

# **nascocom** **journal**

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

---

2. Jahrgang · Dezember 1981 · Ausgabe 11/12

---

**Herausgeber:**

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

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

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

---

**INHALT**

02 NASCOM Journal Intern	
03 FORTH für den NASCOM Teil 4	Günter Kreidl
06 Seite(n) für Einsteiger Teil 1	Günter Böhm
09 FSK-Modem	Günter Böhm
10 NASSYS 3	
11 Software Repeat	Christoph Rau
12 2716 Eprommer	Rüdiger Maurer
15 Leserbriefe	
17 DEBUG und TOOLKIT	Günter Böhm
18 WAIT-Zyklen	Stefan Bürger
19 STAR TREK	Markus Caesar
22 System-Grundlagen	Josef Zeller
24 GALAXIS	Jürgen Weiermann
25 Selbstbau-Plotter	Günter Böhm
27 Sortieren in BASIC Teil 5	W. Mayer-Gürr
28 Tastatur-Erweiterung	Ralph Cramer
29 16-Kanal A/D-Wandler	Peter Bentz
32 Monitor-Umschaltkarte	Dieter Oberle
34 Kleinbuchstaben für T2	Jürgen Weiermann
35 MDCR-Monitor Teil 2	Johannes C. Lotter
39 Microsoft für CLD	Günter Endert
Bildschirm auf Cassette	Karl Trust
40 CONMOD	Jürgen Weiermann
41 LIFE	Karl Trust
KANSAS-CITY	Michael Bach
42 PREISAUSSCHREIBEN	Red.
43 Cassetten-Interface	D. Maisl
46 Programme aus der Luft	Mary Jo Kostya
47 Hochaufl. Graphic Teil 2	H. Martin Pohl
48 Bestellservice	Red.
49 NASCOM PIO-BUS	Red.
Disassembler-Anpassung	Jürgen Weiermann
50 QUEST	Michael Bach
53 Kleinanzeigen	
INFO-Ecke	
54 NASCOMPL	
Impressum	
55 MKS Angebote	

# NASCOM journal

## intern

Liebe Leser,  
das erste Doppelheft des Redaktionsteams ist umfangreicher geworden als wir jemals zu hoffen wagten. Sicher ist auch für Sie etwas dabei, was die Zeit "zwischen den Jahren" schneller verstreichen läßt. Deshalb auch mehr Spiele als gewöhnlich.

Viele Leser haben sich schon um unseren Standard-Monitor NASSYS3 bemüht und Kommentare dazu geschrieben. Alles, was die Funktion und Anwendung dieses Monitors betrifft, soll in der entsprechenden Spalte regelmäßig veröffentlicht werden. Auch einige Programme wurden schon auf NASSYS3 umgeschrieben. Hier wären wir für weitere Beiträge sehr dankbar. (Suchen Sie einmal Ihre T2/T4 Programme durch, ob nicht eines das Umschreiben lohnt. Auch alte Journal-Programme wären geeignet.)

Eine weitere Neuerung zum Service für die Leser stellen die Spalten INFO-ECKE und BESTELLSERVICE dar. Besonders letztere stellt eine große Hilfe für die Standardisierung unserer Systeme und Erweiterungen dar, was einen Programmtausch unter den Lesern (und der Zeitschrift) erleichtert.

Ein Handicap in unserer gemeinsamen Arbeit war bisher allerdings häufig das unterschiedliche Cassetten-Format von NASCOM 1 und 2 und die Anfälligkeit des NI-Formates, die das Einlesen auf verschiedenen Rechnern erschwerte. Hier sollten wir uns auch zu einem Standard durchringen: Das Kansas-City-Format wäre eine optimale Lösung, auch in Verbindung mit anderen Rechner-Systemen. Deshalb hier gleich drei Vorschläge für den Bau eines entsprechenden Interfaces, z.T. sogar mit Platinenlieferung. Der Artikel von Herrn Bach sollte besonders im Hinblick auf die Umschaltmöglichkeiten beachtet werden. Seine Schaltskizze konnten wir leider nicht

mehr aufnehmen, bei Interesse drucken wir sie aber auf jeden Fall im nächsten Heft ab. Was das Einlesen mit dem herkömmlichen NASCOM1-Format angeht, ein Verstellen des Tonkopfes hat hier oft schon Wunder gewirkt!

Hier nun noch einige Anliegen der Redaktion:

Bitte senden Sie Kleinanzeigen nur auf der beiliegenden Karte ein. Wenn sie in einem Brief "versteckt" werden, kann es vorkommen, daß sie vergessen werden.

Herr Mayer-Gürr bittet um Adressen von Floppy-Besitzern. Er möchte den Programmaustausch etwas anregen. Beim Cassettenaustausch habe ich den Eindruck, daß sich potentielle Interessenten scheuen mitzumachen, weil sie keine eigenen Programme besitzen. Der Tausch hat aber die Funktion, sich die mühselige Tipperlei zu sparen, und so sind alle Programme willkommen, egal wo sie herkommen. Es gibt hier keine Copyright-Frage, weil die Aktion mehr oder weniger privat und absolut unkommerziell ist. Also...

Beachten Sie bitte die neue Adresse von Herrn Zeller (Impressum).

Häufig wird gefragt, was es neues bei LUCAS gäbe. Nun, im Augenblick eine Menge Versprechungen. So soll eine Farbgraphic-Karte auf den Markt kommen, ein Speech-Board (sprechender Computer), ein CP/M Controller und eine Menge an Software auf Floppy und Cassette. Eigentlich sollte das alles noch dieses Jahr verfügbar sein, aber auch MKS weiß inzwischen noch nichts Neues. Auf der Hobbytronic in Stuttgart war auch vom angekündigten NASCOM3 nichts zu sehen. Aufgefallen ist dort allerdings (vor allem wegen der Größe) das neue Gehäuse für Tastatur und Grundsystem mit einigen Erweiterungen und der Aufsatz für zwei Floppy-Laufwerke. (Soll es auch mit kleinem Bildschirm geben!) Warten wir ab, wann sich LUCAS endlich hier auf dem Kontinent mit den neuen Produkten meldet. Ich für meinen Teil habe noch mit meiner alten Ausrüstung genug zu tun.

Das wär's nun für dieses Jahr. Ich hoffe auf weitere gute Zusammenarbeit auch im dritten Jahrgang des NASCOM Journals und wünsche allen Lesern weiterhin viel Freude mit dem NASCOM und dem Journal

Ihr Günter Böhm



# FORTH für den NASCOM

## Teil 4 von Günter Kreidl

Zunächst einige Fehlerkorrekturen zum letzten Beitrag: Die als "-" bezeichnete Funktion muß natürlich "'=' heißen (da 'hab' ich einfach gepennt!). Bei der Beschreibung der Funktion FILL muß es heißen: "an die Adresse (T-2)". Außerdem ist im Druck bei der Funktion MCODE das abschließende Semicolon verloren gegangen. Nun aber zu den neuen Funktionen:

```
: SPACE 32 COUT ;
: SPACES 1 FOR SPACE LOOP ;
: CLS 12 COUT ;
```

SPACES erwartet die gewünschte Anzahl der Leerzeichen als Argument auf dem Stack, sie kann z.B. so aufgerufen werden "10 SPACES".

```
3113 CONSTANT CURSADR
```

```
: LINE      64 * 1994 + CURSADR POKEW ;
: POSITION    DEC CURSADR MEM+ ;
: SCREEN    SWAP LINE POSITION ;
```

Die ersten beiden Funktionen erwarten je einen Wert auf dem Stack und setzen den Cursor in die entsprechende Zeile bzw. Position einer Zeile. SCREEN kombiniert diese beiden Funktionen und erwartet deshalb zwei Werte: "(Zeile) (Position) SCREEN".

Einige weitere recht triviale Funktionen:

```
: '          SCAN  IF FIRST POP PEEKB
                ELSE ERROR TYPE THEN ;
: PRINTHEX  3084 POKEW 102 NAS-SYS ;
: =         CONBXA PRINTS ;
: .         DUP = ;
: BACKSPACE 8 COUT ;
```

Die Funktion "'" gibt den dezimalen Wert eines ASCII-Zeichens auf den Stack. Folgendes Beispiel illustriert mehrere Funktionen gleichzeitig:

```
' S . 2 SPACES PRINTHEX
```

83 0053 (Antwort des Interpreters)

Die Tastaturabfrage TERMINAL? muß wieder in Maschinencode programmiert werden. Sie gibt 0 auf den Stack, wenn keine Eingabe von der Tastatur erfolgt ist, sonst den Wert des Zeichens und -1 (=FFFFH) :

```
MCODE TERMINAL?
```

```
XXXX DF 62 21 00 00 30 07 6F
```

```
XXXX 26 00 E5 21 FF FF E5.
```

Die Adressen XXXX werden vom Monitor angegeben. Die folgende Funktion druckt das jeweils gewünschte "Dictionary" aus, dessen Startadresse auf dem Stack übergeben wird

```
(z.B. mit NAMES PEEKW für den Interpreter).
PRINTDIC REPEAT DUP FIRST DUP EQZ UNTIL
          + SWAP PRINTS 2 SPACES
          INC INC          LOOP
          CLEAR ;
```

In der Grundversion belegt der Interpreter 4 KB Speicher, wobei das Dictionary am oberen Ende des Speicherbereichs liegt. Es ist nun sehr leicht möglich und mit den gezeigten Erweiterungen auch bald nötig, das Dictionary zu verschieben. Die obere und untere Grenze sind durch die Variablen MEMORY und NAMES gegeben, und die Verschiebung wird mit der Funktion MOVEBYTES ausgeführt.

Anschließend müssen die Variablen NAMES und MEMORY natürlich angepaßt werden und ebenfalls eine Variable CDICT, die aber nicht im Dictionary enthalten ist und deshalb absolut eingegeben werden muß (bei 1297H bzw. 4759 dezimal). Will man nun das erweiterte Programm auf Cassette speichern, so müssen zwei getrennte Blöcke gespeichert werden. Das geschieht mit der folgenden Funktion:

```
: WRITEM 4096 3084 POKEW CODEADR PEEKW
          3086 POKEW 87 NAS-SYS
          NAMES PEEKW 3084 POKEW MEMORY
          PEEKW INC 3086 POKEW 87 NAS-SYS ;
```

Es folgt nun ein neuer, vereinfachter Interpreter, den wir für den Bildschirm-Editor brauchen sowie die Hilfsvariable:

```
NEGONE VARIABLE IFLAG
: INTERPRETER NEGONE IFLAG POKEW READLINE
  REPEAT SCAN IFLAG PEEKW * EQZ UNTIL
          NAMES PEEKW LOOKUP
          IF EXECUTE
          ELSE CONAXB EQZ IF ERROR TYPE
          PRINTS ZERO IFLAG POKEW
          THEN THEN LOOP
          IFLAG PEEKW EQZ IF CLEAR THEN ;
```

Dieser Interpreter arbeitet ähnlich wie die Funktion INTERACT, nur ohne Prompt-Zeichen und ohne Anzeige des obersten Stackwertes. Mit Hilfe der folgenden Funktion kann man von INTERACT nach INTERPRETER umsteigen und umgekehrt:

```
: ENTRY GETWORD NAMES PEEKW LOOKUP
  IF EXEC POKEW
  ELSE ERROR TYPE PRINTS THEN ;
```

Der Aufruf erfolgt mit "ENTRY INTERPRETER". Für den Editor benötigen wir noch eine Variable, Konstanten und eine Hilfsfunktion:

```
ONE VARIABLE RFLAG
CADR BLINK CONSTANT BLADR
```

```

: READSCR RFLAG PEEKW DUP 48 EQ
      IF POP 13 ONE RFLAG POKEW
      ELSE INC RFLAG POKEW
          CURSADR PEEKW PEEKB THEN ;

```

CADR READSCR CONSTANT RSCR

Nun kann endlich der EDITOR compiliert werden:

```

: EDITOR REPEAT CIN DUP 16 EQ UNTIL
      COUT LOOP
      POP RSCR INVAR POKEW
      INTERPRETER
      BLADR INVAR POKEW
      ONE ONE SCREEN ;

```

In den EDITOR-Modus kann man ebenfalls mit ENTRY einsteigen. Man kann nun beliebig auf den Bildschirm schreiben und dabei auch die Cursor-Befehle des Betriebssystems (NAS-SYS!) benutzen. Dabei sollte stets die erste Position einer Zeile freibleiben, ebenso die letzte (obwohl man dies vermeiden kann, wenn man in der Definition von READSCR den Wert 48 durch 49 ersetzt). Mit "CTRL P" kann der INTERPRETER aufgerufen werden. Der Cursor muß dabei immer am Anfang der Zeile stehen, die ausgeführt bzw. compiliert werden soll. Nach Ausführung des Befehls kehrt das Programm wieder in die EDITOR-Schleife zurück, wobei der Cursor an den Anfang der ersten Zeile gesetzt wird. Es kann zwar der ganze Bildschirm benutzt werden, doch man sollte die erste Zeile als Kommandozeile freihalten (besonders, wenn man Befehle mehrfach ausführen will). Wenn man die 15. Zeile benutzt, wird der Bildschirminhalt nach oben geschoben. Wenn das nicht gewollt wird, muß man auch diese Zeile freihalten. Nach einiger Übung kann man mit dem EDITOR sehr schnell arbeiten und wird dann kaum noch den Interpretermodus benutzen.

Die folgenden Ein- und Ausgaberroutinen für Cassettenrecorder und Drucker sind auf den EDITOR-Modus abgestimmt. Zunächst brauchen wir einige Hilfsroutinen, die teilweise in Maschinencode geschrieben sind und auf das Betriebssystem zurückgreifen:

MCODE OUTC

```
XXXX 21 7A 07 22 73 0C.
```

MCODE OUTD

```
XXXX 21 7E 07 22 73 0C 21 80
```

```
XXXX 0C 22 78 0C.
```

(Bei C80H liegt bei mir das Treiberprogramm für den Hofer-Drucker)

MCODE OUTN

```
XXXX 21 7F 07 22 73 0C 21 2F
```

```
XXXX 00 22 78 0C.
```

```
: DELAY 1 FOR LOOP ;
```

```
: MOTOR 95 NAS-SYS 5000 DELAY ;
```

```
CADR NORMIN CONSTANT NIN
```

Die Verzögerungsroutine DELAY besteht aus einer einfachen Schleife und erwartet den Endwert der Schleife auf dem Stack. Es sind Verzögerungen bis zu einer halben Minute möglich. MOTOR schaltet den Recorder ein oder aus. Für die Ausgabefunktionen brauchen wir die Routine OUTLINES:

```

: OUTLINES SWAP LINE ONE FOR
      REPEAT CIN DUP COUT
          13 EQ UNTIL
      LOOP LOOP ;

```

Diese Funktion wird folgendermaßen aufgerufen: (Startzeile) (Zeilenzahl) OUTLINES. Nun kann der Bildschirminhalt (im EDITOR-Modus!) auf einfache Weise ausgegeben werden:

```

: WRITEC MOTOR OUTC 193 COUT BACKSPACE
      OUTLINES 197 COUT BACKSPACE
      OUTN 5000 DELAY MOTOR ;
: PRINT OUTD OUTLINES OUTN ;

```

WRITEC schreibt als Startzeichen den Wert des Zeichens "A" + 128, d.h. mit gesetztem Bit 7 auf's Band und als Schlußzeichen den Wert von "E" + 128. Ich hatte es erst ohne Startzeichen versucht und dabei häufig Lesefehler erhalten. Nun müssen die auf Cassette geschriebenen Programme (oder auch Text, wie dieser Artikel) noch eingelesen werden können mit der folgenden Funktion:

```

: READC NIN INVAR POKEW MOTOR
      REPEAT CIN DUP 193 EQ UNTIL POP LOOP
      REPEAT COUT CIN DUP 197 EQ UNTIL LOOP
      COUT CIN COUT MOTOR ;

```

Was kann man damit anfangen?

In den bisherigen Beiträgen wurde vor allem der innere Aufbau dieser FORTH-Version behandelt. Wir wollen sie nun einmal von der Anwendersseite her betrachten. Wenn man das Programm gestartet hat, wartet es auf Eingaben. Dies können Zahlen (in unserem Fall ausschließlich 16-Bit-Integer/Dezimalzahlen) oder Worte sein. Als "Wort" gilt jede Folge von Zeichen, die von Leerzeichen eingeschlossen ist. Die Zahlen werden in ihr binäres Äquivalent umgewandelt und auf den Stack gegeben. Worte werden im Dictionary gesucht. Jedes Wort im Dictionary bezeichnet eine Funktion. (Auch Konstante und Variable gelten als Funktion; sie geben den Wert der Konstanten bzw. die Adresse des

Variablenwertes auf den Stack.) Im Interpretiermodus wird diese Funktion nun sofort ausgeführt. Wenn die Funktion "Werte" benötigt, nimmt sie sie vom Stack. Diese Werte müssen also vorher in der richtigen Reihenfolge eingegeben worden sein oder von einer anderen Funktion auf dem Stack hinterlassen worden sein (UPN). Wenn die jeweilige Funktion einen Wert als Ergebnis hat, gibt sie ihn ebenfalls auf den Stack. In der Regel - es gibt aber Ausnahmen - soll eine Funktion ihre Argumente "aufbrauchen". Eine Funktion kann auch ein Wort als Argument fordern (in der Regel ebenfalls eine Funktion), das dann aber nicht vorgestellt wird, sondern auf den Funktionsnamen folgt.

Das Besondere an FORTH ist nun, daß diese "Programmiersprache" durch sich selbst erweiterbar ist. Mit Hilfe der "definierenden Funktionen" (in unserem Fall ":", MCODE, MESSAGE, VARIABLE, CONSTANT) werden neue Funktionen erzeugt, die ebenfalls einen Namen erhalten, der in das Dictionary eingetragen wird. Es gibt, genau genommen, gar keine Unterscheidung zwischen "Programmen" und "Funktionen", also eigentlich gar keine "Programme" im üblichen Sinne, sondern nur ein Anwachsen der Funktionen des Interpreters und ihrer Namen im Dictionary. Ist FORTH denn nun überhaupt eine "Programmiersprache"? Ich glaube, das hängt davon ab, wie man das System anwendet. Geht man immer von einem bestimmten Standard-Wortschatz aus, wenn man den Computer eine bestimmte Aufgabe lösen lassen will, dann kann man sicher von einer "Programmiersprache" im üblichen Sinne sprechen. Man kann den Interpreter aber auch als eine Art komfortables Betriebssystem auffassen oder als eine Bibliothek von Unterprogrammen. Dann wird das System bei jedem Anwender sehr schnell eine individuelle Gestalt annehmen, die den jeweiligen Bedürfnissen angepaßt ist. Die im folgenden beschriebene Funktion stellt eine solche individuelle Erweiterung dar, die nun wirklich nichts mehr mit dem Standard-FORTH zu tun hat:

```

: TESTKL DUP 96 > IF
    125 < IF ZERO ELSE NEGONE THEN
    ELSE POP NEGONE THEN ;
ZERO VARIABLE LASTC
: MISCHTEXT REPEAT CIN DUP TESTKL UNTIL
COUT LOOP DUP 32 EQ IF LASTC PEEKW 32 EQ IF

```

```

COUT MISCHTEXT THEN THEN DUP LASTC POKEW ;
; , 23 COUT READLINE ;
CADR MISCHTEXT CADR READLINE 16 + POKEW
TESTKL ist eine Hilfsfunktion, die prüft,
ob es sich bei dem obersten Stackwert um
das Zahlenäquivalent eines Kleinbuchstabens
handelt, und entsprechend 0 oder -1 (FFFFH)
auf den Stack gibt. Mit Hilfe dieser Funk-
tion wird nun MISCHTEXT definiert. Dies ist
eine Eingaberoutine, die Kleinbuchstaben
und überflüssige Leerzeichen ignoriert. Sie
wird in die Funktion READLINE an Stelle der
normalen Eingaberoutine CIN hineinkopiert.
Beide Interpreter und der Compiler ignorie-
ren nun Kleinschrift, so daß man damit be-
liebig Kommentare einfügen kann, ja man
kann sogar in gut verständlichen "ganzen
Sätzen" Programme schreiben. Anders als bei
Kommentaren in Basic wird dadurch weder der
Speicherbedarf vergrößert noch die Ausfüh-
rung eines Programms verlangsamt. Reizvoll
wäre es z.B. auch, die englischen Namen im
Dictionary durch deutsche zu ersetzen (dann
wird das Wörterverzeichnis allerdings er-
heblich länger).

```

Die Funktion "," koppelt zwei Befehlszeilen oder zwei Funktionsdefinitionen aneinander. Man kann damit z.B. eine ganze Bildschirmseite mit Befehlen vollschreiben, wobei man jede Zeile mit einem Komma abschließt, und dann die ganze Seite auf einmal abarbeiten lassen (EDITOR-Modus!). Ebenso kann man mehrere Definitionen miteinander verbinden, wenn man hinter das abschließende Semicolon einer Definition ein Komma setzt. All diese Funktionen sind für den EDITOR-Modus, weniger für den INTERACT-Modus geeignet, da sich dann beim Löschen mit "Backspace" in einigen Fällen Fehler ergeben können.

(Oben kommt wie an einigen anderen Stellen die Funktion "-" vor; gemeint ist immer "-=", ich habe das bloß bei mir noch nicht korrigiert!)

Zum Abschluß möchte ich noch die angekündigte LIFE-Variante bringen. Das Besondere an dieser Version ist, daß als "Lebensraum" annähernd eine Kugeloberfläche simuliert wird. Zellgebilde, die z.B. über den oberen Bildschirmrand hinauswandern, tauchen am unteren Bildschirmrand wieder auf; ebenso gilt dies für den linken und rechten Bildschirmrand. Nur an den vier Eckpunkten ergeben sich Unregelmäßigkeiten. Das Programm läuft im Editormodus und benutzt die Zeilen

2 bis 15 als Spielfeld. Zeile 1 dient als Kommandozeile. Zunächst wird der Bildschirm mit der gewünschten Zellkonfiguration beschrieben (Zelle = 0). Dann kehrt man in Zeile 1 zurück und startet das Programm mit "(Anzahl) GENERATIONS". Das Programm läuft sehr langsam. Das könnte man ändern, wenn man die innere Schleife in Maschinencode programmieren würde. Es wird kein Zwischenspeicher benötigt, doch wer Bit 7 für die Grafik verwendet, der könnte einige Ueberraschungen erleben - vielleicht sieht das sogar ganz lustig aus!

Die Regeln des "Spiels" hat Peter Bentz im NASCOM-Journal 1/80 beschrieben. Wenn diese Version hier zu langsam ist, dem sei die dort abgedruckte Maschinencodeversion empfohlen.

```

: DODUP OVER OVER ;
: MAT+ ROT + ROT ROT + SWAP ;
47 CONSTANT XLEN 15 CONSTANT YLEN
: CLOSED DUP EQZ IF POP XLEN THEN
    DUP XLEN ^ IF POP ONE THEN
    SWAP DUP ONE EQ IF POP YLEN THEN
    DUP YLEN ^ IF POP 2 THEN
    SWAP ;
ZERO VARIABLE COUNT
: GET CURSADR PEEKW PEEKB ;
: PUT CURSADR PEEKW POKEB ;
L I F E   Bildschirmversion   Seite 1
: MARKL COUNT PEEKW GET 79 EQ
    IF DUP 2 ^
        IF 4 ^
            IF 128 ELSE ZERO THEN
            ELSE POP ZERO THEN
        ELSE 3 EQ
            IF 128 ELSE ZERO THEN THEN
            GET + PUT ; ZERO VARIABLE VPOP ;
: SETL ZERO VPOP POKEW YLEN 2 FOR XLEN ONE FOR
J I SCREEN GET 128 AND EQZ IF 32 PUT ELSE
79 PUT ONE VPOP MEM+ THEN LOOP LOOP ;
L I F E   Bildschirmversion   Seite 2
ZERO VARIABLE VGEN
MESSAGE MGEN
GENERATION: "
MESSAGE MPOP
POPULATION: "
L I F E   Bildschirmversion   Seite 3
: TESTL ZERO COUNT POKEW
    ONE NEGONE FOR ONE NEGONE FOR
    DODUP J I MAT+ CLOSED SCREEN
    GET 127 AND 79 EQ
    IF ONE COUNT MEM+ THEN LOOP LOOP ;
: GENERATE YLEN 2 FOR XLEN ONE FOR

```

```

J I TESTL SCREEN MARKL LOOP LOOP ;
: GENERATIONS ONE FOR GENERATE SETL
    ONE VGEN MEM+ 1 2 SCREEN MGEN TYPE
    VGEN PEEKW = 1 25 SCREEN MPOP TYPE
    VPOP PEEKW = LOOP ;
L I F E   Bildschirmversion   Seite 4

```

## SEITE(N) für EINSTEIGER

### 1. Teil von Günter Böhm

Immer häufiger wird der Wunsch an uns herangetragen, uns doch nicht in fernen Sphären der Programmierkunst zu verlieren, sondern auch an die Einsteiger zu denken, die sich zunächst mit den Grundbegriffen der Maschinensprache vertraut machen wollen. Die Fortgeschrittenen dürfen hier einmal ruhig weiterblättern. (Oder hätte nicht vielleicht jemand Lust, auch einen Teil dieser Serie zu schreiben? Wir wären für jeden Beitrag dankbar).

