0c 0c 2a 0e 0c cd 00 04 c7 00 00 00 00 00

```

Sonderpreis bis
15 JANUAR : 128,-

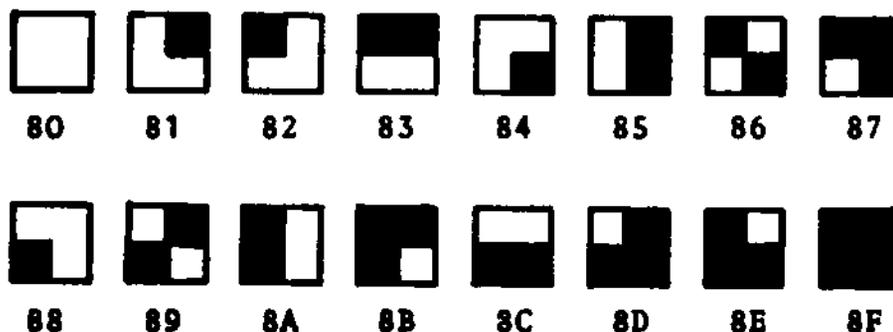
Michael Klein
Waldstrasse 20
6728 Germersheim
Tel.: 07274/2756

Graphik-Zusatzkarte für den NASCOM 1

Einbau der Karte in den NASCOM 1:

1. a) Draht von IC28(DM81LS97) Pin 18 an IC17(74LS273) Pin 18
b) Draht von IC17(74LS273) Pin 19 an IC16(Zeichengenerator) Pin 10
2. Austausch von IC15 (74LS165) gegen 16-poligen Stecker. Pin 1 beachten ! 16-poliger Stecker wird mitgeliefert.
3. Zeichengenerator ausbauen.
4. 24-poliges Kabel vom Graphik-Interface auf die Fassung des Zeichengenerators stecken. Pin 1 beachten ! Stecker darf keinesfalls verkehrt gesteckt werden.
5. Wenn Sie nun den NASCOM 1 einschalten, erscheint der Bildschirm hell.
6. Zeichengenerator auf 24-polige Fassung des Graphik-Interface stecken. Pin 1 beachten !
7. Bildschirm zeigt nach Einschalten (falls Ihr NASCOM 1 noch keinen Power-On-Reset hat) ein Gemisch aus Graphik und den gewohnten Zeichen.
8. Nach dem RESET kann man z.B. über den M-Befehl Zeichen direkt in den Video-Speicher schreiben. Dabei muß für Graphik das oberste Bit gesetzt sein. Graphik-Zeichen sind 80H...8FH ("H" bedeutet "hexadezimal").

Folgende Zeichen stehen zur Verfügung:



Minifloppy für NASCOM 1 und NASCOM 2
Ist jetzt erhältlich. Die seit langem angekündigte Floppy-Disk aus England ist leider noch nicht lieferbar, sodaß wir vor längerer Zeit zu einer Eigenentwicklung entschlossen haben.

Die Grundausstattung besteht aus einem Floppy-Disk Laufwerk (BASF Typ 6106) sowie einer Controller-Karte mit Bootstrap-loader. Sämtliche Steuersoftware ist auf Diskette gespeichert und umfaßt ein Floppy-Betriebssystem (DOS) sowie ein sehr leistungsfähiges BASIC. Im Preis eingeschlossen ist eine einjährige Software-Pflege. D.H.: Sie bekommen ein Jahr lang, vom Kaufdatum an gerechnet, immer die neueste Software nachgeliefert. Geplant sind u.a. ein leistungsfähiger MACRO-Assembler, eine kleines Textverarbeitungssystem sowie ein sog. "Supermonitor".

Der Hardware-Aufwand ist minimal, denn die Controller-Karte, die das übliche Format der NASCOM-Erweiterungskarten hat, ist fertig aufgebaut und getestet. Mit minimalen Hardware-Änderungen am System ist die Floppy nach ca. 30 min. betriebsbereit. Wer keinen Platz mehr im vorhandenen Gehäuse hat, kann auch die aufgebaute und getestete Version mit eigenem Gehäuse und Netzteil wählen. Das ist natürlich etwas teurer.