Dies soll keine Konkurrenz zu den vielen Z80 Büchern, die bereits auf dem Markt sind, sein. Hier soll in einer Serie von einfachen Programmierbeispielen der Befehlssatz des Z80 aufgezeigt und praktisch erprobt werden. Was systemeigene Adressen angeht, beziehen wir uns selbstverständlich immer auf NASCOM 1 und 2.

Zunächst zum Aufbau der Serie. Der Übersichtlichkeit wegen sollen die einzelnen Kapitel nach den Befehlsblöcken im Z80 Handbuch gegliedert werden. (Das erste Kapitel würde also die 8Bit Ladebefehle betreffen). Als Grundlage muß aber das Rechnen im Binär- und Hexadezimalsystem vorausgesetzt werden. Der Vollständigkeit halber also zunächst einmal ein kurzer Abriß dieser Rechensysteme.

Bekanntlich rechnen wir im Dezimalsystem (Zehnersystem), d.h. beim Zählen wechseln wir nach neun Zählschritten in die nächste Stelle. 08 bedeutet also 8 Einerschritte; 10 bedeutet: es ist durch neun Zählschritte die erste Stelle bereits voll besetzt worden, wir mußten also auf die nächste ausweichen und sie mit 1 belegen. Der Zählvorgang könnte auf der ersten Stelle nun fortgesetzt werden, bis wir wiederum die zweite belegen müssen.

Ein Rechner kann aber keine 9 Zustände erfassen (oder 10, wenn man das Nichtvorhandensein einer Zählereinheit =0 mitzählt), sondern nur zwei. Das entspricht den beiden Zuständen (im Z80) von ca +5V für 1 und ca 0V für 0. Wir müssen also beim Zählen schon nach der Ziffer "1" auf die nächste Stelle wechseln. Dieses System heißt Binär,-Dual- oder Zweiersystem.

Binär Dezimal

1 = 1  
 10 = 2  
 11 = 3  
 100 = 4 etc.

Genauer betrachtet bilden die Stellen des Binärsystems 2er Potenzen (wie das Dezimalsystem auf Zehnerpotenzen aufgebaut ist).

Dezimal 1084 = 4 mal 10 hoch 0 = 04  
 +8 mal 10 hoch 1 = 80  
 +0 mal 10 hoch 2 = 00  
 +1 mal 10 hoch 3 = 1000  


---

 1084

Binär 1101 = 1 mal 2 hoch 0 = 1  
 +0 mal 2 hoch 1 = 0  
 +1 mal 2 hoch 2 = 4  
 +1 mal 2 hoch 3 = 8  


---

 13 dez.

Diese Art der Rechnerel mag zwar einem Computer gefallen, für uns ist sie zu unübersichtlich. Deshalb hat man je vier Stellen zusammengefaßt, die man dem Rechner als sogenanntes Byte (Datenwort aus 2 mal 4 Bits=Binärstellen) eingibt. Der wandelt sie wieder intern in sein genehmes Binärsystem um.

Zählt man, welche Werte ein sogenanntes Halbbyte nacheinander einnehmen kann, so kommt man auf 16, (0 ~ 15) Und schon sind wir ins 16er-System gerutscht (Hexadezimalsystem). Da wir in unserem Sprachschatz aber nur 10 Ziffern kennen, müssen wir uns für dieses System etwas einfallen lassen: man nimmt einfach die Buchstaben A bis F dazu. Das Zählen im Hex-System sieht also so aus:  
 Hex 0 1 2 3 4 5 6 7 8 9 A B C D E F 10 11 12  
 13 14 15 16 17 18 19 1A 1B 1C 1D etc.

In diesem System arbeiten wir also mit 16er Potenzen.

A325 hex bedeutet also:

A (=10 dez.) mal 16 hoch 3 = 40960  
 3 mal 16 hoch 2 = 768  
 2 mal 16 hoch 1 = 32  
 5 mal 16 hoch 0 = 5 dez.

Das Ergebnis ist 41765 dezimal.

Glauben Sie nur nicht, man könnte auch nach längerer Übung solche umständliche Rechnerel aus dem Ärmel schütteln. Dafür benutzt man am besten ein Programm zur Umwandlung von Dezimal- in Hexzahlen oder umgekehrt. Aber hie und da muß man es eben doch im Kopf rechnen, und es gehört eben einfach dazu, um die Zusammenhänge zu verstehen.

Bevor wir ins Programmieren einsteigen, sollten Sie den Umgang mit den Hexzahlen einmal üben. Mehr als 4 Stellen benötigen wir nicht, denn unser NASCOM kann auch nicht weiter als FFFF. Beim Rechnen ist hauptsächlich zu beachten, wann der Übergang zur nächsten Stelle erfolgt.

Ab 4096-65535 ab 256-4095 ab 16-255 0-15

Hex1000-FFFF 100-FFF 10-FF 0-0F  
 Ihr Betriebssystem T4 oder NASSYS hält eine Routine bereit, die leicht mit Hexzahlen rechnen läßt. Die Eingabe AYYY XXXX gibt auf dem Bildschirm die Summe von XXXX und YYYY und deren Differenz aus. Beide Eingaben können beliebige Hexzahlen sein. So können Sie das richtige Rechnen im Hex-System üben. (Am Anfang ist es sicher sehr frustrierend.)

Rechenbeispiel: C0 + 2A

Zunächst addiere ich die ersten Stellen 0+A= A, es bleibt kein Übertrag. Dann folgen die zweiten Stellen: C (=12dez) + 2= E (14dez). Das Ergebnis ist EA.

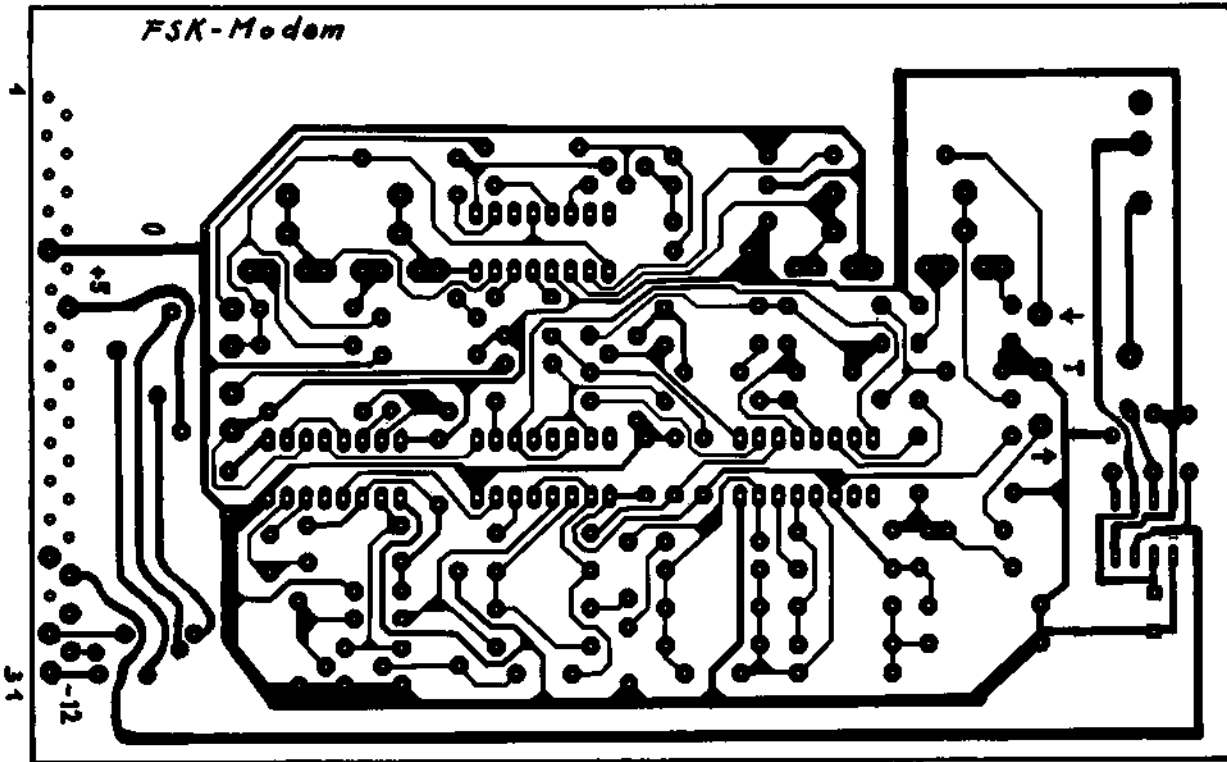
C0 - 2A

Die erste Stelle ist 0. Davon kann ich nichts abziehen, also "leihe" ich mir eine von der nächsten Stelle. Die ist im 16er-System logischerweise 16. Also 16 minus A (10dez) ergibt 6 als erste Stelle. Da ich mir eine "geliehen" habe, muß ich auch eine "behalten": 2+ die behaltene ist 3. 3 und wieviel ist C (12dez)? Ergebnis =9. Die Differenz ist also 96 (müßte auf dem Bildschirm nach Eingabe von A2A C0 NL als zweite Zahl ausgegeben werden.)

Nun aber genug mit der Grundschule für Computerfans. Üben Sie ein bißchen mit dem A-Befehl zur Kontrolle.

Zur Überprüfung Ihrer Umwandlungskünste von Dez. und Hexzahlen können Sie das Rechner-

# FSK-Modem



programm im Journal 4,5-80 benutzen. Die Anwendung ist dort genauer erklärt. Mit T4 läuft das Programm, wenn man in Adresse 0DBA anstatt CD 86 02 - CD DB 01 einträgt.

Folgende Version läuft mit NASSYS3. Gestartet wird mit E D54.

In der nächsten Folge soll es dann mit dem Programmieren losgehen. Bis dahin sind Hexzahlen für Sie sicher schon ein alter Hut!

TC80 E3F

```

OC80 F5 D5 C5 3E 00 B8 EB 21 1D
OC88 00 00 28 05 19 38 32 10 54
OC90 FB C1 D1 F1 C9 F5 D5 11 BE
OC98 00 00 ED 42 38 03 13 18 39
OCA0 F9 EB D1 F1 C9 F5 C5 D5 AA
OCA8 1E 0B E5 CB 3C CB 1D 1D CE
OCB0 28 0A E5 C1 E1 E5 CD 95 BC
OCB8 0C 09 18 EF C1 D1 C1 F1 24
OCC0 C9 EF 1E 4F 55 54 20 4F 09
OCC8 46 20 52 41 4E 47 45 20 C7
OCD0 21 00 C3 59 03 C9 CB 3C EC
OCD8 CB 3C CB 3C CB 3C 44 21 5E
OCE0 E8 03 CD 80 0C EB E1 E5 E1
OCE8 7C E6 0F 47 21 64 00 CD FE
OCF0 80 0C 19 EB E1 E5 CB 3D 5A
OCF8 CB 3D CB 3D CB 3D 45 21 82
    
```

```

0D00 0A 00 CD 80 0C 19 EB E1 55
0D08 7D E6 0F 6F 26 00 19 C9 FE
0D10 E5 CD 95 0C 7D C5 E1 47 DA
0D18 C6 30 F7 00 00 CD 80 0C 6B
0D20 EB E1 ED 52 C9 01 10 27 39
0D28 CD 10 0D 01 E8 03 CD 10 E8
0D30 0D 01 64 00 CD 10 0D 01 9A
0D38 0A 00 CD 10 0D 7D C6 30 AC
0D40 F7 00 00 C9 2A 0C 0C CD 1C
0D48 D5 0C E5 2A 0E 0C CD D5 01
0D50 0C 45 E1 C9 EF 0C 45 49 E1
0D58 4E 47 41 42 45 20 48 65 8F
0D60 78 20 44 65 7A 20 3F 00 87
0D68 DF 7B 00 00 0A FE 48 20 35
0D70 0A 32 CA 0B 3E C9 32 D5 9C
0D78 0C 18 07 FE 44 20 E9 CD C8
0D80 26 0E 3E 09 32 CB 0B EF FF
0D88 20 41 55 53 47 41 42 45 AD
0D90 20 3F 00 DF 7B 00 00 00 56
0D98 FE 48 20 11 32 CC 0B 21 46
0DA0 DF 66 22 25 0D 21 00 C9 30
0DA8 22 27 0D 18 07 FE 44 20 8C
0DB0 E2 CD 2F 0E DF 6A DF 63 34
0DB8 1A F5 13 DF 79 F1 00 FE 2E
0DC0 4D 28 13 FE 44 28 1B FE D8
0DC8 51 28 30 FE 57 28 3C FE 35
0DD0 4E CA 1B 04 18 44 CD 44 81
0DD8 0D CD 80 0C CD 25 0D C3 0D
0DE0 B4 0D 2A 0C 0C CD D5 0C 9E
0DE8 E5 2A 0E 0C CD D5 0C E5 B1
0DF0 C1 E1 CD 95 0C CD 25 0D 0C
0DF8 C3 B4 0D 2A 0C 0C CD D5 6D
0E00 0C 45 CD 80 0C CD 25 0D B7
0E08 C3 B4 0D 2A 0C 0C CD D5 7E
0E10 0C CD A5 0C CD 25 0D C3 6A
0E18 B4 0D 2A 0C 0C CD D5 0C D7
0E20 CD 25 0D C3 B4 0D 32 CA AD
0E28 0B 3E E5 32 D5 0C C9 32 72
0E30 CC 0B 21 01 10 22 25 0D 9B
0E38 21 27 CD 22 27 0D C9 2A A4
    
```



# FSK-Modem von Günter Böhm

Folgende Schaltung besteht aus einem FSK-Modem und einem Taktgenerator (4800 Hz), die als Cassetteninterface für Kansas-City-Standard dienen sollen. Der Unterschied zum Vorschlag von D.Maisl: Die Platine kann problemlos am PIO-Bus angeschlossen werden, nur ein Umschalter am ext.Clock ist erforderlich. Der Nachteil: die Baudrate ist nur änderbar, wenn man Kondensatorwerte am Taktgenerator verändert. (Ein Mangel an Komfort).

Die Veröffentlichung der Modem-Schaltung wurde uns freundlicherweise genehmigt von der Zeitschrift

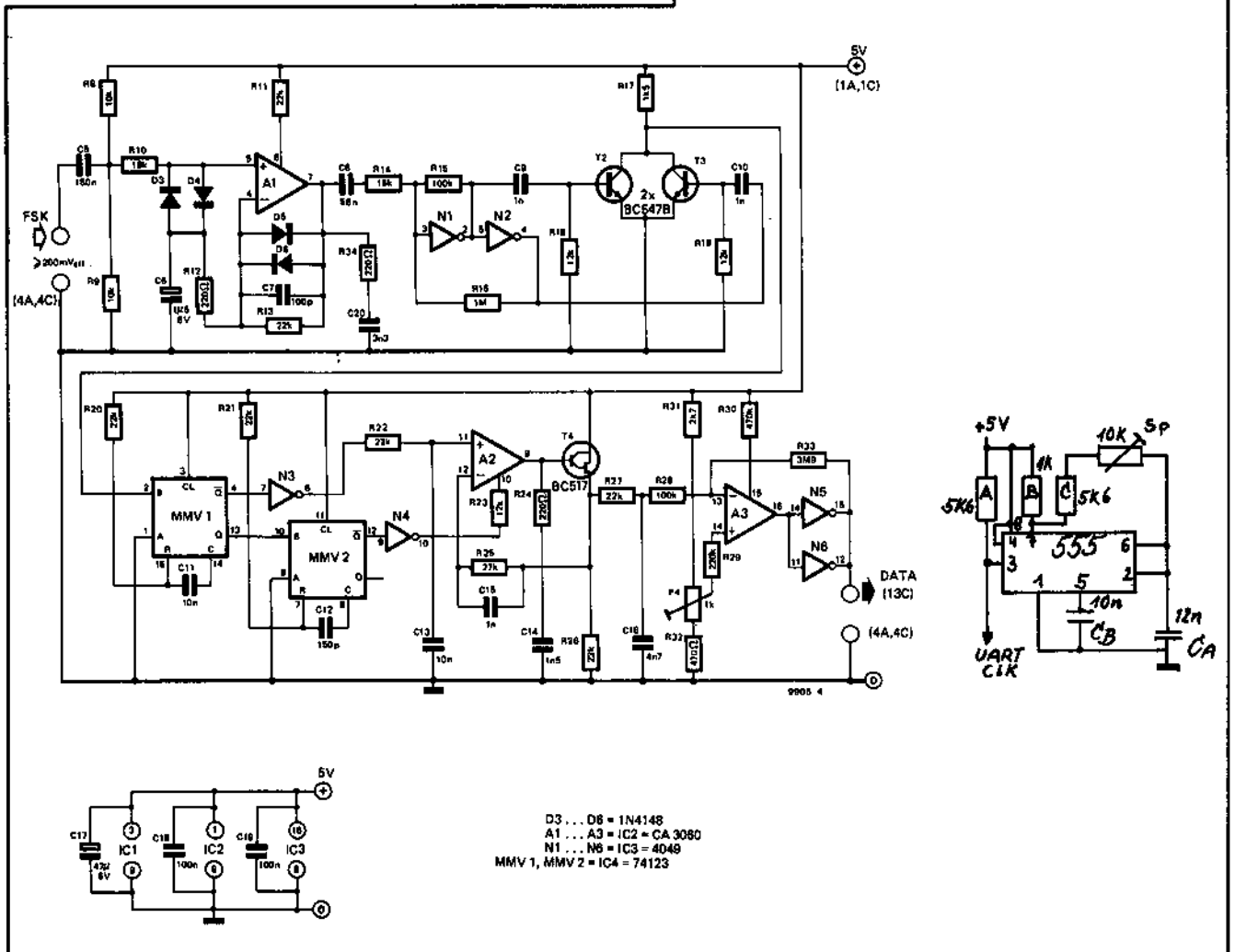
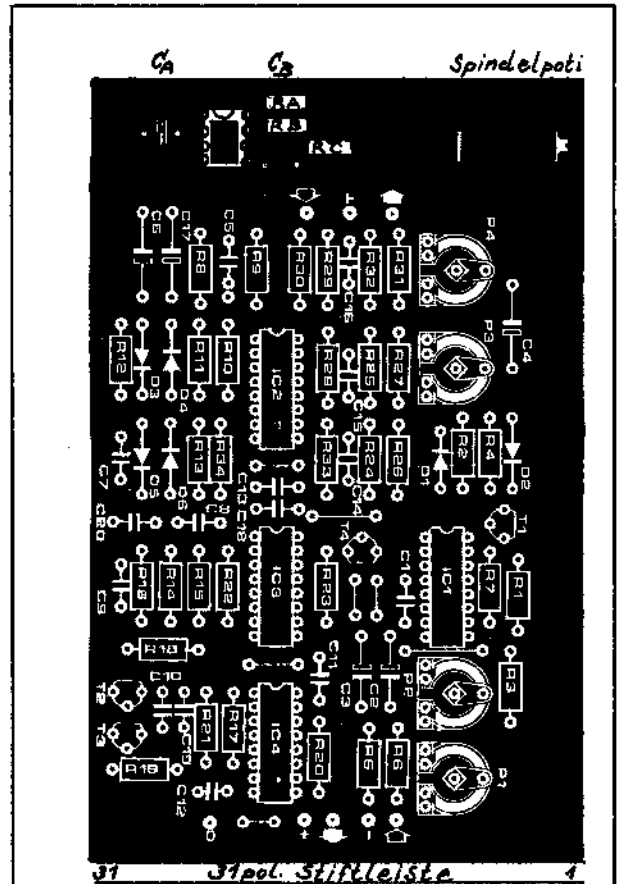
Elektor

5133 Gangelt 1

Postfach 1150

Der Taktgenerator entspricht im Aufbau dem auf der Grundplatte des NASCOM 1.

Näheres über Funktion und Abgleich des Modems erfahren Sie in Elektor, Feb.1978.



# NASSYS 3

Diese Spalte, die von nun an regelmäßig erscheint, soll Informationen über die Möglichkeiten, die Vor- und Nachteile von NASSYS3 liefern.

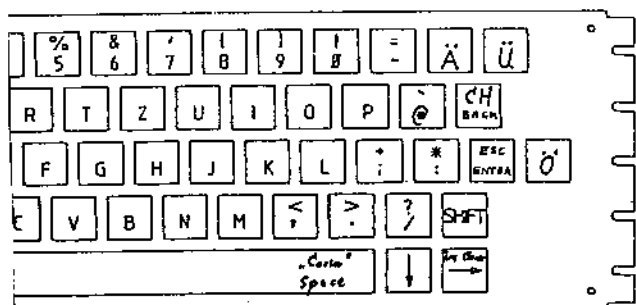
Die Leser sind natürlich wieder aufgerufen, sich an der Sammlung von Material zu beteiligen und über Erfahrungen mit NASSYS3 oder Anwendung von Unterprogrammen etc. zu berichten.

Eine Bitte: wenn Sie Modifikationen vornehmen, achten Sie darauf, daß sie für andere Anwender kompatibel bleiben. Ein Programm sollte mit jeder NASSYS3 Version laufen können. Inzwischen sind schon NASSYS5 und MJK-SYS aufgetaucht; mir graut vor der Zeit, wenn wir für jeden Leser eine Anpassung eines Programmes an sein individuelles Betriebssystem liefern müssen.

Mit folgender Modifikation von Werner Öhring wird die Tastatur auf deutsche Norm umgestellt. Um sie voll auszunutzen, sollte man allerdings einen Character Generator mit deutschem Zeichensatz besitzen.

Deutsch	ASCII		
Ä	5B	[	CS=FF=Ctrl L
Ö	5C	\	LF=Ctrl J
Ü	5D	]	B=Ctrl >
↑	5E		
Cursor	5F	—	L=Execute XXXX
ä	7B	{	
ö	7C		
ü	7D	}	
B	7E	—	
DEL	7F	⌫	
S	40	@	

Endlich kann man nun die Umlaute "shiften", eine Möglichkeit, an die die angelsächsischen "Erfinder" von NASSYS 3 natürlich nicht gedacht haben.