Bei Bestellung geben Sie bitte an, ob Sie die Floppy-Disk an einem NASCOM 1 oder an einem NASCOM 2 betreiben wollen. Versteht sich, daß eine ca. 200-seitige deutsche Beschreibung im Preis inbegriffen ist.

Floppy 1. Laufwerk mit Controller und Steuersoftware 1548.- + MWSt

Floppy 1. Laufwerk mit Controller und Steuersoftware sowie einem Floppy-Gehäuse, in dem Floppy-Laufwerk und Netzteil montiert sind 1898.- + MWSt.

2. Laufwerk 880.- + MWSt.

Eine Controller-Karte kann 2 Laufwerke bedienen. Bei den angebotenen Laufwerken handelt es sich um 5" Minifloppies mit einer Speicherkapazität von ca. 100 kByte.

PIO-Board

Auf der I/O-Karte sind Plätze für 3 PIO-Bausteine, einen UART und einen CTC-Zähler-Zeitgeber. Die Karte wird für 189.- + MWSt ohne alle Optionen geliefert, sodaß man sich für die jeweiligen Anwendungen ein System individuell zusammenstellen kann. Zu jeder Option werden auch die zugehörigen Stecker und Kabel gleich mitgeliefert.

EPR0M-Programmiergerät für NASCOM 1 und NASCOM 2

Das EPR0M-Programmiergerät ist nun schon bei einigen hundert Anwendern in Betrieb. Mittlerweile gibt es schon eine Version, die mit dem NASCOM 2 läuft.

Das Gerät wird über die PIO angesteuert und kann 2708 oder 2716 - EPROMs (1k x 8 bzw. 2k x 8) programmieren. Es können allerdings bislang nur Typen mit 3 Betriebsspannungen programmiert werden.

Die Steuerung des Programmierers erfolgt über ein EPROM, das voll relokaterbar ist. Es ist also gleichgültig, auf welche Adresse Sie die Steuersoftware legen, sie funktioniert immer.

Auch die Programmierspannung stellt kein Problem dar. Die - 26Volt werden aus der 12Volt-Versorgung des NASCOM mit einem einfachen Spannungswandler gewonnen.

Bausatz: 168.- + MWSt; Typ des Rechnersystemes bzw. Betriebssystem angeben.

Wandler auf der Karte !

Umschaltkarte zum Umschalten zwischen dem Betriebssystemen NAS-SYS 1 und NASBUG T4. Mancher Benutzer möchte die Vorteile nutzen, die NAS-SYS 1 für den Betrieb mit BASIC bietet (z.B. frei beweglicher Cursor), aber andererseits nicht auf den gewohnten NASBUG T4 verzichten, für den vielleicht schon zahlreiche Programme vorliegen. So bietet sich als Kompromiß die Umschaltkarte an. Auf diese Karte werden beide Betriebssysteme aufgesteckt. Von der Karte führen zwei Flachsteckverbinder in die Sockel, in denen bislang der NASBUG T4 steckte.

Platine, Bausatz mit allen Teilen einschließlich Flachsteckern: 68.- + MWSt

3k ASSEMBLER für NASCOM 1 und NASCOM 2

Der EPROM-Assembler ist mit ZEAP kompatibel. Das heißt, Sie können die schon vorhandenen Bänder mit symbolischen Code weiterverwenden. Zusätzlich hat der EPROM-Assembler noch einen erweiterten Editor, der z.B. mit einem verbesserten F-Befehl arbeitet. Noch einige andere Dinge sind verbessert worden. So kann es bei ZEAP passieren, daß man mit dem Assembler den Assembler selbst überschreibt und zerstört. Das kann z.B. geschehen, wenn man ein Programm assembliert, dessen Maschinencode bei 0CS0 beginnt, aber weiter als bis 0F00 reicht. Einerseits kann man vom symbolischen Code her nicht ohne weiteres auf einen Blick sagen, wie weit der Maschinencode in den Speicher hinein laufen wird, andererseits "merkt" ZEAP das ohnehin nicht und zerstört sich selbst.