	Engl.	Dtsch
0130	5B	5E : a wirkt nicht mehr als
0148	06	0C : Control-Taste
014C	A0	08 @ → @
014D	18	06 shift+@ →
014E	06	20 unabhängig von KO/K1
014F	CB	18
0150	6E	04
0151	28	FF frei
0152	02	FF "
0153	EE	FF "
0154	40	FF "
018A	80	E0 schnellere Blinkfrequenz
018B	02	00
018C	50	38 kürzere Wartezeit
018E	00	90 für Repeat-Automatik
018F	01	00
05C4	8E	FF LF=Ctrl+J
05C6	88	FF FF=Ctrl+L
05D1	0E	88 CH=shift+Backsp.
0613	3D	3C =Y QWERTZU Tastatur
0614	3C	3D =Z
0616	9E	0E =ö
0619	96	94 Cursor=shift+space
0798	66	XX Sprung zu User-Progr.
0799	03	XX (z.B. Printer On etc.) mit Taste "L"

## NASSYS 3 - KOMPATIBEL ??? von Michael Bach

Als ich kürzlich meinen Rechner auf den NASSYS 3 Monitor umrüstete, mußte ich feststellen, daß er leider in einigen Punkten mit NAS-SYS 1 nicht kompatibel ist. Zur Beruhigung: Alle diese Probleme treten nur auf, wenn man in den normalen Monitorablauf eingegriffen hat. Mir ist bis jetzt folgendes aufgefallen:

1. (Noch harmlos): Wenn in der Titelzeile ein CR gedruckt wird, springt der Cursor nach Zeile 1. Dies ist zwar sinnvoll, aber halt anders als bei NASSYS 1.
  2. Wenn man das "R"-Kommando von Maschinenprogrammen aus aufruft, muß man die Variable "ARGN" auf 0 setzen.
  3. (Und dies ist ärgerlich): Wenn man die Tastatur-Tabelle verändert, ist die Variable "\$KTAB" im Gegensatz zu früher auf das Ende der neuen Tabelle zu setzen!
- Weiterhin sei noch darauf hingewiesen, daß beim Aufruf von Monitorbefehlen vom Programm aus die Parameter mal in "ARG1" usw. und mal

in den Registern übergeben werden müssen; Beispiel: für "W" in ARG1, ARG2; für "I" in HL, DE & BC. Außerdem verändern die meisten Monitor-Routinen (mit Ausnahme von "ROUT", "RIN" und so) die Register; z.B. "RKBD" ändert außer AF auch B, DE & HL.

Na ja, ansonsten ist NAS-SYS 3 ja eine echte Verbesserung (vor allem kann man sich jetzt die Repeat-Routinen sparen) und das fehlende "L"-Kommando zum Eintippen von Hex-Listings mit Prüfsumme kann man ja leicht (und verbessert) selber schreiben.

(Siehe CONMOD in diesem Heft Red.)

## NASSYS3 MODIFIKATION

von Rüdiger Maurer

Ich möchte hier eine NASSYS3-Modifikation vorstellen, die es ermöglicht, bei der Benutzung der T-Funktion die Checksumme auf dem Bildschirm mit aufzulisten, vorteilhaft um ein Zeichen nach rechts versetzt (siehe auch Ausdruck). Ein weiterer Vorteil dieser Modifikation ist der, daß bei der Verwendung des Druckers Seiko GP80 das T-Kommando ebenfalls benutzt werden kann, da das Hex-Zeichen 08 nicht ausgesendet wird, was den Drucker nämlich in den Grafik-Modus schicken würde.

Folgende Bytes sind im Monitor zu ändern:

```
T02F5
 02F5 DF 69 00 79 DF 60 DF 6A 40
```

## SOFTWARE REPEAT von Christoph Rau

Mit dieser kleinen Routine simuliere ich die Repeat-Funktion meiner Tastatur. NAS-SYS pollt die Tastatur mit der Routine Nr.61H (KBD). Für jede Zeile der Tastatur-Matrix merkt das System sich die letzte Eingabe in der Workspace. Nur wenn die Eingabe von der Tastatur sich von dem Eintrag in der Workspace unterscheidet, geht das Programm davon aus, daß ein Zeichen eingegeben werden soll. Deshalb muß für jedes Zeichen die Taste einmal gedrückt und wieder losgelassen werden. (Das hat mit der Entprellung durch Software nichts zu tun, die muß außerdem noch gemacht werden.)

Das habe ich nun so geändert, daß bei jedem neuen Niederdrücken einer Taste zusätzlich zu der Ausgabe des Zeichens ein Repeat-Zähler mit 0300H geladen wird. Solange die Taste gedrückt bleibt, wird bei jedem Durchlauf der Routine dieser Zähler dekrementiert. Ist er auf 0 runtergezählt,

wird er mit 0080H geladen und das Zeichen wird wieder ausgegeben. Durch die verschiedenen Zählerwerte wird am Anfang länger gewartet, der Repeat kann dann aber ziemlich schnell laufen. Durch Verändern von INIT und INTER (in OCBFH und in OCD6H) kann dieses Tempo variiert werden. Das ganze ist natürlich keine vollständige Keyboard-Routine, sondern der geänderte Anfang der Monitor-Routine. Es wird dann zum Bestimmen des Zeichens etc. absolut in die Routine KBD gesprungen.

Um das Programm nun in NAS-SYS einzubinden, muß eine neue Tabelle der Input Routinen geschrieben werden, in der nur die Nr.76H steht, gefolgt von 00H. Wenn nun der Zeiger \$IN (OC75H in der Workspace) auf diese Tabelle gesetzt wird, dann wird nur die vom Benutzer definierte Eingabe Routine abgearbeitet. Dazu muß noch die Adresse unserer Routine bei \$UIN (OC7BH) eingetragen werden. Das Umsetzen dieser Tabellen und Zeiger besorgt die Initialisierung, die mit "E OCFB" aufgerufen wird.

Weil mit \$UIN getrickst wird, führt der "U" Befehl des Monitors dazu, daß immer 2 Zeichen auf einmal auf dem Bildschirm erscheinen, weil dann nämlich sowohl die Standard- als auch die neue Routine ausgeführt werden. Das Programm beginnt bei mir bei OCA5H, weil darunter noch die Workspace meines Assemblers liegt.

OCA5	0010	ORG	#CA5
0002	0020	COUNT	DEFS 2
OCA7	3E02	0030	POLL LD A,2
OCA9	CD4500	0040	CALL #45
OCAC	21010C	0050	LD HL,#C01
OCAF	DB00	0060	IN A,(0)
OCB1	2F	0070	CPL
OCB2	77	0080	LD (HL),A
OCB3	E5	0090	PUSH HL
OCB4	2AA50C	0100	LD HL,(COUNT)
OCB7	010100	0110	LD BC,1
OCBA	ED42	0120	SBC HL,BC
OCBC	3003	0130	JR NC,SKIP
OCBE	218000	0140	LD HL,INTER
OCC1	22A50C	0150	SKIP LD (COUNT),HL
OCC4	E1	0160	POP HL
OCC5	0608	0170	LD B,8
OCC7	3E01	0180	LOOP LD A,1
OCC9	CD4500	0190	CALL #45
OCCC	23	0200	INC HL
OCCD	DB00	0210	IN A,(0)
OCCF	2F	0220	CPL
OCDO	57	0230	LD D,A
OCD1	AE	0240	XOR (HL)
OCD2	280B	0250	JR Z,EQUAL
OCD4	E5	0260	PUSH HL
OCD5	210003	0270	LD HL,INIT
OCD8	22A50C	0280	LD (COUNT),HL
OCD8	E1	0290	POP HL
OCDC	C3A700	0300	JP #A7
OCDF	7A	0310	EQUAL LD A,D
OCE0	B7	0320	OR A
OCE1	2809	0330	JR Z,NULL
OCE3	E5	0340	PUSH HL
OCE4	2AA50C	0350	LD HL,(COUNT)
OCE7	ED6A	0360	ADC HL,HL
OCE9	E1	0370	POP HL
OCEA	2803	0380	JR Z,RUN
OCEC	10D9	0390	NULL DJNZ LOOP
OCEE	C9	0400	RET
OCEF	0EFF	0410	RUN LD C,#FF
OCF1	37	0420	SCF
OCF2	0C	0430	SHIFT INC C
OCF3	1F	0440	RRA
OCF4	30FC	0450	JR NC,SHIFT
OCF6	C3C900	0460	JP #C9
0300	0470	INIT	EGU #300

```

0080      0480 INTER EQU #80
0CF9 7600 0490 TAB DEFW #76
OCFB 21A70C 0500 SYSINI LD HL,POLL
OCFE 227B0C 0510 LD (UTAB),HL
OD01 21F90C 0520 LD HL,TAB
OD04 22750C 0530 LD (INTAB),HL
OD07 DF 0540 RST #18
OD08 5B 0550 DEFB #5B
OC7B 0560 UTAB EQU #C7B
OC75 0570 INTAB EQU #C75

```

```

CA0 D10
OCA0 80 10 EE 06 7F 00 03 3E F0
OCAB 02 CD 45 00 21 01 0C DB D1
OCB0 00 2F 77 E5 2A A5 0C 01 23
OCB8 01 00 ED 42 30 03 21 80 CB
OCC0 00 22 A5 0C E1 06 08 3E CC
OCCB 01 CD 45 00 23 DB 00 2F 14
OCD0 57 AE 28 08 E5 21 00 03 1D
OCD8 22 A5 0C E1 C3 A7 00 7A 7C
OCE0 B7 28 09 E5 2A A5 0C ED 81
OCEB 6A E1 28 03 10 D9 C9 0E 2A
OCF0 FF 37 0C 1F 30 FC C3 C9 15
OCFB 00 76 00 21 A7 0C 22 7B EB
OD00 0C 2E F9 22 75 0C DF 5B 1D
OD08 EC 21 21 0D 46 CB 48 CB 71

```

## 2716 EPROMMER

### Copyright Rüdiger Maurer

Das Programmierprogramm wird in einem EPROM geliefert und ist voll verschiebbar. Das EPROM mit dem Programmierprogramm kann man also in jeden freien EPROM-Sockel (2708) des NASCOM-Systems stecken. Gestartet wird das Programm dann auf der Anfangsadresse des Sockels.

Nach dem Start erfragt das Programm alle nötigen Eingaben vom Benutzer, so daß sich eine genaue Beschreibung erübrigt.

Die Antworten des Benutzers werden jeweils mit "enter" bzw. "new line" abgeschlossen.

Das Programm bietet folgende Möglichkeiten:

- 1) Test, ob das EPROM, das in der Fassung des Programmierers steckt, leer ist.
- 2) Laden des EPROMs in den Speicher des NASCOM, z.B. um EPROMs zu duplizieren.
- 3) Vergleich des EPROMs im Programmierer mit einer Stelle im NASCOM-Speicher. Man kann so überprüfen, ob ein gegebenes EPROM mit dem Speicherinhalt übereinstimmt.
- 4) Programmieren eines EPROMs. Der Speicherinhalt des NASCOM ab einer beliebigen Adresse wird als 2k Block in das EPROM programmiert. Anschließend erfolgt automatisch eine Überprüfung.
- 5) Direktes Programmieren einzelner Bytes. Das Programm fragt nach der Anfangsadresse, von wo aus das EPROM programmiert werden

soll. Anschließend gibt man die einzelnen Bytes ein. Jede Byteeingabe ist mit "enter" bzw. "new line" abzuschließen. Das Byte wird direkt programmiert und überprüft. Will man den Byte-Programmiermodus verlassen, dann gebe man eine nicht hexadezimale Kombination (z.B. "S" für Stop) ein und der Programmiermodus wird verlassen.

Das Programm läuft mit dem Betriebssystem NAS-SYS.

Hardware: Zum Programmiergerät gehört eine kleine Zusatzplatine (PIO BUS kompatibel) die die Bauteile zur Ansteuerung des EPROMs aufnimmt. Es werden 5 Volt Versorgungsspannung und 25 Volt Programmierspannung benötigt, die dem 2708-Programmiergerät entnommen werden können.

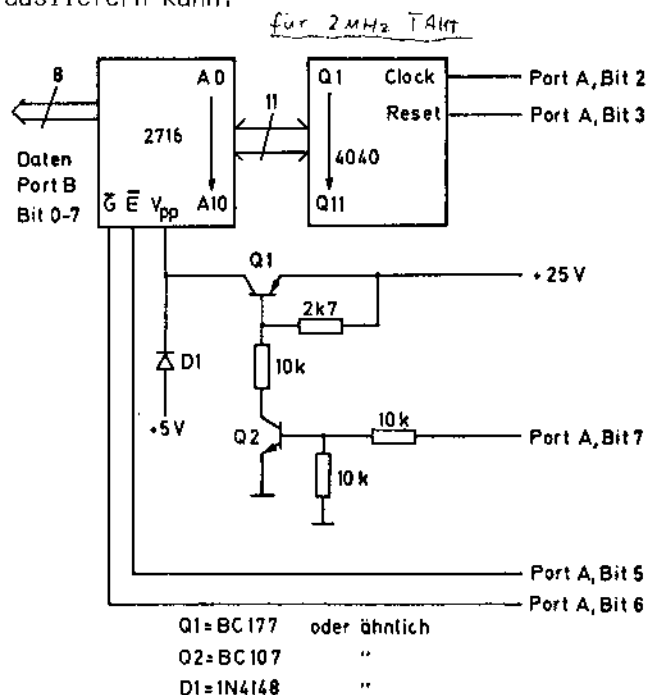
Als Service kann ich Folgendes für NASCOM Journal-Leser anbieten:

Platine: 35,- incl.

Platine + Material: 50,- incl.

Software in 2708 EPROM: 20,- incl.

Ich habe bereits einige Platinen fertiggestellt, so daß ich auch entsprechend schnell ausliefern kann.



Programmer fuer 2716  
mit 5 Volt Betriebsspannung.  
Copyright (c) R.Maurer Taunusstein

```

T1000 1000 3E 07 D3 06 D3 07 EF 0C 03
1008 00 ED 4B 0C 0C EF 0C 45 A8
1010 50 52 4F 4D 2D 50 72 6F 8C
1018 67 72 61 6D 6D 63 72 20 33
1020 66 6F 72 20 32 37 31 36 67
1028 20 45 50 52 4F 4D 27 73 75
1030 0D 20 20 62 79 20 20 52 FA
1038 75 65 64 69 67 65 72 20 4D

```

```

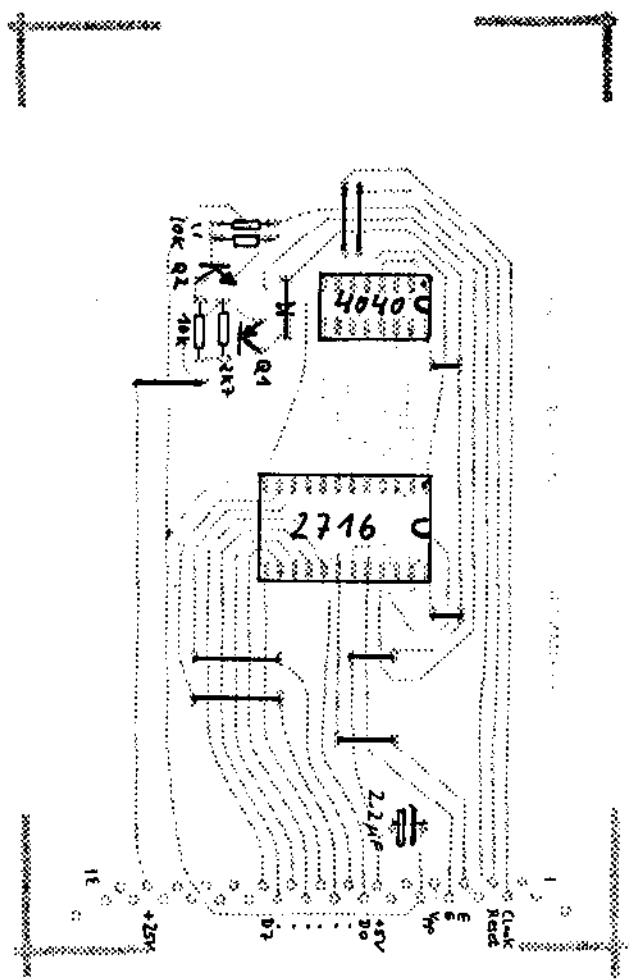
1040 4D 61 75 72 65 72 00 20 E9
1048 20 20 20 20 20 46 75 65 18
1050 6C 6C 65 6E 67 61 72 74 89
1058 65 6E 20 31 0D 20 20 20 F9
1060 20 20 20 36 32 30 34 20 8C
1068 54 61 75 6E 75 73 73 74 DF
1070 65 69 6E 20 34 0D 20 20 5D
1078 20 20 20 20 54 65 6C 2E 5B
1080 3A 20 30 36 31 32 38 20 0B
1088 2F 20 35 33 30 35 00 EF A3
1090 0D 0D 59 6F 75 20 77 61 EF
1098 6E 74 20 74 6F 0D 72 65 71
10A0 74 75 72 6E 20 74 6F 20 9C
10A8 4E 41 53 2D 53 59 53 3A 00
10B0 20 20 20 20 20 20 20 20 C0
10B8 20 20 74 79 70 65 20 4E 38
10C0 0D 74 65 73 74 2C 69 66 98
10C8 20 74 68 65 20 45 50 52 40
10D0 4F 4D 20 69 73 20 65 6D 6A
10D8 70 74 79 3A 20 74 79 70 FC
10E0 65 20 54 0D 6C 6F 61 64 76
10E8 20 74 68 65 20 45 50 52 60
10F0 4F 4D 20 69 6E 74 6F 20 96
10F8 52 41 4D 3A 20 20 20 20 A2
1100 74 79 70 65 20 4C 0D 63 AF
1108 6F 6D 70 61 72 65 20 45 02
1110 50 52 4F 4D 20 77 69 74 D3
1118 68 20 52 41 4D 3A 20 20 0B
1120 20 20 20 74 79 70 65 20 73
1128 43 0D 70 72 6F 67 72 61 14
1130 6D 6D 65 20 74 68 65 20 01
1138 45 50 52 4F 4D 3A 20 20 46
1140 20 20 20 20 20 20 74 79 FE
1148 70 65 20 50 0D 70 72 6F FC
1150 67 72 61 6D 6D 65 20 6F 69
1158 6E 6C 79 20 6F 6E 65 20 3E
1160 42 79 74 65 3A 20 20 20 9F
1168 20 74 79 70 65 20 42 0D CA
1170 00 D5 DF 63 1A D1 FE 4E CF
1178 CA 00 00 FE 54 DD 21 20 C3
1180 02 20 44 FE 4C DD 21 D5 1C
1188 01 20 1B FE 43 DD 21 ED 09
1190 02 20 13 FE 50 DD 21 75 9F
1198 02 20 08 FE 42 DD 21 41 5D
11A0 03 20 24 18 CC 00 EF 00 E0
11A8 41 74 20 77 68 69 63 68 A1
11B0 20 61 64 72 65 73 73 3F A2
11B8 20 20 0D 00 D5 DF 63 DF 0C
11C0 64 D1 38 E2 2A 21 0C ED 64
11C8 48 0C 0C DD 09 FD 21 8F CF
11D0 00 FD 09 DD E9 3E FF D3 BD
11D8 07 D3 07 D3 06 AF D3 06 2B
11E0 3E 08 D3 04 AF D3 04 06 9A
11E8 08 C5 06 00 DB 05 77 23 46
11F0 3E 04 D3 04 AF D3 04 10 80
11F8 F3 C1 10 ED EF 0D 54 68 72
1200 65 20 45 50 52 4F 4D 20 3A
1208 69 73 20 6C 6F 61 64 65 1B
1210 64 20 69 6E 74 6F 20 52 D2
1218 41 4D 20 21 0D 00 FD E9 EC
1220 3E FF D3 07 D3 07 D3 06 FC
1228 AF D3 06 3E 08 D3 04 AF 8E
1230 D3 04 1E 00 06 00 C5 06 10
1238 00 DB 05 FE FF 28 02 1E 6F
1240 01 3E 04 D3 04 AF D3 04 F2
1248 10 EF C1 10 E9 EF 0C 0D 1B
1250 54 68 65 20 45 50 52 4F D9
1258 4D 20 69 73 20 00 7B FE 4C
1260 00 20 06 EF 6E 6F 74 20 00
1268 00 EF 65 6D 70 74 79 20 88
1270 21 0D 00 FD E9 3E FF D3 A6
1278 06 AF D3 06 3E FF D3 07 2F
1280 AF D3 07 EF 0C 0D 0D 54 84
1288 68 65 20 70 72 6F 67 72 81
1290 61 6D 6D 65 72 20 69 73 80
1298 20 77 6F 72 68 69 6E 67 CB
12A0 20 21 0D 00 3E C8 D3 04 DD
12A8 3E C0 D3 04 2A 21 0C 06 EC
12B0 08 C5 06 00 7E 22 A6 08 E3
12B8 D3 05 3E C0 D3 04 3E E0 95
12C0 D3 04 F5 C5 06 0A FF 10 82

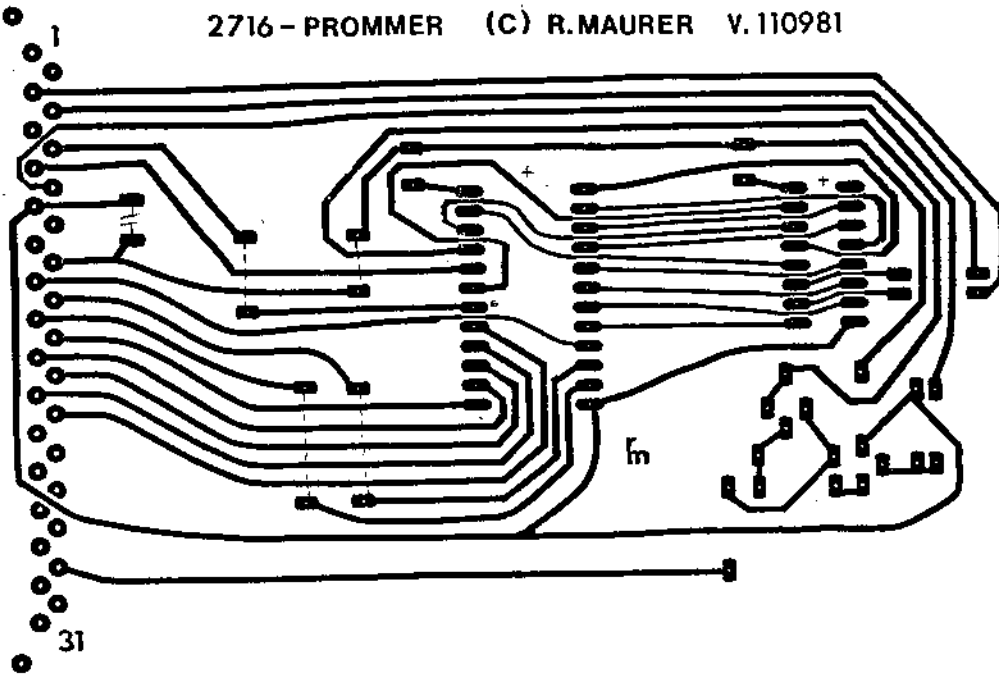
```

```

12C8 FD C1 F1 3E 00 D3 04 23 C1
12D0 3E 04 D3 04 3E 00 D3 04 10
12D8 F5 C5 06 0A FF 10 FD C1 81
12E0 F1 10 D1 C1 10 CB EF 0C 5B
12E8 0D 00 2A 21 0C 3E FF D3 6E
12F0 07 D3 07 D3 06 AF D3 06 44
12F8 3E 08 D3 04 64 D3 04 1E 80
1300 00 06 08 C5 06 00 DB 05 CC
1308 BE 28 02 1E 01 23 3E 04 87
1310 D3 04 AF D3 04 10 EF C1 40
1318 10 E9 EF 0C 0D 54 68 65 4D
1320 20 45 50 52 4F 4D 20 69 5F
1328 73 20 00 7B FE 00 28 06 75
1330 EF 6E 6F 74 20 00 EF 6F 01
1338 2E 68 2E 20 21 0D 00 FD 5D
1340 E9 3E FF D3 06 AF D3 06 DA
1348 3E FF D3 07 AF D3 07 3E 39
1350 08 D3 04 AF D3 04 EF 0D C4
1358 45 50 52 4F 4D 20 73 74 F5
1360 61 72 74 20 61 64 72 65 76
1368 73 73 3A 20 0D 00 D5 DF 7C
1370 63 DF 64 D1 38 E0 2A 21 5D
1378 0C 7D FE 00 28 0A 2B 3E AD
1380 04 D3 04 AF D3 04 18 F1 FD
1388 7C FE 00 20 F1 D5 DF 63 3D
1390 DF 64 D1 38 02 18 02 FD 08
1398 E9 2A 21 0C 7D D3 05 3E 7E
13A0 C0 D3 04 3E E0 D3 04 F5 34
13A8 C5 06 08 FF 10 FD C1 F1 4C
13B0 3E 80 D3 04 3E FF D3 07 6F
13B8 D3 07 DB 05 BD 20 10 3E B0
13C0 04 D3 04 AF D3 04 3E FF 71
13C8 D3 07 AF D3 07 18 BE EF 03
13D0 0C 0D 0D 0D 54 68 65 20 57
13D8 6C 61 73 74 20 42 59 54 AE
13E0 45 20 69 73 6E 27 74 20 5D
13E8 70 72 6F 67 72 61 6D 6D 60
13F0 65 64 20 21 00 FD E9 FF F2
13F8 FF FF FF FF FF FF FF FF 03

```





Die Mitarbeiter dieser Ausgabe waren:

Michael Bach, [REDACTED], [REDACTED] Stegen - Peter Bentz, [REDACTED], [REDACTED]  
 Mülheim/Ruhr - Günter Böhm, siehe Impressum - Stefan Bürger, [REDACTED], [REDACTED]  
 Tamm-Hohenstange, [REDACTED] - Markus Caesar, [REDACTED], [REDACTED] Leichlingen -  
 Günter Endert, [REDACTED], [REDACTED] Ebringen - Mary Jo Kostya, [REDACTED]  
 [REDACTED], [REDACTED] Frankfurt - Günter Kreidl, siehe Impressum - Johannes C. Lotter,  
 [REDACTED], [REDACTED] Darmstadt, [REDACTED] - D. Maisl, [REDACTED],  
 [REDACTED] Göppingen - Rüdiger Maurer, [REDACTED], [REDACTED] Taunusstein - Wolfgang  
 Mayer-Gürr, siehe Impressum - Klaus Mombaur, [REDACTED], [REDACTED] Nürnberg -  
 Dieter Oberle, [REDACTED], [REDACTED] Vollmersweiler, [REDACTED] - Werner Öhring,  
 [REDACTED], [REDACTED] München [REDACTED] - H.-Martin Pohl, [REDACTED], [REDACTED]  
 Stuttgart 1, [REDACTED] - Christoph Rau, [REDACTED], [REDACTED] Bonn - Karl  
 Trust, [REDACTED], [REDACTED] Rosenthal - Jürgen Weiermann, [REDACTED],  
 [REDACTED] Nideggen - Josef Zeller, neue Adresse siehe Impressum -

Die Autoren tragen die Verantwortung für ihre Beiträge selbst.