Der EPROM-Assembler verhindert dies. Er schützt zusätzlich noch die Variablenfelder, die er im Speicher ablegt.

Der NASCOM-Assembler hat den großen Vorzug, daß man die vollen Editiermöglichkeiten nutzen kann, die NAS-SYS 1 bietet. Man kann mit dem Cursor überall auf dem Bildschirm herumlaufen, den Text verändern und als neue Eingabe verwenden.

Beide ASSEMBLER kommen mit deutscher Beschreibung. Bei Bestellung bitte Typ des Betriebssystemes angeben !
Jeder EPROM-Assembler 240.- + MWSt

2k Disassembler für NASCOM 1 und NASCOM 2

Der Disassembler dient dazu, Maschinenprogramme in Assemblerprogramm zurück zu übersetzen. Unser Disassembler kann sogar Bänder erzeugen, die man mit ZEAP oder unseren EPROM-Assemblern gleich weiterverarbeiten kann.

Den Assembler gibt es für NASCOM 1 auf Band mit den Startadressen 1000H und 4000H.

Für NASCOM 2 ist der Assembler auf EPROMs erhältlich.

NASCOM 1 - Disassembler 79.80 + MWSt
NASCOM 2 - Disassembler 120.00 + MWSt

kleinanzeigen

Konditionen: Die Spalte "Kleinanzeigen", die je nach Bedarf ggf. erweitert wird, steht allen NASCOM-Benutzern für Anzeigen mit bis zu 40 Worten kostenlos zur Verfügung. Was über 40 Worte hinausgeht, kostet 3.- pro Zeile. Dies gilt nicht für gewerbliche Kleinanzeigen, für die Sie bei Bedarf bitte eine Preisliste anfordern.

VERKAUFE

- CLD-Floppy in Veroframe und Tastatur in Pultgehäuse
- 15 Leerdisketten
Preis VB
Wolfgang Both
Tel.: [REDACTED]

- NASCOM 1 - Netzteil - Cassettenrecorder - Tiny Basic - 3 Gehäuse - I/O Anzeige - 16k ROM Karte 1k bestückt - Thermodrucker - T2 NAS SYS - Monitor - Buffer Board - Software.
gegen Höchstangebot abzugeben
Jürgen Weiermann
Tel.: [REDACTED]

- Maschinensprache (Cassette) - Assembler - Disassembler - Pfeilspiel -
- Basic - Hoch-Tiefpass-Berechnung Ratespiele - Mondlandung - Trafoberechnung (8k Speicher) - Schwingkreisberechnung u.v.a.
Klaus Bott, [REDACTED]
Tel.: [REDACTED]

- 8k-Mikrosoft-Basic auf EPROMs
- Double-Precision-Basic auf Floppy
- NASPEN - NASDIS - NASDEBUG -
- ZEAP 2.0 - Siemens -T100 Fernschreiber (Interface, Software)
Gerhard Baier, [REDACTED]
Tel.: [REDACTED]

- EPROM Typ 2708 DM 17,--
- EPROM Typ 4116 DM 13,--
Rüdiger Maurer, [REDACTED]

- Siemens T100 Fernschreiber mit Lochstr. Leser und Stanzer im Standgehäuse
DM 350,--
- 5-Kanal-Steifensender
DM 80,--
Bei Abholung beide Geräte zusammen DM 400,--
Hans Gnirck, [REDACTED], [REDACTED]
Tel.: [REDACTED]

- 300 Byte-Text-Editor für NASCOM1-Grundeinheit mit NAS-SYS. Speichert (auch auf Cassette) und druckt (ohne überflüssige Blanks) Zeile 1-12 des Bildschirms. (Mit Druck-Programm für Hofer-Drucker).