Artikel, die besonders durch einen Copyright-Vermerk gekennzeichnet sind, dürfen nicht nachgedruckt oder anderweitig vervielfältigt werden ohne schriftliche Genehmigung des Verlags oder des Authors. Alle anderen Artikel dürfen für jeden unkommerziellen Zweck veröffentlicht werden, vorausgesetzt, es wird als Quelle das NASCOM Journal angegeben.

# LESERBRIEFE

Zuerst möchte ich Ihnen und Ihren Mitarbeitern dafür danken, daß Sie alle es geschafft haben, den Untergang des NASCOM-Journals zu verhindern, und nicht nur das, Sie haben es auch noch fertiggebracht, dem Journal ein Maß an Niveau zu verleihen, welches für mich bei weitem höher liegt, als das einer professionell organisierten Zeitschrift. (Dieses Kompliment müssen wir an die Leser weitergeben, die durch ihre eigentlich professionelle Arbeit mit dem NASCOM und die daraus resultierenden Beiträge für's Journal das alles erst ermöglichen! Red.) Vielen Dank an alle!

Anbei sende ich Ihnen eine Cassette mit mehreren Programmen und Vorschlägen, welche mit Text versehen wurden.

Ich hoffe, daß meine Beiträge interessant genug sind, damit sie bei der Gestaltung des Journals von Nutzen sein können. (Sie sind es! Siehe diese Ausgabe. Red)

Im übrigen würde ich mich bereit erklären, für Interessenten nach dem Kreidl'schen Prinzip unter Einsendung einer Cassette etc. den Disassembler zu kopieren. (Oh mann, ist das ein Satzbau - 'tschuldigung). Hier noch einmal mein Ladeformat: NASCOM1 T2 oder NASSYS.

Der Beitrag von Herrn Pohl hat es mir angetan, und ich möchte hiermit bitten, daß Sie mir ein paar Notizen zur Anpassung an den NASCOM1 schicken. (Siehe Artikel von Herrn Pohl in diesem Heft. Red.)

Jürgen Weiermann, Nideggen

Falls Interesse besteht, kann ich in nächster Zukunft ein Menue-Programm anbieten, das 2K-Speicher beansprucht und ab Adresse A0000 abläuft. Mit diesem Programm ist es möglich, direkt in die einzelnen Systemprogramme incl. Optionen einzusteigen; dazu existieren noch einige zusätzliche sehr nützliche Programme, z.B. ein Druckeransteuerungsprogramm für die Centronics Schnittstelle. Das Druckerprogramm steuert bei mir den Seiko GP80, und es beinhaltet auch ein modifiziertes kleines "g" und "p", welche viel besser lesbar sind als die Original-

buchstaben; denn leider kann der GP80 keine Unterlängen.

Hier ein kleiner Vorgeschmack auf die Verwaltung des Menue Programmes.

(Wenn Sie interessiert sind, lassen Sie es uns wissen ! Red.)

```
Tool-Kit ..... T
Basic Cold / Warm ..... BC / B
Naspen Cold / Warm ..... PC / P
Assembler Cold / Warm .... AC / A
Debugger / Disassembler .. DB / D
2708 / 2716 Programmer ... S or 6
Return to NAS-SYS ..... N
RAM Test ..... R
Clean or Compare Memory .. C
Printer Option ..... X
D
```

```
Disassembler Option Mode .. D
" Direct Mode .. D
Return to Menue ..... M
```

C

```
Clear All ..... A
Clear Range ..... R
Compare 1k Block ..... C
Return to Menue ..... M
```

Bis zur Fertigstellung des Programmes warte ich noch auf NASSYS3, um einige der neuen Routinen mit zu berücksichtigen. Ich hoffe, daß NASSYS3 bald zum Standard wird.

Meine Meinung zum Strichcode: Lieber ein lesbares BASIC-Programm, als ein vielfach platzbeanspruchendes Strichcode Programm, welches ich nur als Lückenfüller betrachte. (Lückenfüller brauchen wir nicht. Die Leser haben Verständnis, wenn wir einmal nur 12 Seiten veröffentlichen. Was halten die anderen Leser vom Strichcode? Wenn er nicht gewünscht wird, lassen wir ihn weg - wie in diesem Heft. Red.)

Ansonsten bin ich mit dem Journal recht zufrieden, vor allem ist eine Niveauanhebung von Heft zu Heft bemerkbar - nur weiter so! Damit mein Zeigefinger nicht zu sehr belastet wird, (der drohende oder der schreibende ? Red.) mache ich jetzt Schluß und verbleibe mit....(Lückenfüller werden nicht gedruckt! Red.)

Rüdiger Maurer, Taunusstein

Das NASCOM-Journal gefällt mir immer besser! Besonders freue ich mich, daß Herr Kreidl die Sprache FORTH vorstellt und Schritt für Schritt ein FORTH-System entwickelt. Ich

habe zwar schon im Frühjahr ein FORTH auf meinem NASCOM geschrieben und werde in vielem bei meiner Version bleiben, aber in eine zweite Version wird sicher auch manche Anregung aus dem Journal mit eingehen. Den Soundgenerator, über den Sie geschrieben haben, werde ich sicher nachbauen, wenn ich die Zeit dazu finde. Aber das Größte war für mich der Artikel von Herrn Pohl im letzten Heft über die hochauflösende Grafik mit Busrequest. Ich bin schon eifrig dabei, die Schaltung (mit Änderungen) nachzubauen und hoffe, daß die Kiste bald funktioniert. Könnten Sie mir wohl freundlicherweise die Adresse und/oder Telefonnummer von Herrn Pohl schicken?

Der beiliegende Artikel mit Listings ist vielleicht auch für andere NASCOMer interessant und etwas für's Journal. (Software Repeat in diesem Heft Red.)

Christoph Rau, Bonn

Die Anschrift von Herrn Pohl finden Sie in der Spalte "Mitarbeiter..." Lassen Sie uns aber bitte wissen, wenn Sie mit Herrn Pohl etwas aushecken, was auch für andere Journal-Leser interessant wäre. Die Grafik fasziniert eine Menge Leser teuflisch! Red.

Ich halte Nr.1-9/81 des NASCOM-Journals, die ich gerade bestellt habe, in Händen. Ich bin freudig überrascht und möchte Ihnen und MKS herzlich gratulieren. Das ist ein guter Schritt voran, die Sache mit Ihnen und Ihren "Fachkollegen".

Ich bin 46 Jahre alt und Fachlehrer an der Schule für Rundfunktechnik, wo ich das Feld der Steuer- und Regeltechnik beackere. Nach einem Fernlehrgang, den ich belegte, um beim Auftauchen der ersten Microcomputer am Ball zu bleiben, war mein Interesse erst richtig geweckt, und nun ging es ans Suchen auf höherer Ebene. P\*T? T\*S80? N\*NOCOMP.? und, und, und... Ich verfiel auf den NASCOM2, weil er mir Garant zu sein schien, 1. komfortabel in Maschinenebene programmierbar zu sein, 2. BASIC zu können und 3. eine professionelle Tastatur zu haben. Natürlich war auch Preis/Leistung o.k. Ich habe den Schritt bis heute nicht bereut, obwohl natürlich mehr Zeilen (aber nicht mehr als 20 wegen der Augen) oder bessere Grafik nicht schlecht wären.

Das war ca.7/80, und seither ist mein Schreibtisch nicht mehr so leicht "abzustauben".(Meiner Frau habe ich die Sache übrigens mit einem Menüprogramm für den wöchentlichen Speisezettel schmackhaft gemacht.)

(Wie machen das andere Leser???? Red.)

Warum habe ich eigentlich das NASCOM-Journal bestellt? Weil M\* und C \*ip zwar für mich teilweise etwas Brauchbares bringen, aber doch auch sehr viel.....Also, ich hoffe, bei Ihnen besser aufgehoben zu sein. Außerordentlich begrüße ich den Vorschlag, nur noch NASSYS zu benutzen. Was mich besonders interessiert sind PIO, UART Steuerungen und ganz besonders Programmertips und -tricks für Z80 und auch BASIC, denn damit können sich doch Amateure am besten gegenseitig befruchten.(Auf zur Befruchtung, liebe Leser! Red.) Es müssen nicht immer ganze Programme sein.

Nebenbei, machen Sie mit Ihrem NASCOMPL ruhig weiter. Etwas menschliches in dieser vielen Technik ist völlig o.k.

Die Leser sollten sich nicht scheuen, Fragen zu Hard- und Software zu stellen (oder weiß jeder alles?). Hier gleich mal einige:

1. Im BASIC-Handbuch S.44 steht die Sache mit LIST oder PRINT und INPUT unter X0 Unterprogramme zu speichern. Damit komme ich schlecht klar! Wer gibt Beispiele an?
2. Wie kann man in BASIC reine Daten von Cassette lesen?

Wenn andere ihre Programme schließlich kostenlos im Journal weitergeben, bin natürlich auch ich dazu bereit. (Hier folgt eine Liste von 4 interessanten Programmen, die wir im Laufe der Zeit veröffentlichen werden. Danke, Herr Mombaur! Red.)

Vielleicht erzählt ab und an auch mal jemand anders über den "Hintergrund". Ich habe nun ja einen Anfang gemacht. Was sind das für Leute, die sich zur NASCOM-Familie gehörig fühlen?

Viel Erfolg wünscht für MKS und NASCOM  
Klaus Mombaur, Nürnberg

An die Redaktion des NASCOM Journals, zunächst einmal ein dickes Lob für das neue Journal. Das Warten hat sich also gelohnt - mit NASCOM scheint es ja nun wirklich zügig aufwärts zu gehen. Dazu auch einmal besten Dank an Herrn Klein, der uns -die NASCOM Anwender- nicht im Stich gelassen hat.



Die Ankündigungen aus England sind ja ebenfalls äußerst vielversprechend (Graphik, CP/M 2.2 etc.). Aber gleichzeitig sollten wir versuchen, das Journal als Sprachrohr der NASCOM-Anwender in Deutschland zu etablieren. Dazu sollte das JOURNAL evtl. ein wenig mehr in die Öffentlichkeit treten (Fachzeitschriften). Dies jedoch müssen die Leser kräftig mittragen.

Mit den folgenden Anregungen hoffe ich, einen Teil dazu beitragen zu können.

#### Punkt 1 Monitor

Meine Meinung: unbedingt Einigung auf NASSYS3  
Mein Vorschlag: Dokumentation der Unterschiede (im Assembler) zwischen Version 1 und 3 (Die Rechte dazu werden gerade mit LUCAS L. geklärt, Red.)

#### Punkt 2 Assembler Listings

Assembler Listings sind einerseits zum Verständnis und zum Programmieren unbedingt nötig, andererseits nehmen sie im JOURNAL einfach zuviel Platz weg.

Mein Vorschlag: Versendung der Listings an Interessenten gegen Unkostenbeitrag. Statt dessen Programmbeschreibungen und evtl. Ablaufpläne. (Dies wurde mit Recht schon mehrfach vorgeschlagen. Es bringt aber eine Menge Mehrarbeit mit sich! Wir werden es in der Redaktion besprechen. Red.)

#### Punkt 3 Forth

Meine Meinung: hervorragend!!!

Tip: billiger wissenschaftlicher Taschenrechner o.ä. als Arithmetikprozessor + FORTH-Routinen, um diesen anzusprechen. Wer macht's?

#### Punkt 4 PIO-BUS

Den vorgeschlagenen 31 poligen BUS (JOURNAL Sept.) finde ich recht gut; - Steckverbindung nach DIN 41617. (Danach haben sich schon einige Leser erkundigt. Red.)

#### Punkt 5 Projekte

Nicht jeder hat ein fertiges Programm oder eine Bauanleitung für das JOURNAL parat. Aber wir sollten nicht jeder für sich das Rad immer wieder neu erfinden. Also: mehr Kommunikation!

1. Vorschlag: Leserliste mit Adressen herausgeben. (Wurde inzwischen von Herrn Öhring übernommen. Red.)

2. Info Ecke "Wer arbeitet an was?" einrichten. (Wurde in diesem Heft übernommen. Danke! Red.)

#### Punkt 6 Leserwünsche

Vorschlag: Sammeln und etwa in Form einer "Hit-Parade" in Blöcken zur Diskussion stellen.

mit freundlichen Grüßen

Stefan Bürger, Tamm-Hohenstange

## **DEBUG und TOOLKIT von Günter Böhm**

Häufig kommen Anfragen, was denn das TOOLKIT alles könne oder was eigentlich DEBUG sei. Um weiteren Fragen zuvorzukommen, hier eine kurze Vorstellung beider Programmierhilfen.

#### NAS-DEBUG V 3.1

DEBUG vereinfacht das Schreiben und Testen von Maschinenprogrammen. Es hat eine Länge von 1 K und arbeitet mit NAS-SYS und dem Disassembler NAS-DIS.

Die Haupteigenschaft von DEBUG ist eine umfassende Single-Step-Anzeige, die automatisch die Anzeige von NASSYS ersetzt. Sie kann mit einer zusätzlichen CRT-Seite benutzt werden, die es ermöglicht, Programme, die den Bildschirm benutzen, in Einzelschritten zu testen. (Es ist wirklich teuflisch, wenn beim normalen single step die erwarteten Bildschirmanzeigen nach oben verschwinden!)

Alle Z80 Register (einschließlich des zweiten Registersatzes AF', HL' etc.) und das 1FF2 Flip-Flop werden angezeigt. Der mnemonische Code für den nächsten Befehl wird neben dem Programmzähler angezeigt, und den sechs Hauptregisterpaaren folgen zehn fortlaufende Speicherplätze. Das fünfte Byte ist dasjenige, auf das das Register zeigt.

Mit den NASSYS Cursor-Kontrollmöglichkeiten können Speicherplätze und Registerwerte nach Belieben verändert werden.

DEBUG enthält auch einen FIND-Befehl, der die Suche nach einem String bis zu 8 Bytes im Speicher ermöglicht. Die Bytes können als ASCII, Hex oder als "wilde" Bytes definiert werden. Wenn der angegebene String gefunden ist, wird er angezeigt und kann daraufhin im Speicher geändert werden.

Auch eine einfache Methode, um die Opcodes und mnemonischen Befehle mit Hilfe des Disassemblers in einem Programm Schritt für Schritt anzuzeigen, ist vorgesehen. Zudem

stehen dem Benutzer zehn zusätzliche Befehle zur Verfügung, um die Möglichkeiten des Betriebssystems zu erweitern. Die Anfangsadresse des DEBUG ist #C000. Es benötigt den Disassembler bei #C400. Der Bereich von 0E00 bis 0EFF wird von Disassembler und DEBUG als Arbeitsspeicher benötigt und sollte vom Anwender nicht benutzt werden, wenn DEBUG läuft.

#### BASIC TOOLKIT

Das TOOLKIT ist eine Hilfe beim Erstellen von BASIC-Programmen. Nach dem Start des TOOLKIT erscheint auf dem Bildschirm das Menu. Sie können nun die gewünschte Funktion auswählen. Folgende Befehle können gewählt werden:

**RENUMBER** Zeichennummern des Programms können verändert werden.

#### STEP

BASIC-Programme können schrittweise abgearbeitet werden. Bestimmte Variable werden angezeigt.

#### FIND

Ein String (bis max.48 Zeichen) wird im Programmtext gesucht.

#### HELP

Nach Eingabe von Help wird die fehlerhafte Zeile angezeigt, und der Cursor zeigt auf den Befehl, der den Fehler verursacht hat.

#### HEX

Bis zu 10 Hex-Zahlen werden in Dezimalzahlen umgewandelt. Mit der Editierfunktion kann man sie leicht in ein DATA-Statement übernehmen.

#### INKEY & RINK

Dies sind keine zusätzlichen Befehle, sondern Unterprogramme, die mitUSR aufgerufen werden können. INKEY fragt die Tastatur ab, bis eine Taste gedrückt wurde. RINK fragt die Tastatur einmal ab und übergibt bei gedrückter Taste den ASCII-Wert, andernfalls 0.

#### KEYBOARD REPEAT

Beim Start des TOOLKIT wird diese Funktion automatisch eingeschaltet. (Mit NASSYS 3 ist sie nicht mehr nötig).

#### MON

Dieser Befehl bewirkt einen Rücksprung zu NAS-SYS.

#### APPEND

Mit diesem Befehl kann ein BASIC-Programm an ein bereits im Speicher befindliches angehängt werden.

#### AUTO LINE NUMBER

Hiermit werden automatisch Zeilennummern erzeugt. Die Parameter mm (Beginn) und nn (Zeilenabstand) können mit eingegeben werden.

#### DELETE

Der Befehl löscht die Programmzeilen von mm bis nn.

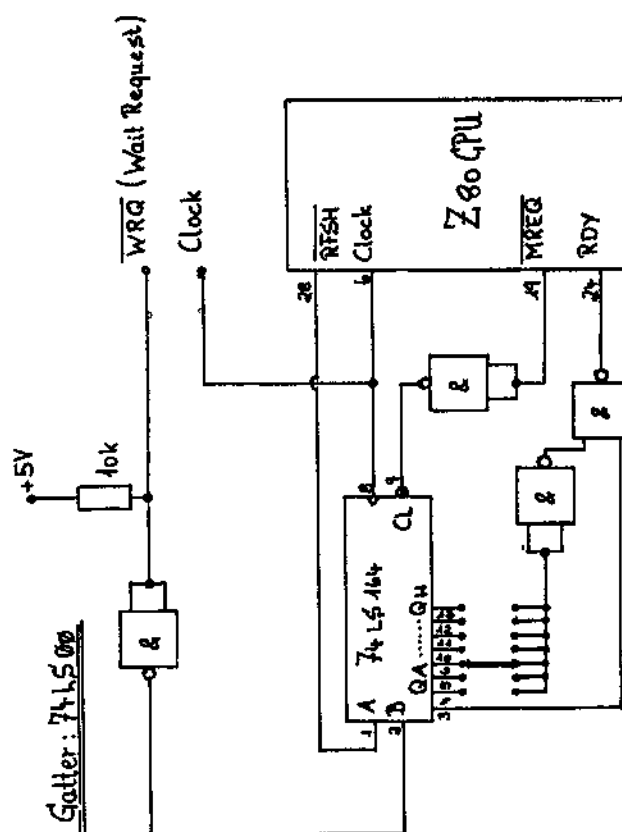
#### DUMP

Mit DUMP werden numerische und String-Variablen (außer Variablenfeldern) aufgelistet. Das TOOLKIT stellt auch ein Unterprogramm für Drucker mit Handshake zur Verfügung. Seine Startadresse ist beliebig, sollte aber B000 sein, dann kann man es mit der "JUMP on RESET-Funktion" des NASCOM 2 aufrufen.

Sowohl DEBUG wie auch TOOLKIT bieten also eine Menge von Komfort für die Erstellung und vor allem für das "Debugging" von Programmen. Vielleicht sind diese "alten" Programme dadurch für viele aktuell.

## WAIT-ZYKLEN v. Stefan Bürger

Folgende Schaltung zeigt einen SLP Slow Memory Controller. Bei  $\overline{WREQ}$  auf GND wird durch  $\overline{MREQ}$  eine einstellbare (1-7) Anzahl von Wait-Zyklen zugefügt.



```

1 PRINT" ,l*****
****OUTOUTOUT* ---
2 PRINT" ----B*OUTBOUT*B--- 'l*****
*****OUTOUTOUTOUT'
3 PRINT" 'BBBBB--*OUT_OUT*--0-B'0' 10U
T1
4 PRINT" B 088 0 1OUT1
5 PRINT" 0=100TOUTOUTOUTOUTO
UTOUTOUTOUTOUTOUTOUT*
6 PRINT" 0B=100TOUTOUTOUTOUTO
UT1===='BBB'
7 PRINT" B'B
9 PRINT"

10 PRINT"Space the final frontier...
11 PRINT"These are the voyages of the starsh
ip
12 PRINT" USSS Enterprise NCC 1701
13 PRINT"on its five year mission to seek ou
t new
14 PRINT"life and new civilizations and to b
oldy go
15 PRINT"where no man has gone before ..."
16 DOKE4100,DEEK(3201)+1949
17 DIM X(7),Y(7)
18 DATA12,1,12,4,23,7,2,3,36,6,45,2,26,2
19 FORI=1TO7:READY(I),X(I):NEXTI:A$=""
20 FORJ=1TO7:SCREENY(J),X(J):PRINTA$
21 FORT=1TO30:NEXTT:IFA$="" THENA$="":GOTO2
3
22 FORT=1TO30:NEXTT:IFA$="" THENA$=""
23 NEXTJ
28 A=USR(0):IFAU*0THEN 156
29 X=RND(1):GOTO20
30 SCREEN1,1:RETURN
31 GOSUB30:FORI=1TO4:FORR=1TO47:PRINT " ";:NE
XTR
32 NEXTI:GOSUB30:RETURN
156 CLS:CLEAR:DIMG(9,12),Q(24,10)
157 INPUT"Do you need instructions ";W$
159 ILEFT$(W$,1)="Y"THEN 3900
220 PRINT;"Difficulty (1-9), Commander ";
230 INPUTZ$:IFZ$="" THENZ$="9"
233 PRINT
235 Z8=INT(ABS(VAL(Z$)))
242 IFZ8=1THEN220
243 IFZ8=9THEN220
245 Z7=10-Z8
250 PRIT"Please stand by , Commander"
251 PRINT"...the Galaxy is forming"
265 GOSUB3265
330 P1=3000*Z7:P2=1500*Z7;V9=.3:T1=10
350 E1=FR(23):E2=FNR(09):Q1=FNR(8):Q2=FNR(1
1)
360 RIN"Type 'HE' for help when needed"
370 GOSUB2020
380 GOSUB30:GOSUB2209
390 GOSUB1790:REM Desaster?
395 IFQ(E1,E2)=3THENPRINT:GOTO1710
400 INPUT" What is your command ";C1$
405 IFK1*K2=0THEN410
406 IFH2*0THEN01=B1:02=B2:GOSUB6000:REM Klin
gon
407 IFH2=0THEN01=E1:02=E2:GOSUB6000:REM move
410 C$=LEFT$(C1$,2):IFLEN(C1$)U3THEN420
411 G$=MID$(C1$,3,1):IFASC(G$)U49THEN420
412 IFASC(G$)*56THEN420
413 IFV9U.1THENV9=0:GOTO600
414 IFV9*.5THENV9=1
415 V9=VAL(G$)/10:GOTO420
420 IFP1+P2U*0THEN440
430 PRINT"Captain,the Ship is out of fuel !"
:GOTO 1720
440 IFC$="SL"THENGOSUB30:GOTO380
450 IFC$U"SG"THEN470
460 GOSUB30:GOSUB2130:GOTO390
470 IFC$U"SN"THEN540
480 GOSUB31
481 FORJ=Q2+1TOQ2-1STEP-1:PRINTJ;" ";
483 FORI=Q1-1TOQ1+1
490 IFI*J*0THEN500
493 IFI=9THEN495
494 GOTO500
495 IFJ=12THEN530
500 IFG(I,J)U1000THENG(I,J)=G(I,J)+1000
510 IFI=Q1THENIFJ=Q2THENPRINT"+";:GOTO520
513 PRINT"";
516 IFG(I,J)*1000THENPRINTG(I,J)-1000;"!";:G
OTO530
520 PRINTG(I,J)-1000;"!";
530 NEXTI:PRINT:NEXTJ
531 PRINTTAB(7);Q1-1;" ";Q1;" ";Q1+1;
535 PRINT" Code = 'lebK*!'";:GOSUB2210:GOT
0390
540 IFC$U"IM"THEN560
550 V9=.3:GOTO380
560 IFC$U"WP"THEN630
565 GOSUB31
570 INPUT"Warp Faktor (1-8) ";D1:D1=INT(D1)
580 IFD1U0THEN590
581 IFD1=8THEN590
582 GOTO600
583 IFD1=0THEN600
585 V9=D1:GOTO610
590 PRINT"Invalid Warp Speed requested,Sir..
";:GOTO570
600 V9=D1:IFV9=0THENPRINT"Aye,Sir ... all en
gines stopped"
610 IFV9*0THENPRINT"Aye,Sir ...Warp ";V9;"
set"
615 FORI=1TO500:NEXT
620 GOSUB30:GOTO380
630 IFC$U"AF"THEN690
635 GOSUB31
640 PRINT"Total fuel remaining: ";P1+P2
650 INPUT"Amount to allocate to screens,Sir
";D1
660 IFD1=0:P1+P2THEN670
663 IFD1=0THEN680
670 PRINT"Allocation Impossible,Sir ...":GOT
0650
680 P2=P1+P2-D1:P1=D1:PRINT"Aye,aye,Sir"
681 GOSUB30:GOTO380
690 IFC$U"DO"THEN780
700 D1=0:GOSUB31
710 FORI=E1-1TOE1+1:FORJ=E2-1TOE2+1:IFI*JUOT
HEN730
720 IFI=24THEN730
721 IFJU12THENIFQ(I,J)=2THEND1=1
730 NEXTJ:NEXTI:IFD1U*0THEN760
740 PRINT"Dock,Sir ??? With what?":GOSUB30:G
OTO380
760 V9=0:P1=3000*Z7:P2=1500*Z7:T1=10
761 PRI"Docked, Sir":GOSUB30:GOTO380
780 IFC$U"HE"THEN890
785 GOSUB31:CLS
790 PRINT" ***** Star Trek Commands ***
**
795 PRINT:PRINT
800 PRINT"HE, this Help message/DO. Dock at
Star Base
810 PRINT"FP,fire Phasers /FT,fire Phot
on torpedoes"
820 PRINT"SL,scan local Quad /SN,scan neig
hbor Quads"
830 PRINT"SG,scan Galaxy map /AF,allocate
fuel"
840 PRINT"IM,set Impulse Speed /WP,set Warp
850PRINT"BC,Battle Computer /MD,move the
Ship down"
860 PRINT"MU,move the Ship up /MR,move the
Ship right"
870 INPUT"ML,move the Ship left/ (struggle o
n = ENTER) ";0$
880 GOTO380
890 IFC$U"FP"THEN1080
900 GOSUB31:IFC9=2THEN915
910 PRINT"No target, Sir":GOTO390
915 INPUT"Phasers bearing, Sir ";W
920 INPUT"Number of fuel units to fire, Sir
";D

```

```

930 IFD°0THENIFDμ=P2THEN950
940 PRINT"Invalid response, Sir":GOTO920
950 D1=SGR((K2-E2)2+(K1-E1)2):D2=D/D1:P2=P
2-D
953 GOSUB8000:GOSUB30
956 IFH1=1THEN380
960 GOTO5000
961 S8=S8-D2:IFS8μ0THEN970
962 IFS8μ0THEN970
965 GOTO1000
970 PRINT"*** Enemy destroyed ***"
971 FORT=1T0400:NEXTT
980 G(Q1,Q2)=G(Q1,Q2)-10:K9=K9-1:C9=1:Q(K1,K
2)=0
981 K1=0:K2=0:M=0:N=0
985 GOSUB1990
990 IFK9=0THEN1680
995 GOTO380
1000 GOSUB8002:IFH1=1THEN380:REM Kling schie
sst?
1003 D1=(K2-E2)2+(K1-E1)2
1005 D=S8:S8=S8/4:D=D-S8
1010 GOSUB31:PRINT"*** Enemy attacks ***"
1011 FORT=1T0400:NEXTT
1020 P1=P1-D/D1:PRINTD/D1;"unit hit from ene
my"
1030 PRINT" Sensors report,that enemy is dow
n to"
1031 PRINTD;"units of fuel"
1070 IFP1μ0 THEN1710
1075 FORI=1T02000:NEXTI:GOSUB30:GOTO380
1080 IFC$μ"BC"THEN1180
1085 GOSUB6000
1090 IFK3-E1μ°0THEN1120
1100 IFK4°E2THEND=90:GOTO1150
1105 D=270
1110 GOTO1150
1120 D=(K4-E2)/(K3-E1):D=ATN(D)*180/3.14159
1130 IFK3μE1THEND=D+180
1140 IFDμ0THEND=D+360
1150 GOSUB31:PRINT"*** Battle Computer ***"
1160 PRINT"The bearing of the enemy is";D;"d
egrees"
1170 GOSUB6000:C$="":GOTO1000
1180 IFC$μ"FT"THEN1320
1181 GOSUB31:IFC9=2THEN1190
1183 GOTO1192
1190 IFT1μ°0THEN1210
1191 GOTO1200
1192 PRINT"Where is Your enemy,Sir ?"
1193 FORI=1T0500:NEXTI:GOSUB31
1194 GOTO380
1200 PRINT"No Photon torpedoes left, Sir"
1201 GOSUB30:GOTO380
1210 INPUT"Torpedo bearing,Sir ";W
1220 T1=T1-1:GOSUB8000:IFH1=1THEN380
1240 D1=K1-E1:IFD1μ°0THEN1270
1250 IFK2°E2THEND2=90:GOTO1260
1255 D2=270
1260 GOTO1300
1270 D2=ATN((K2-E2)/D1)*180/3.14159
1280 IFD1μ0THEND2=D2+180
1290 IFD2μ0THEND2=D2+360
1292 IFD2=360THEND2=0
1294 IFW=360THENW=0
1300 IFABS(W-D2)μ2.5THEN970
1310 PRINT"Photon torpedo missed ";
1315 PRINT"by";ABS(W-D2);"degrees , Sir":GOT
O1000
1320 IFLEFT$(C$,1)μ°"M"THEN1670
1380 GOSUB31:C1=E1:C2=E2:IFV9°=1THENGOSUB153
0:GOTO1580
1390 FORI=1T0500:NEXTI:PRINT"Impulse engines
engaged, Sir"
1400 P2=P2-20*V92:IFP2μ0THENP2=0
1410 IFP2μ°0THEN1430
1420 PRINT"Insufficient engine power, Sir"
1421 GOSUB30:GOTO380
1430 GOSUB31:GOSUB1530
1440 C2=C2+D2*10*V9

```

```

1450 C1=C1+D1*10*V9
1470 IFC1μ=0THENC1=1:GOTO1480
1471 IFC1°=24THENC1=23:GOTO1480
1472 IFC2μ=0THENC2=1:GOTO1480
1473 IFC2°=10THENC2=09:GOTO1480
1474 GOTO1491
1480 PRINT"Enterprise blocked at edge of sec
tor, Sir"
1491 FORC=E1TOC1STEPSGN(C1-E1)
1492 FORC=E2TOC2STEPSGN(C2-E2)
1494 IFQ(E1,E2)=Q(E,C)THEN1496
1495 IFQ(E,C)μ°0THENIFQ(E,C)μ5THEN1500
1496 NEXTC:NEXTE:GOTO1510
1500 PRINT"Enterprise blocked by Objekt, Sir
"
1505 C1=E-SGN(E-E1):C2=C-SGN(C-E2)
1510 Q(E1,E2)=0:E1=C1:E2=C2:Q(E1,E2)=1
1520 IFC9=2THEN 1000
1525 GOSUB31:GOTO380
1529 GOSUB 1535:GOTO1580
1530 GOSUB30:REM
1535 D1=0:D2=0
1540 IFC$="MD"THEND2=-1
1550 IFC$="ML"THEND1=-1
1560 IFC$="MR"THEND1=1
1570 IFC$="MU"THEND2=1
1575 RETURN
1580 PRINT"Warp engines engaged, Sir"
1590 P2=P2-20*V92:IFP2μ0THENP2=0
1600 IFP2=0THEN1420
1610 C1=Q1+D1*V9:C2=Q2+D2*V9
1620 IFC1μ1THENC1=1:GOTO380
1630 IFC1°8THENC1=8:GOTO380
1640 IFC2μ1THENC2=1:GOTO380
1650 IFC2°11THENC2=11:GOTO380
1655 IFG(Q1,Q2)μ1000THENG(Q1,Q2)=G(Q1,Q2)+10
00
1660 Q1=C1:Q2=C2:GOSUB2020:GOTO380
1670 PRINT"Invalid command, Sir":GOSUB30:GOT
O380
1680 PRINT"Congratulations !!! You have got
them !!"
1690 PRINT"The Klingons have been defeated a
nd the univers"
1700 PRINT"is once again safe from their ter
ror !";END
1710 PRINT"You have been destroyed by the en
emy battle cruiser !"
1720 PRINT"You were the galaxy's last chance
.All is lost."
1730 PRINT"The Klingons will now rule the ga
laxy";END
1740 DEFFNR(N)=INT(RND(1)*N+1):RETURN
1760 DEFFNP(N)=SGN(SGN(N/10-RND(1))+1)
1780 RETURN
1790 D=INT(RND(1)*100)
1800 IFD°20THENRETURN
1810 IFDμ16THENRETURN
1820 GOSUB31:PRINT"***Warning -- Condition Ale
rt**"
1825 ONINT(RND(1)*6+1)GOSUB1830,1840,1870,18
90,1910,1930:GOTO380
1830 PRINT"Enterprise hit by secret weapon -
all weapon energy's gone"
1835 P2=0:RETURN
1840 PRINT"Enterprise hit by solar storm sc
reens reduced"
1850 P1=P1-RND(1)*100:IFP1μ0THENP1=0
1860 PRINT"Screen power now at";P1;"units":R
ETURN
1870 PRINT"Enterprise hit by secret weapon -
all torpedoesdestroyed"
1880 T1=0:RETURN
1890 PRINT"Enterprise hit by secret weapon -
all weapon energy gone"
1900 P2=0:RETURN
1910 PRINT"Enterprise caught in ion storm -
energy sourcesdepleted"
1920 P1=P1/2:P2=P2/2:RETURN
1930 PRINT"Enterprise hit by Plasma Bomb - e
nergy sources depleted"

```

```

1940 IFP1μ100THENP1=0
1950 P1=P1/10;P2=P2/10;T1=INT(T1/2);RETURN
1960 C1=FNR(23);C2=FNR(09)
1970 IFQ(C1,C2)μ°0THEN1960
1980 RETURN
1990 ONC9GOTO1992,1991
1991 C$(C9)="*** RED ***";RETURN
1992 C$(C9)="*** Green ***";RETURN
2020 FORI=1TO23;FORJ=1TO09
2030 Q(I,J)=0;NEXTJ;NEXTI
2035 E1=FNR(23);E2=FNR(09)
2036 Q(E1,E2)=1
2040 IFG(Q1,Q2)μ1000THENG(Q1,Q2)=G(Q1,Q2)+10
00
2050 K=G(Q1,Q2)-1000
2060 K1=INT(K/100);K=K-K1*100;IFK1=0THEN2080
2070 GOSUB1960;Q(C1,C2)=2;B1=C1;B2=C2;H2=1:C
9=2
2080 K1=INT(K/10);K=K-K1*10;C9=1;IFK1=0THEN2
110
2090 GOSUB1960
2095 IFABS(E1-C1)μ3THENIFABS(E2-C2)μ3THEN209
0
2096 IFABS(B1-C1)μ3THENIFABS(B2-C2)μ3THEN209
0
2097 K1=C1;K2=C2;Q(K1,K2)=3;C9=2;S8=200*Z8
2100 IFFNP(1)=1THENS8=S8*2
2105 IFK=0THENRETURN
2110 FORI=1TOKSTEP.5;GOSUB1960;Q(C1,C2)=4;NE
XTI
2120 RETURN
2130 CLS;PRINTTAB(15);"*** Galaxy Map ***"
2140 FORJ=Q2+1TOQ2-1STEP-1
2141 FORI=Q1-1TOQ1+1
2142 IFG(I,J)μ1000THENG(I,J)=G(I,J)+1000
2143 NEXTI;NEXTJ
2146 FORJ=11TO1STEP-1
2147 IFJ°9THENPRINTJ;" ";:GOTO2149
2148 PRINTJ;" ";
2149 FORI=1TO8
2150 K=G(I,J)-1000;IFKμ0THEN2185
2160 IFI=Q1THEN2180
2170 GOTO 2185
2180 IFJμ°Q2THEN2185
2184 A$="+ ";GOSUB2193;GOTO2187
2185 GOSUB2193
2187 IFI=8THEN2188;PRINTCHR$(17);:A$=";"+"A$
2188 PRINTTAB(I*4+1);A$;:A$=""
2189 NEXTI;PRINT;NEXTJ
2190 PRINTTAB(6);"1 __ 2 __ 3 __ 4 __ 5 __ 6
7 __ 8"
2191 PRINTTAB(5);"( 'b' = 100 P. 'k' = 10 P.
,* = 1P. )
2192 GOTO2206
2193 IFKμ0THENA$=A$+" ";:RETURN
2194 IFLEN(A$)=0THEN2196
2195 A$=LEFT$(A$,LEN(A$)-1)
2196 A$=A$+STR$(K)
2197 IFMID$(A$,2,1)=" "THENA$=LEFT$(A$,1)+RI
GHT$(A$,LEN(A$)-2)
2200 IFKμ100THENA$=A$+" "
2201 IFKμ10THENA$=A$+" "
2203 A$=A$+" ";:RETURN
2206 PRINT"Enterprise ( ) ='+'";:RETURN
2209 GOSUB30;GOSUB2340
2210 H2=0;FORJ=09TO1STEP-1;FORI=1TO23
2240 ON Q(I,J)GOTO2290,2300,2310,2320,2280
2280 PRINT " ";:GOTO2330
2290 PRINT " e";:GOTO2330
2300 PRINT " b";:H2=1;GOTO2330
2310 PRINT " k";:GOTO2330
2320 PRINT " *";
2330 NEXTI;PRINT;NEXTJ;RETURN
2340 GOSUB30;PRINT"Quadrant:";Q1;" ";Q2;
2344 PRINTTAB(22)"Condition: ";:GOSUB1990;PR
INTC$(C9)
2350 PRINT"Speed : Warp";V9;
2360 PRINTTAB(22)"Starbases:";S9;" "
2370 PRINT"Screen:";P1;
2380 PRINTTAB(22)"Engine/Phasers:";P2;" "
2390 PRINT"Photon torpedoes:";T1;
2400 PRINTTAB(22)"Klingons remaining:";K9;"
"
2420 RETURN
3265 GOSUB1740;GOSUB1760
3270 K9=0;S9=0;FORI=1TO8;FORJ=1TO11
3290 K8=FNR(1)*FNP(8);K7=FNP(Z8*.20)
3300 G(I,J)=FNR(9)*FNP(9)+K8*10+K7*100
3310 K9=K9+K8;S9=S9+K7;NEXTJ;NEXTI
3320 IFS9=0THEN3270
3330 RETURN
3900 FORI=1TO14;READX$;NEXTI
3901 FORI=1TO8;READX$;FORJ=1TOLEN(X$)
3905 FORT=1TO20;NEXTT
3910 IFX$="S"THEN220
3915 PRINTMID$(X$,J,1);:NEXTJ
3920 PRINT;NEXTI
3921 A=USR(0);IFA=0THEN3921
3930 GOTO3901
4000 DATA"Sie sind Commander der Enterprise
und
4010 DATA"sollen alle Klingonen in der Galax
is
4020 DATA"vernichten.Es stehen Ihnen dazu
4030 DATA"Phaser & Photonentorpedos zur
4050 DATA"Verfuegung.(Bei den Photonentorped
os
4055 DATA"und den Phasern muss der Abschussw
inkel"
4060 DATA"auf 2.5 Grad genau , dem folgendem
4070 DATA"Schaubild entsprechend angegeben w
erden)"
4080 DATA" 90
4090 DATA" 1
4100 DATA" 180 μ- e -° 0 ( e = Enterprise
)
4110 DATA" 1
4120 DATA" 270
4121 DATA"Wenn entlang des Schusswinkels im
4122 DATA"Abstand von 2.5 Grad ein Objekt is
t &
4123 DATA"dies ein '*' ist,wird der Schuss u
ngueitig
4124 DATA"dies ein 'b' ist,wird 'b' getroffe
n !!
4130 DATA"Die Wirkung der Phaser nimmt mit d
em
4140 DATA"Quadrat der Schussentfernung ab !!
!
4160 DATA"Einschlagende Phaserstrahlen reduz
-
4170 DATA"ieren die Restegergie ('fuel').
4180 DATA "('fuel'='weapon/engines'+ 'screens'
)
4210 DATA"Wenn man von einem Quadranten in
4220 DATA"einen der naechsten gelangen will,
4230 DATA"muss der Wharp-Antrieb benutzt wer
den
4250 DATA"Wharp 2 bringt Sie 2 Quadranten we
iter
4300 DATA"Kommandos bei'What is your command
?';
4310 DATA" Wharp :WP 1 ImpulsA. ;IM
4320 DATA"z.B: MD7 (1-8) 1 z.B: ML4 (1-5)
4330 DATA"=move down 7 Q. 1 =move left 4
4340 DATA" MR MD ML MU
4350 DATA"move right down left up
4360 DATA"BC=Kampfcomputer,liefert
4361 DATA" hauptsaechlich nur den
4362 DATA" Abschusswinkel fuer Phaser,
4363 DATA" derweil der Klingone schon
4364 DATA" angreift ( Sie sollen ja nicht
4365 DATA" nur Vorteile von der Einricht-
4366 DATA" ung des BC haben )
4370 DATA"AF=Kraftstoffbereitstellung fuer
4380 DATA" Schutzfelder
4390 DATA"DO=Andocken an Starbasis (nur unte
r
4400 DATA" ImpulsA. & wenn direkt neben de
r
4410 DATA" Enterprise)
4413 DATA" Starbases dienen zur Generalueb
er-
```

```

4416 DATA"holung der Enterprise)
4420 DATA"SL=Bringt
4430 DATA"den lokalen Quadrant ins Bild
4440 DATA"SN=Diagramm der umliegenden 8 Quad
r.
4450 DATA"SG=Diagramm der Galaxis
4460 DATA"Symbole
4470 DATA" 'e' = Enterprise
4480 DATA" 'k' = Klingone
4490 DATA" 'b' = Starbasis
4500 DATA" '*' = Stern
4520 DATA"HE=HELP-listet alle moeglichen Kom
-
4530 DATA" mandos auf, falls sie nicht weit
er
4540 DATA" wissen.
4550 DATA" (Alles weitere geht aus dem Spi
el
4560 DATA" hervor)
4561 DATA"1BBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBB
BBBBBBBB1
4562 DATA"1 ACHTUNG : Die Klingonen versuch
en zu-
4563 DATA"1 erst vorhandene Starbasen zu ram
men,
4564 DATA"1 danach erst die Enterprise. Wenn
die
4565 DATA"1 Enterprise jedoch neben ihnen st
eht,
4566 DATA"1 wird sie auch gerne gegen Starba
sen
4567 DATA"1 'eingetauscht' ! --° desha
lb :
4568 DATA"1 ABSTAND HALTEN !
4569 DATA"1
4570 DATA"1BBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBB
BBBBBBBB1
4571 DATA"1 ACHTUNG : Um einen Klingonen zu
tref-
4572 DATA"1 fen, muessen Sie auf d i e Koordi
naten
4573 DATA"1 zielen, die der Klingone im naech
sten
4574 DATA"1 Zug erreicht ! Wenn Sie den BC
benut-
4575 DATA"1 zen, gelten die durch den BC ausg
erech-
4576 DATA"1 neten Winkel schon fuer den vorr
aus-
4577 DATA"1 sichtlichen Standort des Klingon
en
4578 DATA"1 nach dem 2. Zug ( Sie brauchen je
1 Zug
4579 DATA"1 um BC & A FT 'o. FP Ü ausfuehren
zu
4581 DATA"1 lassen ).
4585 DATA"1
4586 DATA"1 S"
5000 V1=K1-E1:IFV1µ°THEN5010
5003 IFK2°E2THENV2=90:GOTO5010
5005 V2=270:GOTO5020
5010 IFK1-E1=0THENN=.000001
5012 V2=ATN((K2-E2)/N)*180/3.14159
5013 IFV1µ°THENV2=V2+180
5016 IFV2µ°THENV2=V2+360
5018 IFW=360THENW=0
5019 IFV2=360THENV2=0
5020 IFABS(V2-W)µ2.5THEN5050
5030 PRINT"Phaser missfired by";ABS(V2-W);"d
egrees"
5040 PRINT", Sir !":GOTO1000
5050 PRINT"The bearing of the Phasers was ri
ght, Sir"
5060 GOTO961
6000 F=SGN(O1-K1):FORM=K1+FTOK1-FSTEP-F
6010 G=SGN(O2-K2):FORN=K2+GTOK2-GSTEP-G

```

```

6011 IFMµ1THENM=1:N=N+F:GOTO6013
6012 IF M°23THENM=23:N=N-F
6013 IF N°09THENM=09:M=M-G:GOTO6020
6014 IF Nu1THENN=1:M=M+G
6020 IFQ(M,N)°=4THEN6060
6029 IFQ(M,N)=1THENQ(K1,K2)=0:Q(M,N)=0:GOTO1
710
6030 IFQ(M,N)=2THENIFCµ°"BC"THEN6070
6035 IFK1=MTHENIFK2=NTHEN6060
6040 IFQ(M,N)=0THEN6080
6060 NEXTN:NEXTM:RETURN
6070 G(Q1,Q2)=G(Q1,Q2)-110:H2=0:S9=S9-1:K9=K
9-1
6071 Q(K1,K2)=0:K1=0:K2=0:M=0:N=0:Q(O1,O2)=0
6072 C9=1:RETURN
6080 IFCS="BC"THENK3=M:K4=N:M=0:N=0:RETURN
6090 Q(K1,K2)=5:Q(M,N)=3:K1=M:K2=N:RETURN
6990 END
7000 CLS:FORI=1TO1000:NEXT:SCREEN12,7
7002 PRINT"Kass.-Interface auf 300 Bd?"
7006 X=USR(0):IFX=0THEN7006
7009 FORI=1TO9:POKE3017+I,48+I:CSAVE"S":NEXT
7010 END
8000 REM Enterprise schiesst
8001 A1=E1:A2=E2:Z1=K1:Z2=K2:GOTO8009
8002 REM Klingone schiesst
8003 A1=K1:A2=K2:Z1=E1:Z2=E2
8009 H1=0:FORI=A1TOZ1STEPSGN(Z1-A1)
8010 FORK=A2TOZ2STEPSGN(Z2-A2)
8020 IFE=A1 THENIFK=A2 THEN8050
8021 IFE=Z1 THENIFK=Z2 THEN8050
8022 GOSUB8200
8050 NEXTK:NEXTE:RETURN
8200 IFQ(E,K)=0THENRETURN
8201 N=E-A1:IFNu°THEN8210
8202 IFZ°A2 THENM=90:GOTO8210
8204 M=270:GOTO8220
8210 IFN=0 THENN=.000001
8211 M=ATN((K-A2)/N)*180/3.14159
8213 IFNu° THENM=M+180
8216 IFMu° THENM=M+360
8218 IFW=360 THENW=0
8219 IFM°=360 THENM=M-360
8220 IFA1=K1 THENM=W
8221 IFABS(M-W)°=2.5 THEN8260
8240 H1=1:REM Marker ob Hindernis beim Schus
s
8250 IFQ(E,K)=2 THENQ(E,K)=0:G(Q1,Q2)=G(Q1,Q2
)-100
8251 S9=S9-1:H2=0
8255 REM Enterprise schießt Starbase ab
8260 RETURN

```

```

° = >
µ = <
s = @
£ = #

```

## SYSTEMGRUNDLAGEN von Josef Zeller

Auf Wunsch verschiedener Leser versucht Josef Zeller in folgendem Artikel die Funktionen von DBDR und M1 etwas zu erhellen. Wenn Sie Fragen zur Hardware und damit zusammenhängenden Grundlagen haben, wenden Sie sich direkt an Herrn Zeller.