Beschreibung, Listing und Cassette gegen 15,-- DM auf Postscheckkonto :
Hmb 376273-203
Martin Riedel, [REDACTED], [REDACTED]
Tel.: [REDACTED]

- ASCII-Teletype (Olivetti TE 318) mit RS232 IV 24-Schnittstelle (Kompatibel mit NASCOM), 110 Baud;
Preis VB 800,-- DM
- Siemens TTY (T1005); ehem. Computerterminal; 75 Baud, Großbuchstaben; Lochstreifenleser und Stanzer, Tischgehäuse; autom. Endabschaltung
Preis: VB 500,-- DM
H. Mollé, [REDACTED], [REDACTED]
Tel.: [REDACTED]

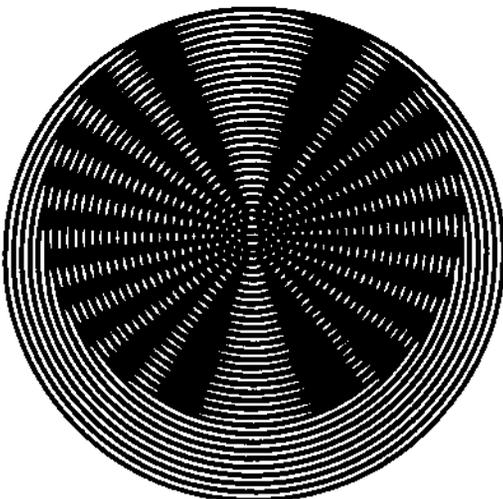
- CLD-FLOPPY mit Gehäuse, Netzteil und einem Laufwerk ; Alter ca. 1 1/2 Monate.

Preis: DM 1.980,--

Uwe Kühn, [REDACTED], [REDACTED]
Tel.: [REDACTED]

- Fernschreiber Siemens T 100 S (Automatik)
Preis: DM 395,--
- Lorenz Lo 133 (Automatik)
Preis DM 395,--
- Nixdorf Drucker, 2 Druckköpfe, 330 Zeichen/sec., Centronicsschnittstelle 123 Zeichen Schreibbreite, Vor-Rückwärtsdruck, interner Zeichenpuffer
Preis: nur DM 1.980,--
- Gehäuse für NASCOM 1 mit 3-4 Erweiterungskarten. Platz genug für Netzteil. Einige Buchsen sind bereits eingebaut.
Preis: DM 35,--
- Superschneller Lochstreifenleser 1500 Zeichen/sec. Volle Dokumentation
Preis: DM 980,--

M K - Systemtechnik
Waldstr. 20
6728 Germersheim



Erratum zum Artikel "Umlaute"
(NASCOM-Journal 5/80, S. 7):

Alpha-Shift für X00: Adr. #E0 ändern,
Keyboard-Table ab #62F abändern.
= E (kennzeichnet Hexa-Zahl).

SUCHE + BIETE

- Spiel- und technische Berechnungsprogramme in Assembler oder BASIC.
- Programmierer 2708 nach Muster oder Cassette zu DM 25,--
nach Listing zu DM 35,-- (incl. EPROM)
Rüdiger Maurer, [REDACTED], [REDACTED]

- Fortran für das NASCOM-MKS-Floppy
- Schach / Spiele
Gerhard Baier, [REDACTED]
Tel.: [REDACTED]

- Programme aller Art für NASCOM 2
Klaus Bott, [REDACTED], [REDACTED]

machen sie einen drucker aus ihrer
elektrischen schreibmaschine !!!

Unser Interface wird mit einem Handgriff von oben auf die Tastatur aufgesetzt und ist ebenso schnell wieder abgenommen. Dadurch ist kein Einriff in die Maschine notwendig, und sie kann jederzeit auch ganz normal verwendet werden.

Für den Aufbau des Interface-Bausatzes brauchen Sie einen Lötkolben und etwa vier Stunden Zeit.

Preis: DM 390,-- + Porto + MWST

Kanis GmbH,
Lindenberg 113
8134 Pöcking