Jeder Eingang eines ICs stellt für die treibenden Ausgänge der CPU eine Belastung dar. Da die CPU-Ausgänge nicht genügend

Treiberleistung aufbringen, um in einem größeren System viele andere ICs anzusteuern, müssen die Busleitungen durch Treiber gepuffert werden. Für unidirektionale Leitungen (Daten fließen nur in eine Richtung), also Adreßleitungen (A0-A15) und Controlleitungen ( $\overline{MREQ}$ ,  $\overline{IORQ}$ ,  $\overline{RD}$ ,  $\overline{WR}$ ,  $\overline{RFSH}$ ,  $\overline{M1}$ ,  $\overline{HALT}$ ,  $\overline{BUSAK}$ ), reicht es aus, wenn die Leitungen in einer Richtung gepuffert werden. Für den bidirektionalen Datenbus ist es nun notwendig, daß die Daten in 2 Richtungen gesendet werden können. Die Treiberbausteine müssen also einmal Daten senden oder Daten empfangen können. Dieses Umschalten, in welche Richtung die Treiberbausteine die Daten puffern sollen, wird nun durch das Signal " $\overline{DBDR}$ " (Data Bus Drive) gesteuert. Während  $\overline{WR}$ -Zyklen ist  $\overline{DBDR}$  auf high. Anhand der Schaltung der Buffer-Karte ist ersichtlich, daß bei  $\overline{DBDR}$ =high der Treiberbaustein IC2 gesperrt ist, während die Treiber IC1 und IC8 Daten vom Nascom auf den Nasbus weiter-senden. Nur wenn man von einer der Speicher-karten oder einer I/O-Platine die CPU lesen will ( $\overline{RD}$ -Zyklus), und während des  $\overline{INTACK}$  Zyklus wird die  $\overline{DBDR}$ -Leitung auf "low" gesetzt. Dann sperren IC1 und 8. Der Treiber IC2 gibt die Daten vom Nasbus zum Nascom weiter. Normalerweise liegt " $\overline{DBDR}$ " auf logisch high. Das "low" Signal wird auf der Karte generiert, von der die CPU lesen will. Dazu gibt es auf jeder Karte eine Decodier-logik, die erkennt, ob die CPU von dieser Karte liest. Sie erzeugt, wenn  $\overline{MREQ}$ , bzw.  $\overline{IORQ}$  und  $\overline{RD}$  aktiv sind, durch ein IC mit Open-Collector Ausgang auf  $\overline{DBDR}$  "low". (Beim  $\overline{INACK}$  wird getestet, ob ein Pheripheriebau-stein auf der Karte den Interrupt ausgelöst hat und ob  $\overline{M1}$  und  $\overline{IORQ}$  aktiv sind). Da es möglich ist, daß sich RAM-Speicher mit dem 4K-Memory Block auf der Nascomkarte überla-gern, wird durch die Decodierlogik auf den Memorykarten auch das  $\overline{MEXT}$  Signal abgefragt, und bei Lesezugriff auf diesen Speicherblock wird das  $\overline{DBDR}$  nicht auf low geschaltet. Zum Schluß sei noch bemerkt, daß bei  $\overline{BUSAK}$  und  $\overline{RESET}$  aktiv die Datenleitungen in beide Richtungen (unabhängig von  $\overline{DBDR}$ ) gesperrt sind.

Der  $\overline{M1}$ -Zyklus (Instruction OP-Code Zyklus) ist der CPU-Zyklus, in dem sich die CPU aus dem Speicher den nächsten auszuführenden

Befehl (OP-Code) holt und anschließend noch einen Refresh-Zyklus durchführt. Der Ablauf ist in Fig.1 dargestellt.

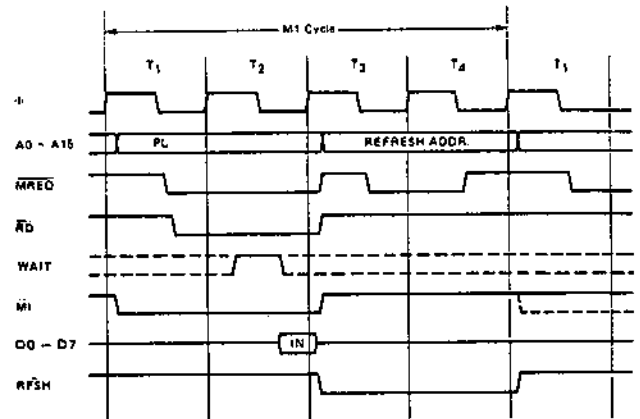


FIG. 1

Während Taktzyklus 1 (T1) legt die CPU den Wert des Programmzählers PC auf den Adreß-bus. (PC ist ein internes 16bit Register der CPU, welches auf die Speicherzelle zeigt, in welcher der nächste abzuarbeitende Befehl steht). Gleichzeitig wird der  $\overline{M1}$ -Ausgang der CPU aktiv (deshalb  $\overline{M1}$ -Zyklus). Anschließend werden  $\overline{RD}$  und  $\overline{MREQ}$  aktiv. Diese beiden Controlleitungen zeigen an, daß die CPU aus dem Speicher ein Byte lesen will. Am Anfang von T3 wird nun das Byte, also der OP-Code, vom Datenbus in die CPU übernommen. Das Byte, welches die CPU nun über den Datenbus erhalten hat, wird nun während der folgenden zwei Takt-Zyklen (T3 und T4) analysiert und intern ausgeführt. Während die CPU damit beschäftigt ist, intern den Befehl zu verar-beiten, kann extern der Refreshzyklus ablau-fen. Damit werden Daten in dynamischen Spei-cherzellen aufgefrischt, die sonst verloren gehen würden.



# GALAXIS

## von Jürgen Weiermann

In einem Feld von 14x14 Feldern sind vier Raumstationen versteckt. Die Aufgabe des Spielers besteht nun darin, diese Raumstationen zu finden. Damit bei der Suche der Erfolg nicht ganz und gar dem Kommissar "Zufall" überlassen sein soll, hilft der Computer ein wenig. Nach der Eingabe der Koordinaten vom Spieler errechnet er, wieviele Funksprüche er von diesem Punkt aus von den Raumstationen über die Senkrechte Horizontale und Diagonale empfangen kann. Den errechneten Wert zeigt er dann an dem vom Spieler eingegebenen Punkt an. (Übrigens: das rotierende kleine Quadrat im großen Quadrat soll den sich drehenden Radarspiegel darstellen.) Fehler bei der Eingabe können durch BS verbessert werden. Findet man nun eine der Raumstationen, so druckt der Computer an der betreffenden Stelle das Klingelzeichen als Symbol für die gefundene Station. Im oberen rechten Bildrand wird die Anzahl der Züge angezeigt; somit steht einem Wettkampf in der Familie oder unter Freunden nichts mehr im Wege. Wurden alle vier Stationen gefunden, so gratuliert der Computer dem Spieler in Form einer Laufschrift.

Sinn des Spiels ist es, das logische sowie kombinatorische Denken zu fördern.

Das Programm läuft unter NAS-SYS und wurde so konzipiert, daß es auch auf der Minimal-konfiguration des NASCOM läuft, damit jeder etwas davon hat.

Und hier noch einige Unterprogramme und deren Funktionen für den Teil der Leser, die Programme disassemblieren und auf ihre Funktion hin untersuchen.

1. CD 65 ØD : gibt den Text "Bitte mehrmals Taste "S" druecken." aus.
2. CD BE ØE : ruft den Zufallsgenerator auf.
3. CD ØB ØF : erzeugt Spielfeld.
4. CD 96 ØE : schreibt "Ihre Eingabe".
5. CD AC ØE : holt Eingabe
6. CD 68 ØE : berechnet Cursorposition im Spielfeld.
7. CD 68 ØF : erzeugt rotierendes Quadrat.
- CD 98 ØF : Time Delay
- CD A3 ØF : setzt Cursor um eine Stelle zurück

- CD A9 ØF : bringt A zur Anzeige, setzt Cursor zurück, ruft Time Delay auf
- CD 82 ØF : Trefferanzeige
8. CD B4 ØD : berechnet Treffer; A=ØF Volltreffer, sonst A=Trefferanzahl
9. CD DØ ØC : erzeugt Laufschrift.

```

0C80 3E 00 F5 CD 65 0D CD BE 89
0C88 0E CD 0B 0F DF 5D CD 96 28
0C90 0E CD AC 0E CF CD 68 0E F5 69
0C98 CD 68 0F F1 1E 00 CD B4 78
0CA0 0D 7B FE 0F 28 04 DF 7A C6
0CA8 18 06 21 64 0D 7E 3C 77 95
0CB0 F1 3C 27 21 38 08 22 29 BC
0CB8 0C F5 DF 68 F1 F5 21 64 77
0CC0 0D 7E FE 04 20 07 F1 CD 3E
0CC8 D0 0C C3 80 0C C3 91 0C 5F
0CD0 21 02 0D E5 1E 30 21 CA 2A
0CD8 0B 01 CB 0B 0A 77 23 03 6D
0CE0 1D 20 F9 CD 98 0F E1 7E F5
0CE8 FE 00 28 08 32 F9 0B 23 7B
0CF0 E5 C3 D4 0C CF FE 53 28 CC
0CF8 02 C7 00 21 64 0D 3E 00 9D
0D00 77 C9 2A 2A 2A 20 47 72 A4
0D08 61 74 75 6C 69 65 72 65 70
0D10 2C 53 69 65 20 68 61 62 B5
0D18 65 6E 20 61 6C 6C 65 20 D6
0D20 53 74 61 74 69 6F 6E 65 74
0D28 6E 20 67 65 66 75 6E 64 3C
0D30 65 6E 20 2A 2A 2A 20 53 21
0D38 3D 4E 65 75 65 73 20 53 F5
0D40 70 69 65 6C 20 20 20 20 77
0D48 20 20 20 20 20 20 20 20 55
0D50 20 20 20 20 20 20 20 20 5D
0D58 20 20 20 20 20 20 20 20 65
0D60 20 20 20 00 00 EF 0C 42 0A
0D68 69 74 74 65 20 6D 65 68 85
0D70 72 6D 61 6C 73 20 54 61 71
0D78 73 74 65 20 22 53 22 20 A8
0D80 64 72 75 65 63 6B 65 6E DE
0D88 21 00 C9 F5 F5 E6 0F CD 2B
0D90 4C 0E 28 15 F1 E6 F0 CB C6
0D98 3F CB 3F CB 3F CB 3F CD CF
0DA0 4C 0E 28 06 F1 CD 54 0E 55
0DA8 C9 F1 F1 E1 C5 01 07 00 0E
0DB0 09 C1 E5 C9 CD 54 0E F5 59
0DB8 7B FE 00 28 07 CD 82 0F CB
0DC0 F1 1E 0F C9 F1 F5 E6 0F 8F
0DC8 47 3E 0D 90 4F F1 F5 3C 68
0DD0 CD 8B 0D 00 00 00 00 00 42
0DD8 18 F5 F1 F5 3D CD 8B 0D 7A
0DE0 00 00 00 00 00 18 F5 F1 EB
0DE8 F5 D6 10 CD 54 0E 05 20 24
0DF0 F8 F1 F5 C6 10 CD 54 0E E0
0DF8 0D 20 F8 F1 C3 04 0E 18 08
0E00 00 00 00 00 06 02 C5 21 FC
0E08 10 0E 18 03 36 D6 48 F5 98
0E10 C6 0F F5 F5 E6 0F CD 4C EB
0E18 0E 28 16 F1 E6 F0 CB 3F 43
0E20 CB 3F CB 3F CB 3F CD 4C 65
0E28 0E 28 07 F1 CD 54 0E 18 AB
0E30 DF F1 F1 F1 05 20 D5 C1 AB
0E38 05 28 0B C5 36 C6 23 36 98
0E40 11 06 02 2B 18 C9 36 C6 6F
0E48 23 36 0F C9 FE 0E 28 02 BD
0E50 FE 0F C9 00 E5 C5 06 04 E8
0E58 21 00 0E F5 BE 20 01 1C 85
0E60 23 05 20 F8 F1 C1 E1 C9 0A
0E68 F5 F5 E6 0F 3C 47 F1 E6 AF
0E70 F0 CB 0F CB 0F CB 0F CB C7
0E78 0F F5 21 4A 08 22 29 0C 54
0E80 3E 12 F7 F7 05 20 F9 F1 DB
0E88 C6 00 28 07 47 3E 14 F7 1B
0E90 05 20 FA F1 C9 C9 21 2A 8B
0E98 09 22 29 0C EF 49 68 72 18
0EA0 65 20 45 69 6E 67 61 62 79
0EA8 65 3A 00 C9 21 6A 09 22 D4

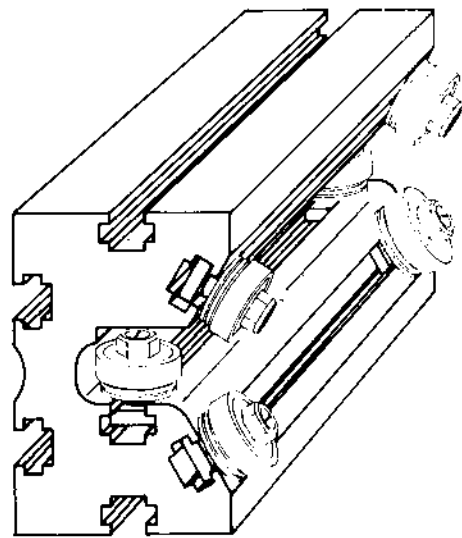
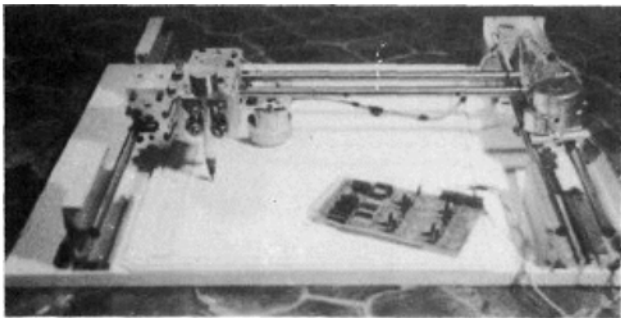
```



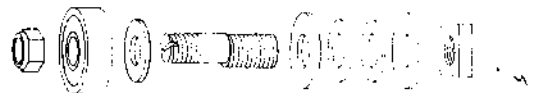
OEBO	29	OC	DF	63	11	6A	09	DF	98
OE88	64	21	21	OC	7E	C9	21	00	E0
OEEO	0E	06	04	3E	00	F5	DF	62	5A
OE88	FE	53	28	1D	F1	3C	F5	CB	59
OEEO	BF	CB	B7	CB	AF	CB	A7	EE	F9
OEEO	0E	20	03	F1	18	EF	D6	01	E6
OEEO	28	F9	F1	FE	DD	28	DC	18	F7
OEEO	DC	F1	F5	C5	E5	21	00	0E	91
OEEO	05	05	05	05	BE	28	06	04	02
OEEO	28	08	23	18	F7	E1	C1	F1	FB
OF00	18	C1	E1	C1	F1	77	23	05	1A
OF08	20	B9	C9	3E	OC	F7	21	1E	39
OF10	0F	01	D8	0B	7E	A7	28	16	75
OF18	02	23	03	18	F7	00	2A	2A	B2
OF20	2A	20	47	61	6C	61	78	69	CF
OF28	73	20	2A	2A	2A	00	21	0B	74
OF30	08	22	29	0C	06	0E	0E	00	CO
OF38	3E	20	F7	79	DF	7A	0C	05	7F
OF40	20	F6	06	0E	16	00	D5	3E	A2
OF48	0D	F7	D1	7A	DF	7A	14	D5	E8
OF50	CD	58	0F	05	20	F1	D1	C9	43
OF58	0E	0E	3E	20	F7	3E	7F	F7	8C
OF60	0D	20	F7	C9	37	F8	50	6E	49
OF68	0E	04	C5	3E	1C	F7	CD	98	04
OF70	0F	CD	A3	0F	3C	0D	20	F5	6B
OF78	C1	0D	20	01	C9	C5	0E	04	16
OF80	18	E9	0E	06	3E	2B	CD	A9	83
OF88	0F	3E	2A	CD	A9	0F	3E	58	29
OF90	CD	A9	0F	0D	20	EE	18	19	70
OF98	C5	0E	45	05	20	FD	0D	20	0E
OFA0	FA	C1	C9	F5	3E	11	F7	F1	5F
OFA8	C9	F7	CD	98	0F	CD	A3	0F	6A
OFB0	C9	3E	07	F7	C9	21	64	0D	1F
OFB8	3E	00	77	C3	80	0C	5F	72	9C

Zunächst wollte ich den Plotter in eigener Regie herstellen, habe aber bald gemerkt, daß eine Unmenge an feinmechanischem Werkzeug nötig wäre, bis mir der Katalog der Firma Isert-Electronic in die Hände fiel. (Bahnhofstr., 6419 Eiterfeld 1) Dort wurde genau das angeboten, was ich brauchte. Inzwischen wurden die Produkte der Firma vervollkommen, und man kann einen vollständigen Bohr- und Fräsautomaten für DM 1480,- bekommen, der sich natürlich auch als Plotter einsetzen läßt. Schon dieser Preis für eine fertige Anlage läßt sich kaum mit den Preisen für handelsübliche Plotter vergleichen. Eine Schablonenvorrichtung und Motoren für die Fräse und Bewegung in Z-Richtung brauche ich aber nicht, und so läßt sich der Preis wiederum reduzieren.

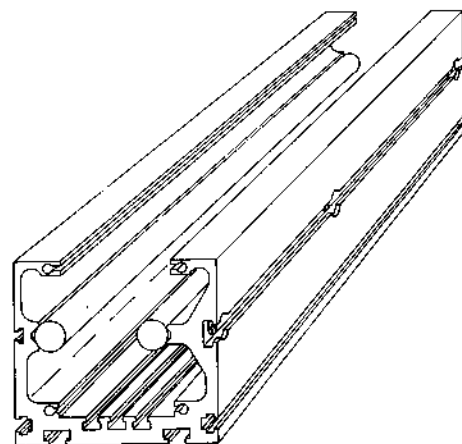
## SELBSTBAU-PLOTTER von Günter Böhm



„isert“-Linear-Lagerprofil IPL



Kugellager mit Zubehör

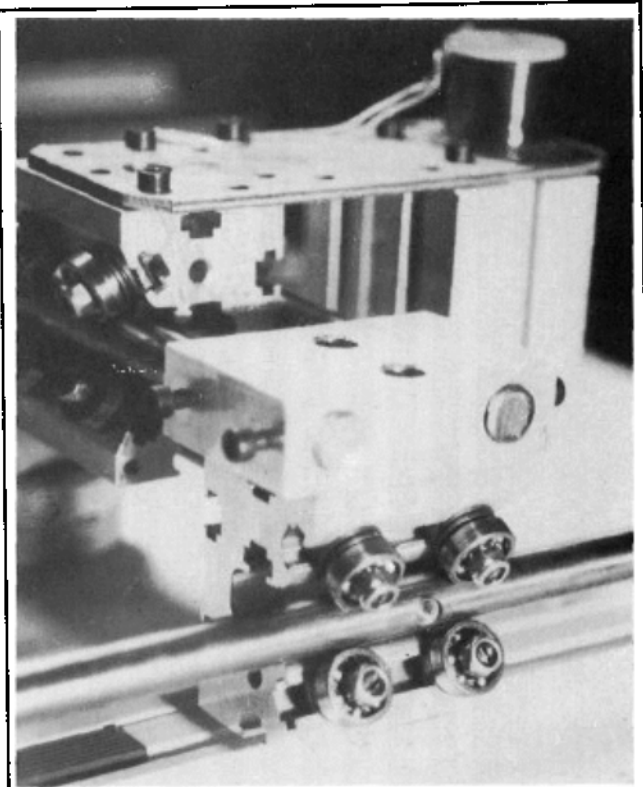


„isert“-Linear-Wellenprofil IPW mit Wellen

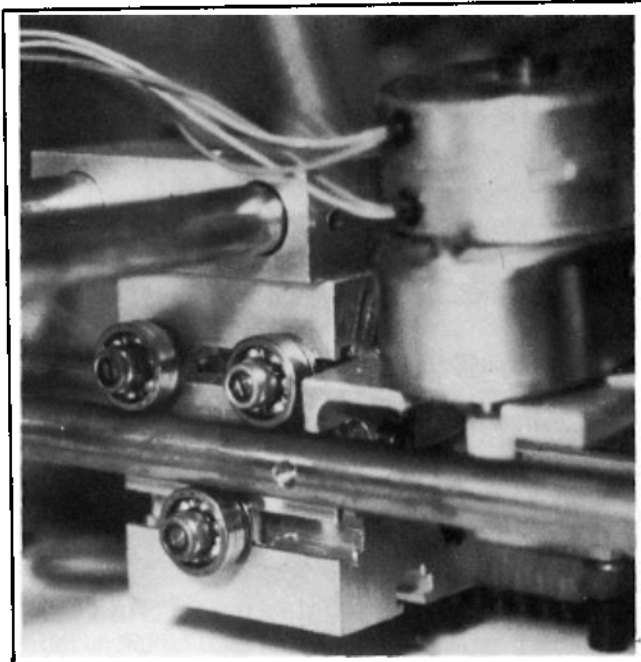
Ein Leser schrieb einmal, der Artikel von Herrn Bentz über Plotteransteuerung (Heft 3/80) habe ihm garnicht gefallen, denn das Material sei nur sehr schwer zu bekommen. Nun, das sind zwei Paar Stiefel. Der Artikel von Herrn Bentz war ausgezeichnet und animierte mich damals, endlich konkret an die Verwirklichung einer Idee zu gehen, die ich schon lange mit mir herumgetragen hatte: den Eigenbau eines Plotters. Was das Material für einen Selbstbau-Plotter angeht, habe ich auch lange gesucht, bis ich endlich eine akzeptable Lösung fand. (Gebrauchte Linien-schreiber gibt es eben nicht wie Sand am Meer).

Ich habe für meinen Plotter ca. DM 400,- für Wellen und Profile benötigt. Zusätzlich noch den Betrag für Motoren und Getriebe. Hier kann man eine Menge Geld sparen, wenn man die von Herrn Bentz vorgeschlagenen Fernschalteinheiten bestellt. (ca. DM 6,- bei der Fa. Völkner, Postfach 5320, 3300 Braunschweig). Sie haben nur einen Nachteil: Das Plotten dauert unheimlich lange. (1,5mm/sec bei einer Schrittlänge von 5 m) Als mir die Sache zu viel Zeit beanspruchte, bestellte ich bei VALVO (Unternehmensbereich Bauelemente, Postfach 106323, 2000 Hamburg) Motoren und Getriebe, die etwa 10 mal schneller sind. (Preis zusammen ca. DM 180,-). Der Zeitvorteil wurde allerdings mit einem Verlust an Genauigkeit erkaufte, denn die Anlage begann beim Richtungswechsel zu schwingen, was ich mechanisch nicht ganz beseitigen konnte. So habe ich bei bestimmten Zeichenvorgängen eine Pause einprogrammiert, die natürlich die Gesamtgeschwindigkeit bremst.

(Was Herr Bentz wohl schon geschafft hat. Siehe dieses Heft.)

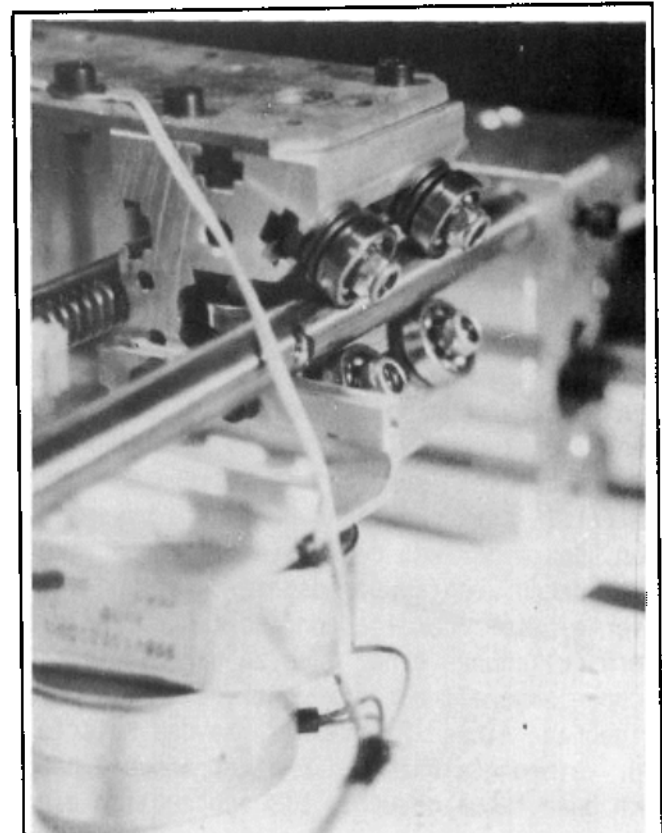


Die Stiftführung sitzt auf einem verzinkten Eisenblechprofil (normal als Balkenverbinder gebraucht)



Die Motorhalterung ist aus Abfällen des abgesägten Isel-Profils gemacht.

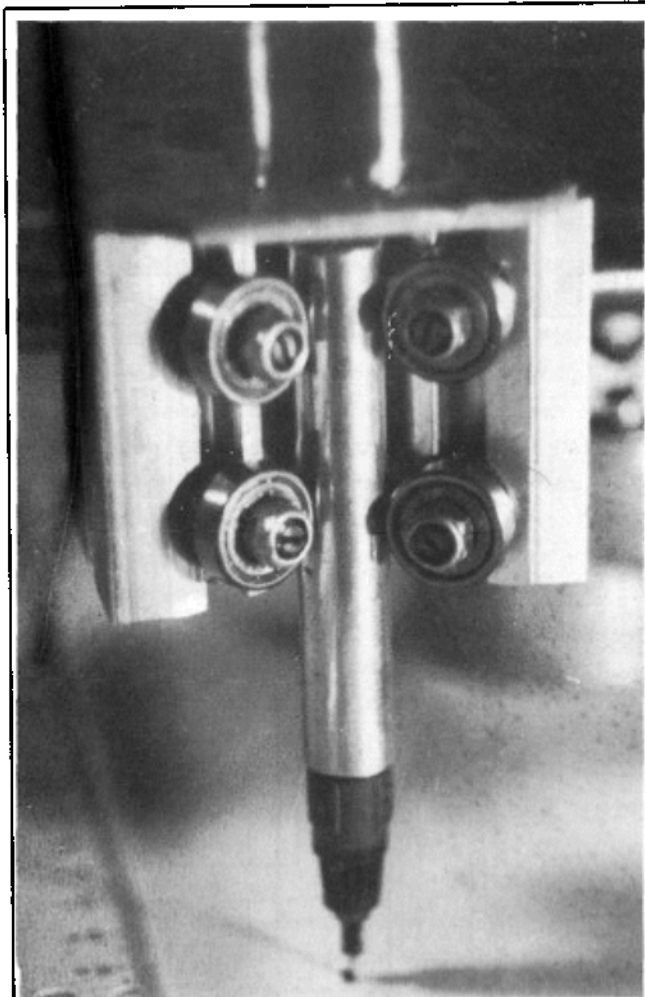
Alles in allem bin ich aber mit dem Ergebnis zufrieden. Inzwischen lasse ich Barcode plotten; Überschriften für unsere Schülerzeitung werden in Blockschrift geschrieben, und die Filmtitel für unsere S8-Filmgruppe sehen geplottet auch sehr ansprechend aus. Mit Computergrafik habe ich weniger im Sinn. Was mir noch vorschwebt, ist eine Hilfe beim Erstellen von Platinen-Layouts und der Einsatz beim Digitalisieren von Bildvorlagen



"Y-Träger" mit Zahnstange

Ich hoffe, die eingestreuten Bilder und Zeichnungen geben einen Eindruck von der Bauart des Plotters. Die einzigen Teile, die ich herstellen (drehen) lieb, waren zwei Alublöcke zur Führung der Y-Wellen. Selbstverständlich könnte man dafür auch Profile von Isert verwenden, aber hier wollte ich Geld und Maße in Z-Richtung sparen. (erstes war wohl wichtiger). Wer Beziehungen zu einer feinmechanischen Werkstatt hat, kann viele Teile vielleicht auch sehr viel billiger herstellen lassen; nur die Zahnstangen (für die Isert auch passende Stirnräder liefert) werden wohl problematisch.

Jedenfalls würde ich mich freuen, wenn einige Leser durch diesen Artikel dazu angeregt würden, sich in das Abenteuer Plotterbau zu stürzen. Für Auskünfte über meine Erfahrungen beim Bau stehe ich jederzeit gerne zur Verfügung, und auch Herr Bentz als alter Plotterhase (Er hat kürzlich einen Preis für Computergrafik erhalten!) gibt

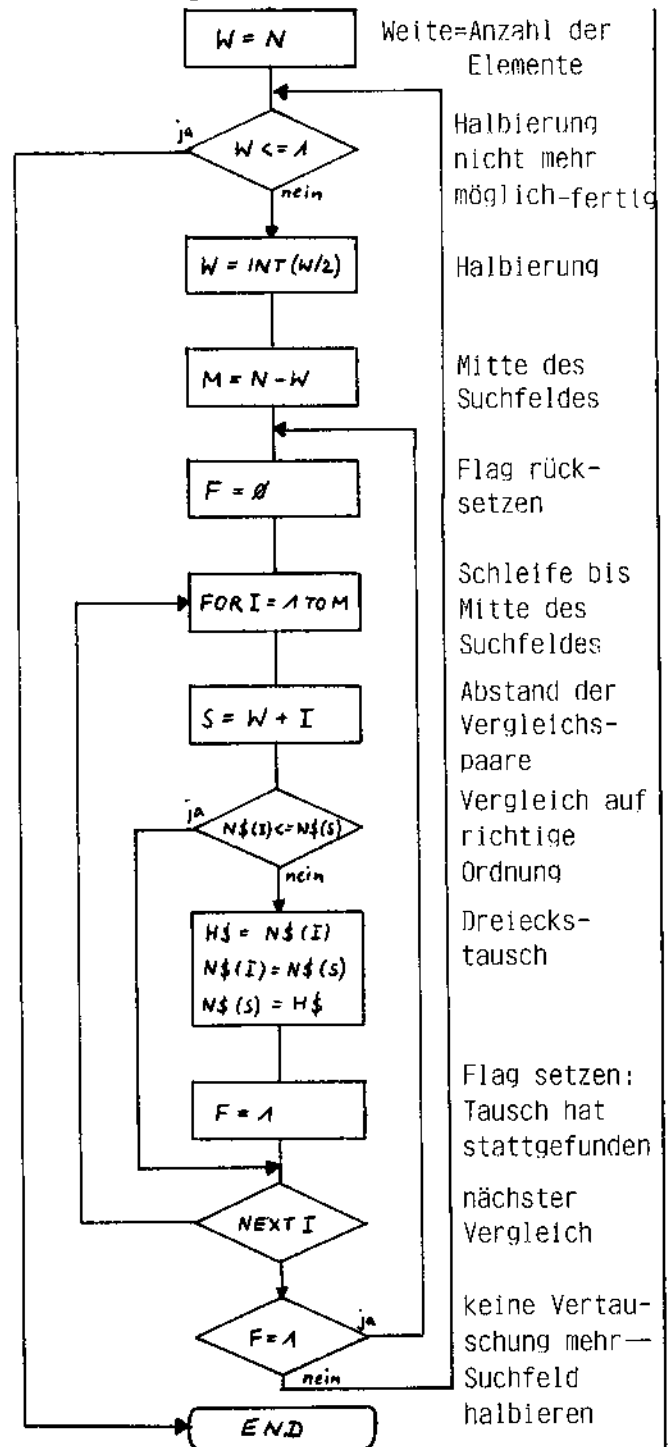


Der Tuschestift wird von einem Alu-Rohr gehalten. Eine eingeklebte Metallschraube läuft durch die Magnetspule.

Ihnen sicher gerne einen Tip. (Eine Platine zur Schrittmotoransteuerung, die PIO-Bus kompatibel ist, könnte übrigens auch veröffentlicht werden, ebenso eine Menge Software)

Lassen Sie sich nicht durch den Arbeitsaufwand abschrecken; es ist halb so schlimm und lohnt sich.

## SORTIEREN in BASIC Teil 5 v. Wolfgang Mayer-Gürr



Beim Bubble-Sortieren vergleicht der Rechner bei einem 10-Elemente-Feld im ersten Durchgang Inhalt Nr.1 mit den restlichen 9 Inhalten, im zweiten Durchgang Inhalt Nr.2 mit den restlichen 8 usw. Das ergibt  $9+8+7+\dots$ , also 45 Vergleiche. Bei einem normal gemischten Feld kommen dazu noch durchschnittlich 19 Tauschoperationen per Dreieckstausch.

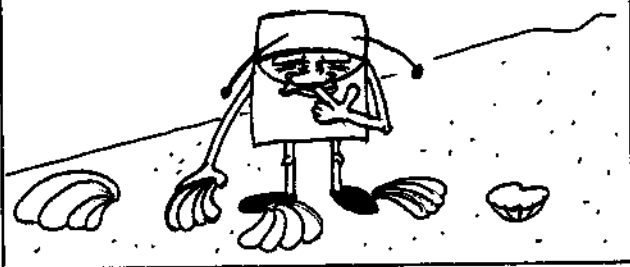
Beide Rechenvorgänge lassen sich beim Shell-Sortieren erheblich verringern, was sich natürlich besonders bei großen Feldern angenehm auf die Rechengeschwindigkeit auswirken wird.

Bei den vorausgegangenen Sortierverfahren wurden immer nur direkt benachbarte Elemente verglichen. Der Ablauf des Shell-Programms läßt sich aus dem Flußdiagramm erkennen. Die Verringerung der Vergleiche beruht auf folgendem Effekt: Wenn  $A < B$  und  $B < C$ , dann ist es nicht nötig, auch noch A und C zu vergleichen, denn A muß kleiner als C sein.

```

100 REM *****
110 REM * SHELL SORTIEREN *
120 REM *****
130 N = 10
140 REM * N = ANZAHL DER ELEMENTE
150 DIM N$(N)
160 FOR I = 1 TO N
170 PRINT "NR. "; I; TAB( 8);
180 INPUT N$(I)
190 NEXT I
200 GOSUB 260
210 REM * ZUM UNTERPROGRAMM SORTIEREN
220 FOR I = 1 TO N
230 PRINT N$(I)
240 NEXT I
250 END
260 REM * SORTIEREN
270 W = N
280 IF W <= 1 THEN 420
290 W = INT ( W / 2)
300 M = N - W
310 F = 0
320 FOR I = 1 TO M
330 S = W + I
340 IF N$(I) < N$(S) THEN 390
350 H# = N$(I)
360 N$(I) = N$(S)
370 N$(S) = H#
380 F = 1
390 NEXT I
400 IF F = 1 THEN 310
410 GOTO 280
420 RETURN

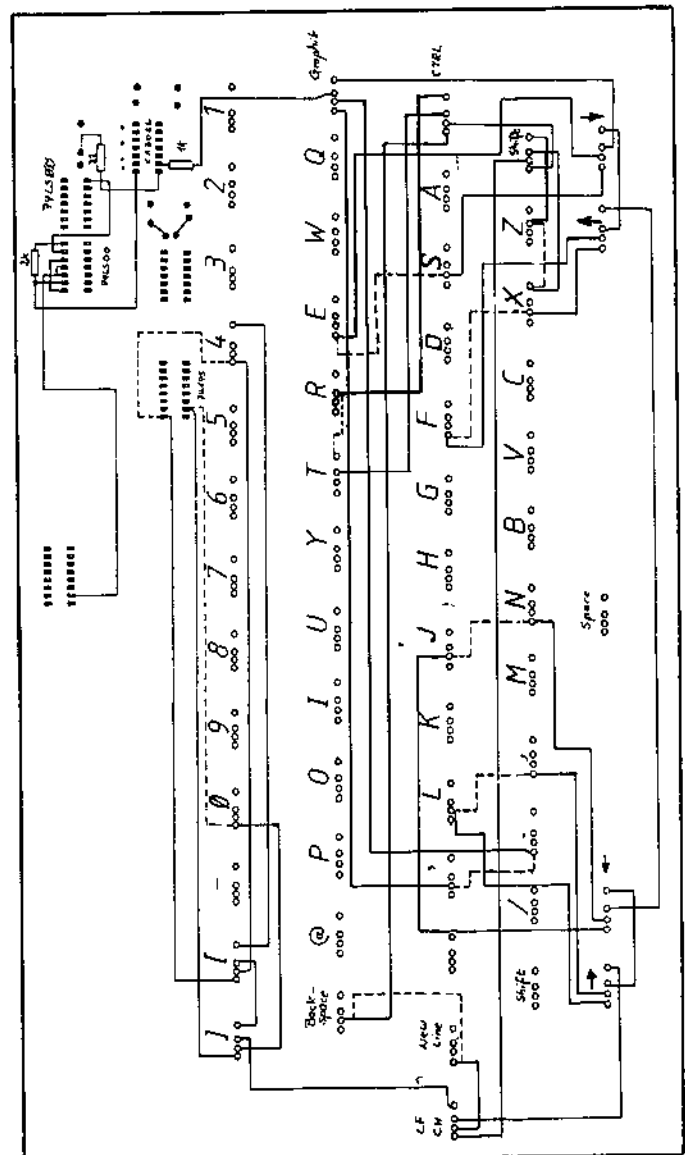
```



## TASTATURERWEITERUNG von Ralph Cramer

Das Tastaturerweiterungs-Kit (Upgrade-Kit) von N.A.S. München weist erhebliche Mängel auf, besonders im mitgelieferten Verdrahtungsplan. Ich habe das Kit gebaut und den Verdrahtungsplan verbessert. Die gestrichelt gezeichneten Verbindungen bestehen schon auf der Tastatur-Platine und sollten unterbrochen werden. Die durchgezogenen Linien stellen neue Verbindungen dar. Dazu kommen die im Plan eingezeichneten Widerstände. Ich hoffe, anderen NASCOM Journal Lesern damit geholfen zu haben.

(Siehe dazu auch den Artikel von Dieter Oberle, Heft 10/81. Red.)



# 16-KANAL A/D-Wandler von Peter Bentz

Das Herz dieser Schaltung ist der IC MK-50816 von Mostek (Preis : ca. 30 DM), der einen 8-Bit Analog/Digital-Wandler und einen 16-Kanal-Analogmultiplexer enthält. Die Analogeingänge sind mit je einem Operationsverstärker gepuffert, die Digitalausgänge und die Steuereingänge sind direkt mit dem NASCOM über 2 PIO-Ports verbunden.

Der Umwandlungsvorgang läuft nun so ab, daß zuerst über Bit 0 bis 3 des Ports A die Adresse des gewünschten Analogeinganges an den A/D-Wandler übergeben wird. Dann wird ALE (Adress Latch Enable) und START (Bit 5 und 6 von Port A) auf logisch High gesetzt. Dadurch wird die angelegte Adresse in den Decoder des Multiplexers übernommen und der entsprechende Kanal wird durchgeschaltet, so daß die zu messende Spannung am COMMON-Ausgang des Multiplexers und am COMPARATOR-Eingang des Wandlers anliegt.

Wenn nun START wieder auf Low gesetzt wird, läuft der eigentliche Wandlungsvorgang an und gleichzeitig geht der EOC-Ausgang (End of Conversion) auf Low, bis die Umwandlung abgeschlossen ist. In einer Abfrageschleife wird abgewartet, bis der EOC-Ausgang wieder auf High geht. Wenn nun der TRS-Eingang (Three-State) auf High gesetzt wird, werden die Datenausgänge vom hochohmigen in den aktiven Zustand gebracht, und die Daten können über den Port B der PIO in den NASCOM übernommen werden.

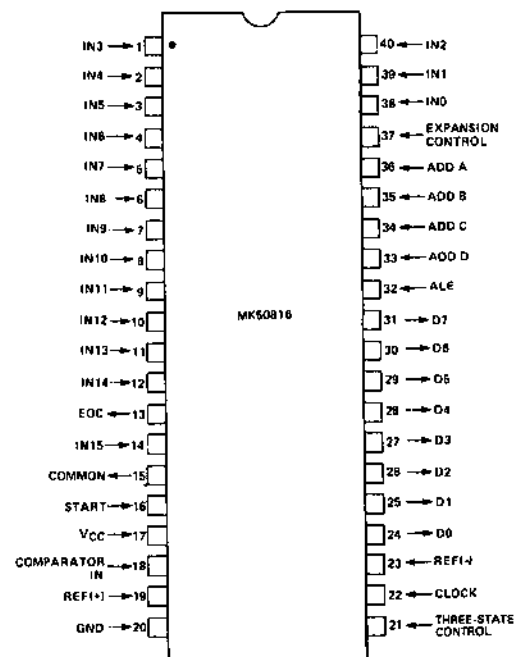
Man kann noch zwei Ausgänge des Ports A einsparen, wenn man TRS fest an +5 Volt legt, d.h. die Datenausgänge des A/D-Wandlers sind immer aktiv, und wenn man ALE und START miteinander verbindet und mit nur einem Ausgang von Port A gleichzeitig ansteuert.

Der A/D-Wandler verfügt über einen internen Taktgenerator, der in Gang gesetzt wird, wenn man den CLOCK-Eingang auf Gnd legt, was in der hier beschriebenen Schaltung der Einfachheit halber auch gemacht wurde. Laut Datenblatt kann man jedoch einen externen Taktgenerator von ca. 640 kHz an den CLOCK-Eingang anschließen und kann damit die eigentliche Umwandlungszeit von ca. 150  $\mu$ S auf 108  $\mu$ S verkürzen.

Zwischen die Ausgänge des A/D-Wandlers und

die PIO sind Widerstände eingefügt, um in dem Fall, daß die PIO-Eingänge versehentlich auf Datenausgabe gesetzt wurden, einen Kurzschluß zu verhindern.

An die Referenzspannungseingänge REF+ und REF- wird die höchste und die niederste zu messende Spannung angelegt, es sind jedoch nur Spannungen zwischen 0 und +5 Volt zulässig. Will man z.B. ohne die Eingangsverstärker messen, können die Referenzspannungseingänge direkt an Gnd und Vcc gelegt werden und dann kann man den gesamten Messbereich von 0 bis +5 Volt erfassen. Die vorgeschalteten Operationsverstärker können zwar noch sehr hochohmige Spannungsquellen erfassen, sind aber nicht über den gesamten Spannungsbereich linear. Deshalb kann man nur den linearen Übertragungsbereich von ca. 1 bis 4 Volt benützen und dazu kann man mit den beiden Potentiometern die Referenzspannungen einstellen.



Die Analoganschlüsse erfolgen über 16 Schraubklemmen, zusätzlich können noch Schraubklemmen für Gnd und Vcc eingelötet werden. Geätzte Platinen (100 x 160 mm) können gegen Unkostenerstattung gerne abgegeben werden.

## Unterprogramm zum Erfassen eines Analog-Wertes

Das Register A enthält vor dem Aufruf die Adresse des Kanales, nach dem Aufruf den di-

igitalisierten Analogwert.

Aktivieren der PIO :

```

OC90 F5 ADW PUSH AF
      3E 4F LD A, 4F
      D3 0B OUT (11),A
      3E FF LD A, FF
      D3 0A OUT (10),A
      3E 80 LD A, 80
      D3 0A OUT (10),A
      F1 POP AF
  
```

Adresse ausgeben :

```

E6 0F ADW1 AND OF
D3 08 OUT (8),A
F6 60 OR 60
  
```

Adresse in Decoder übernehmen :

```

D3 08 OUT (8),A
3E 10 LD A, 10
  
```

Start der Umwandlung,

Ausgänge aktivieren :

```

D3 08 OUT (8),A
  
```

EOC abfragen :

```

DB 08 IN A,(8)
E6 80 AND 80
28 FA JRZ
  
```

fertig, Daten übernehmen :

```

DB 09 IN A,(9)
C9 RET
  
```

Die Umwandlungszeit beträgt 0,27 msec und wenn man die PIO-Aktivierung, die ja nur einmal durchlaufen werden muß, wegläßt und das Programm bei ADW1 aufruft, kann man die Zeit auf 0,24 msec verkürzen. Da die Digitalisierung sukzessiv erfolgt, ist die Zeit nicht vom Analogwert abhängig und damit konstant.

Zur Ansteuerung des A/D-Wandlers in BASIC (8K-ROM-Version) benötigt man noch folgendes Unterprogramm :

```

OC80 CD 8B E9 CALL DEINT
      7B LD A,E
      CD 90 OC CALL ADW
      47 LD B,A
      3E 00 LD A,0
      CD F2 F0 CALL POF2
      C9 RET
  
```

Das BASIC-Programm zur Datenerfassung sieht so aus :

```

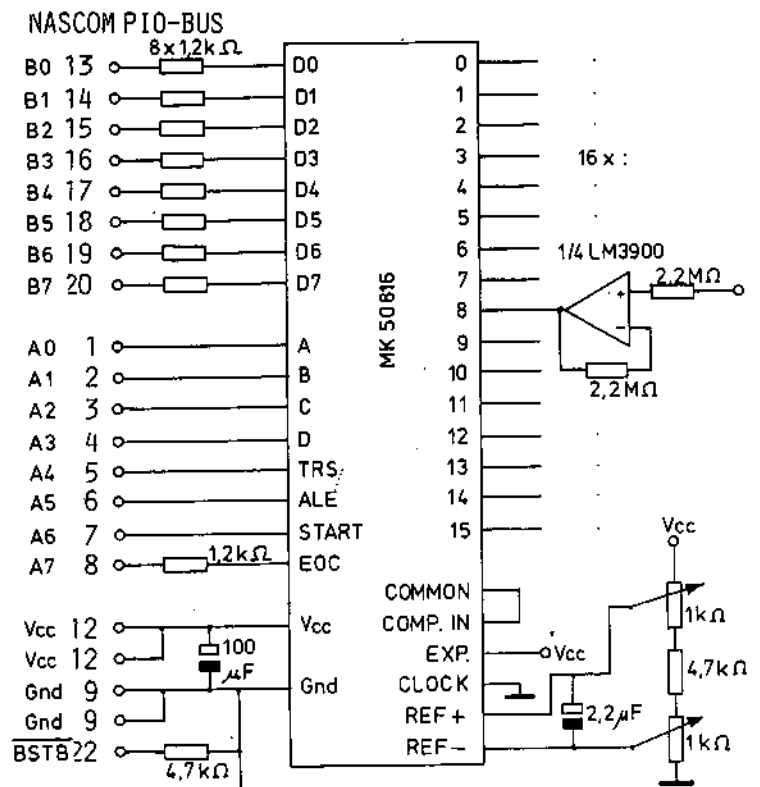
DOKE 4100,3200
X =USR(KN)
  
```

Im Laufe des Programmes muß einmal die Adresse des Benutzer-Unterprogrammes (OC80 = 3200) an das BASIC-Programm übergeben werden. Mit dem USR-Aufruf wird die A/D-Wandlung durchgeführt. Die Kanalnummer KN muß vor dem Aufruf einen Wert von 0 bis 15 haben und nach dem Aufruf enthält X den digitalisierten Wert der Analogspannung am Kanal KN. Die Umwandlung benötigt wegen des trägen BASIC-Interpreters 3 msec.

Ein Beispiel, wie man den A/D-Wandler an ein Z80-Bus-System anschließen kann, beschreibt R.D.Klein in seinem Buch Mikrocomputer Hard- und Softwarepraxis, München 1981.

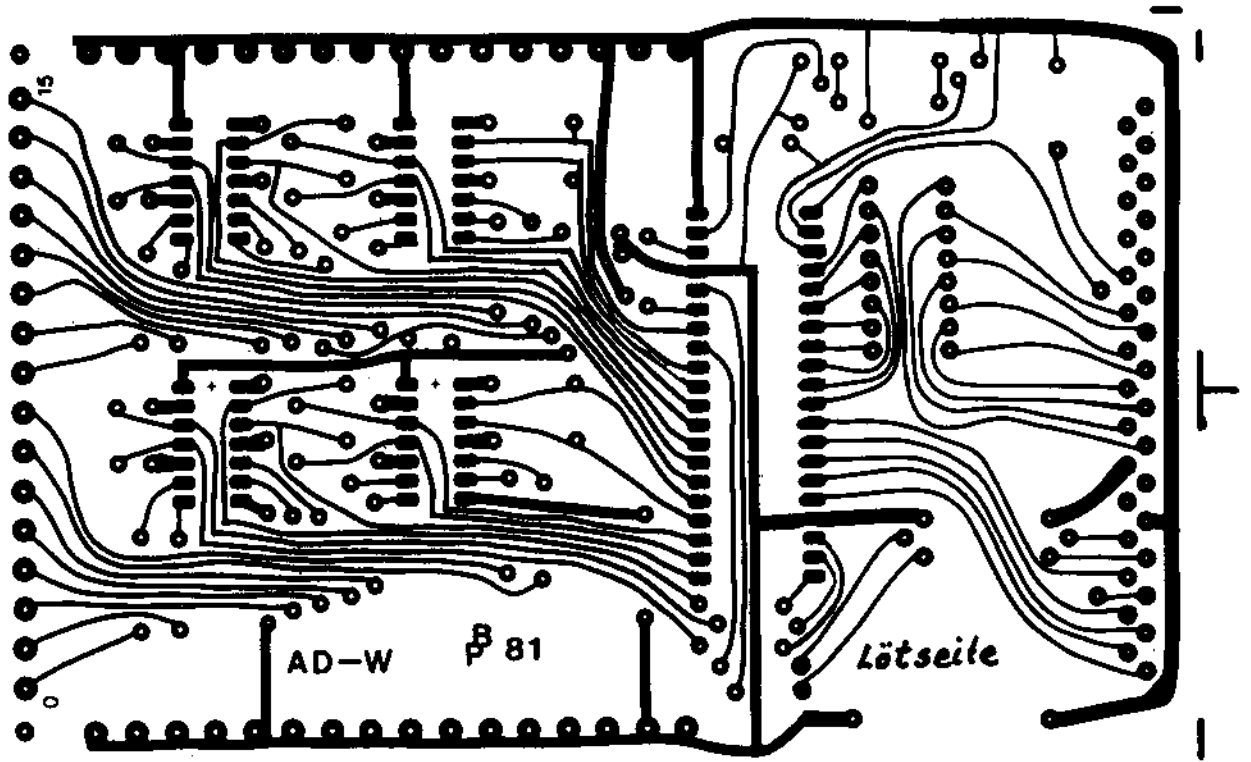
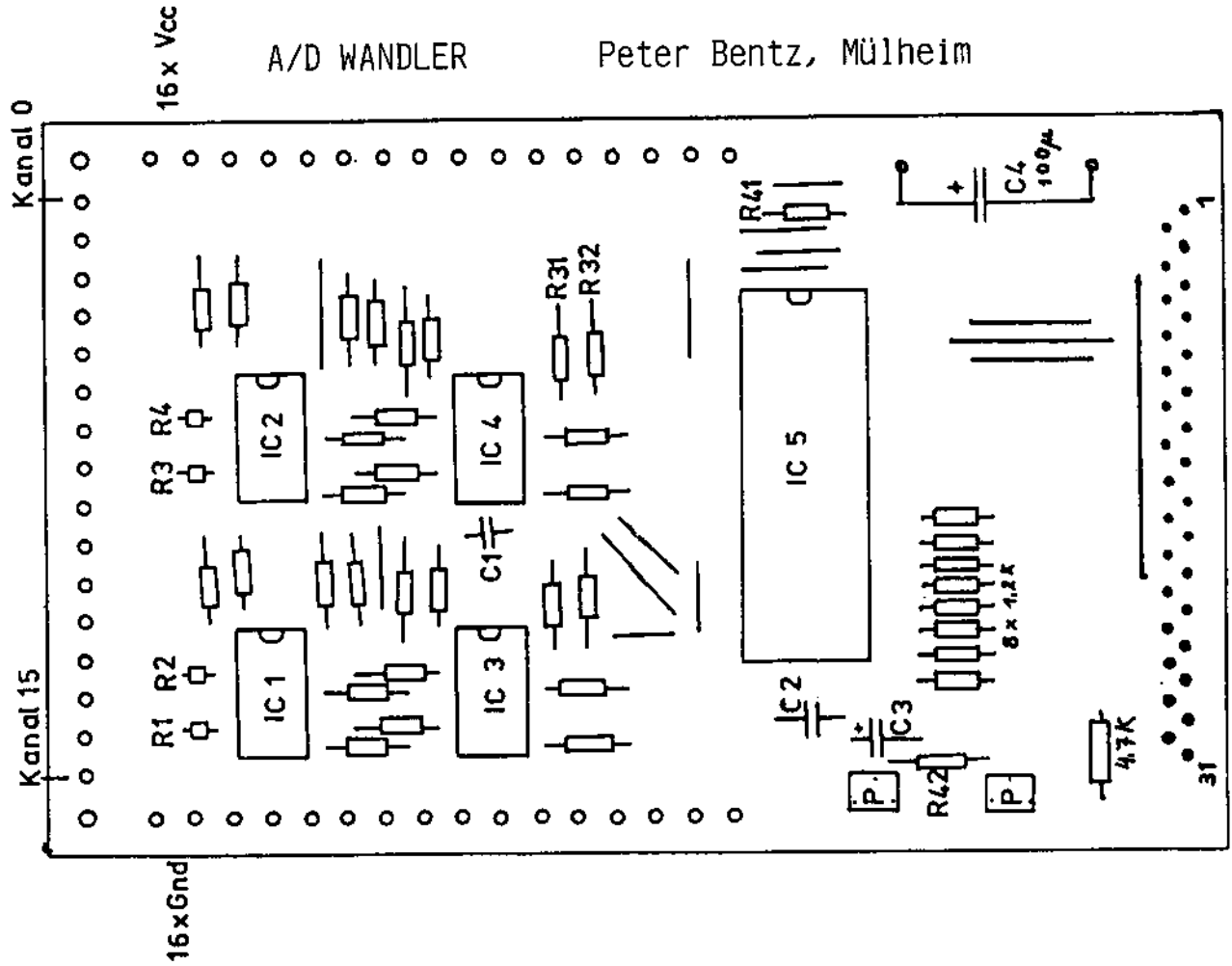
#### Bestückungsliste

IC1 - IC4 : 4x LM3900	R1 - R32 : 32x 2,2M $\Omega$
IC5 : MK 50816	R33 - R40 : 8x 1,2k $\Omega$
C1 : 100nF. Keramik	R41 : 1k $\Omega$
C2 u. C3 : 2x 2,2 $\mu$ F/ 16V Ta.	R42 u. R43 : 2x 4,7k $\Omega$
C4 : 100 $\mu$ F/ 20V Elko	48x Schraubklemmen
P : 2x 1k $\Omega$	oder Lötnägel
	Trimmer stehend 15x Drahtbrücken



# A/D WANDLER

Peter Bentz, Mülheim



# Monitor Umschalt Karte von Dieter Oberle

## Anwendungsmöglichkeiten:

Die Vielzahl der verfügbaren Monitorprogramme für NASCOM-Mikrocomputer von denen jedes seine Vor- und Nachteile hat und der Umstand, daß diese nicht vollständig kompatibel zueinander sind, hat mich veranlaßt, eine Adapterkarte zu entwickeln, die zumindest die Benutzung von zwei Monitorprogrammen oder auch den vollen RAM-Ausbau ab Adr. 0000H ohne ständiges Umstecken der EPROM's erlaubt.

Mit dieser Karte kann ohne zusätzliche Kabelverbindungen (bei Verwendung von 2708 EPROM's) einfach durch manuelles- oder durch Umschalten per Software eines von zwei Monitorprogrammen (z.B. NASSYS I oder NASBUG 4) ausgewählt werden. Ebenso ist es damit möglich, bei Bestückung mit 1K Statik-RAM's 4118 den gesamten Speicher des NASCOM I mit RAM's auszubauen.

Dies ist dann sinnvoll, wenn man andere Monitorprogramme oder komfortable Betriebssysteme, wie z.B. CP/M, installieren will, die zum Betrieb RAM-Speicherbereich ab Adr. 0000H benötigen. Diese können dann mit einem der NASCOM Monitorprogramme von Kassette oder auch von einer Floppy zunächst in den RAM-Speicher höherer Adr. geladen werden. Anschließend werden dann per Software die Monitor EPROM's abgeschaltet und gleichzeitig ab Adr. 0000H die RAM's eingeschaltet. In diese kann nun das vorher geladene Betriebssystem oder Monitorprogramm copiert und gestartet werden. Viele Betriebssysteme und Monitorprogramme anderer Z-80 Systeme brauchen gerade ab Adr. 0000H RAM-Speicher (z.B. CP/M).

Bei Verwendung von Floppydisks kann auf diese Weise jedes Betriebssystem schnell geladen und gestartet werden. (z.B. auch NASSYS I, NASSYS III oder NASBUG 4)

Dazu ist es lediglich notwendig, ein kleines Ladeprogramm zu schreiben und in den EPROM's ab Adr. 0000H-07FFH abzuspeichern. Am besten eignet sich dazu der alte 1KB Monitor NASBUG T2, weil dann die zweite Fassung frei ist für das Ladeprogramm.

Dies hat den Vorteil, daß bei entsprechend konfigurierter Systemdiskette auch gleichzeitig die gesamte benötigte Utility-Software wie ZEAP, NASPEN, BASIC oder auch eigene Softwareroutinen mitgeladen werden können, sodaß ganz auf EPROM's verzichtet werden kann. Mit einem einfachen Floppybetriebssystem, wie es bei mir bereits läuft, können dann auch beliebige andere Programme von Floppy geladen oder auf Floppy gespeichert werden.

Der Aufbau der beschriebenen Adapterkarte mit RAM's ist nur dann sinnvoll, wenn ein schnelles Speichermedium, wie Floppydisk oder schnelle Kassettenrecorder, zur Verfügung stehen, da sonst mit erheblichen Ladezeiten zu rechnen ist.

Auf meinem NASCOM I mache ich schon seit längerer Zeit von dieser Möglichkeit mit zwei Minifloppy Laufwerken Gebrauch und habe sehr gute Erfahrungen damit gemacht.

## Funktionsbeschreibung:

Über die Gatter in IC1 (74LS32) werden die CS Signale für die EPROM's oder die RAM's abhängig vom Pegel am TTL-Eingang "MON" (kann z.B. von einem freien Bit in Port 0 oder von der PIO angesteuert werden) oder von der Schalterstellung S1 an die jeweiligen ausgewählten Chips durchgeschaltet. Zusätzlich zeigen die beiden LED's den gerade selektierten Bereich an.

## Aufbauanleitung:

Vorweg: Layoutkopie und Bestückungsplan sind gegen Freiumschatz bei der Redaktion erhältlich.

Um möglichst ohne Kabelverbindungen auszukommen, wurden die IC's so platziert, daß der Abstand der original eingebauten EPROM-Fassungen auf der NASCOM Platine genau übernommen wurde. So ist es durch einfaches auflöten von zwei 24-poligen IC-Fassungssteckern auf der Unterseite der Adapterkarte möglich diese direkt in die Fassungen der Grundplatine zu stecken. Der Abstand der Karte ist dann ausreichend, sodaß die Adapterkarte über den anderen IC's Platz hat. (Bild 1)

Falls der Aufbau mit RAM's erfolgen soll,



# MONITOR-RAM-KARTE

für NASCOM I

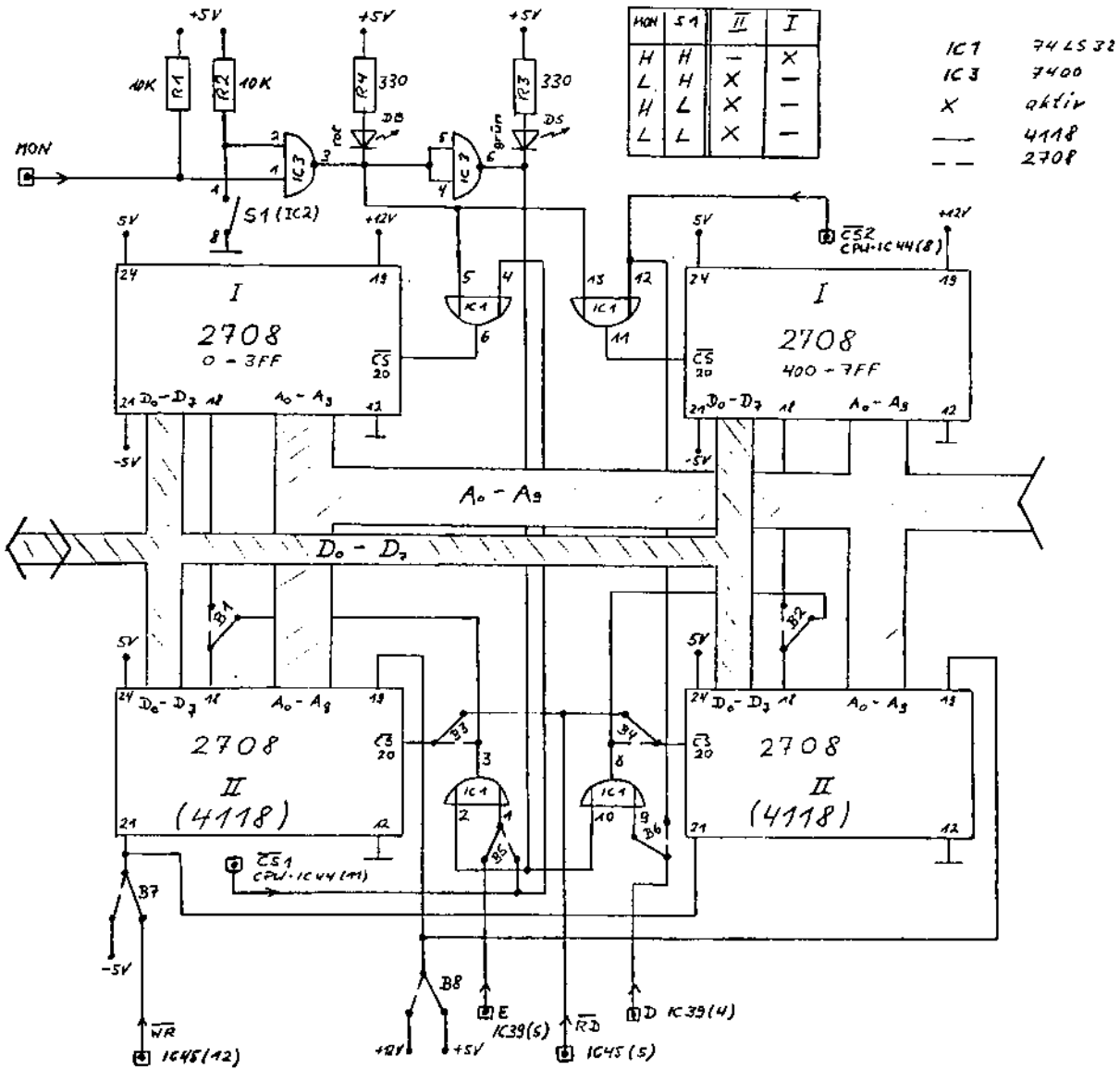


Bild 2

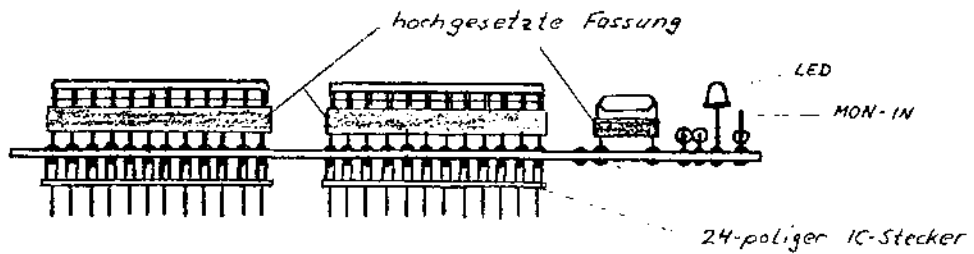


Bild 1

muß die Platine entsprechend dem Schaltplan (Bild 2) geändert werden. Außerdem sind dann noch 5 Drahtverbindungen von der CPU-Platine aus notwendig. Diese führt man am besten steckbar aus, damit die Karte ohne Löten auch leicht entfernt werden kann. Die IC-Pin's auf der NASCOM Platine zu denen die Drahtverbindungen hergestellt werden müssen sind im Schaltplan (Bild 2) bezeichnet.

Zur Durchkontaktierung müssen die IC-Fassungen möglichst hoch eingelötet werden damit auch auf der Bestückungsseite der Platine an den IC-Fassungen gelötet werden kann. (klappt mit den meisten Fassungsstypen) Eine andere Möglichkeit bietet die Verwendung von Band- oder Einzelkontakten, die ohne Schwierigkeiten von oben gelötet werden können.

Durchkontaktierungen, die nicht mit Bauteileanschlüssen zusammenfallen, müssen als erstes eingebaut werden und zwar am besten wie folgt:

- a) Eine Styroporplatte von 1-2 cm Dicke unter die gebohrte Platine legen. (Löcher maximal 0,6 mm)
- b) Abfalldrahtenden oder Silberdraht 0,5 mm durch die Löcher in das Styropor stecken bis sie fest sitzen.
- c) Überstehende Drahtenden sauber und knapp abzwicken und anschließend verlöten.
- d) Platine umdrehen, überstehende Drahtenden knapp abzwicken und ebenfalls sauber verlöten.

Werden beim Löten die Drähtchen wieder aus dem Loch gezogen so müssen dickere Drähte oder kleinere Löcher verwendet werden, damit sie fest sitzen.

Auf diese Weise habe ich schon mehrere tausend Durchkontaktierungen ohne Fehler bewerkstelligt.

Nach dem Durchkontaktieren werden die IC-Sockel in der oben beschriebenen Art eingelötet. Dort, wo auf der Bestückungsseite eine Leiterbahn an den Sockel führt, muß nun ebenfalls oben gelötet werden, um die Durchkontaktierung herzustellen. Auf sinnvolle Reihenfolge beim Einlöten der IC-Sockel muß geachtet werden, damit die Durchkontaktierungsstellen zum Löten auch noch zugänglich sind.

Danach werden alle übrigen Bauteile eingelötet und durchkontaktiert, wo es

erforderlich ist.

Zum Schluß werden die beiden 24-poligen IC-Stecker von unten (bis auf PIN 20 (CS)) auf die Sockel gelötet. Es ist unbedingt auf den richtigen Abstand zu achten. Pin 20 des Steckers wird nach außen gebogen und mit einem Draht an das dort vorgesehene Lötauge angelötet.

## Klein Buchstaben für T2 von Jürgen Weiermann

Wenn man sein Formatierprogramm nur unter T2 laufen lassen kann, wird man sehr bald feststellen müssen, daß man der Aufforderung der Redaktion, Groß- und Kleinbuchstaben zu verwenden, nicht Folge leisten kann. Aus diesem Grund habe ich das folgende Programm GKUS (Gross-Kleinschrift Umschaltung) geschrieben. Anstelle des Aufrufs CD 69 00 wird CD 90 0F gesetzt. Man kann nun durch Drücken der Taste "#" die Groß- bzw. Klein-schrift im FLIPFLOP Modus an- und ausschalten. Um diese Routine im Formatierprogramm benutzen zu können, müssen an den Stellen 0C83 und 0E5C die Unterprogrammaufrufe geändert und die Warteschleife 30 F B entfernt werden.

0F90	CD 69 00	Call KEYB
0F93	30 FB	Jr,nc FB
0F95	FE 23	CP,"#" (entspr.Brit.£)
0F97	28 0E	Jr,z 0E
0F99	18 00	Jr, 00 ODER 0
0F9B	C9	RET
0F9C	F5	Push A
0F9D	C6 BF	Add A,BF
0F9F	38 02	Jr,c 02
0FA1	F1	Pop A
0FA2	C9	RET
0FA3	F1	Pop A
0FA4	C6 20	Add A,20
0FA6	C9	RET
0CA7	3A BC 0F	LD A,(0FBC)
0FAA	F5	Push A
0FAB	3A 9A 0F	LD A,(0F9A)
0FAE	32 BC 0F	LD (0FBC),A
0FB1	F1	Pop A
0FB2	32 9A 0F	LD (0F9A),A
0FB5	CD 69 00	Call KEYB
0FB8	30 FB	Jr,nc FB
0FBA	18 DD	Jr DD
0FBC	01	

# MDCR-Monitor

## Teil 2 von J.C. Lotter

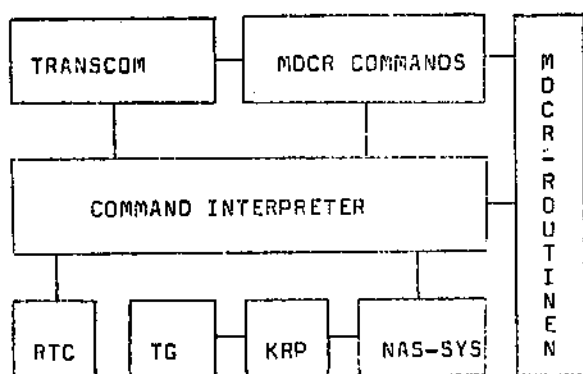
Das hier vorgestellte Programm ist das schon seit langem angekündigte MDCR-Steuerprogramm. Es dient primär dem Betrieb des MDCR-Laufwerks, ist jedoch darüber hinaus universell einsetzbar. Außer den MDCR-Routinen ist unter anderem enthalten:

- ein Keyboard-Repeater mit akustischer Tastendruck-Quittierung
- ein Interruptgesteuerter Tongenerator
- eine Interruptuhr mit Stundensignal und Stoppuhrfunktionen (1/1000 sec.)
- ein komfortabler Command-Interpreter, der auch von externen Programmen (z.B. Editor, Eprom-Programmer) benutzt werden kann.

UNICON braucht zum Laufen ein interruptfähiges NAS-SYS (z.B. NAS-SYS 1), einen Lautsprecheranschluß an Port 0 (siehe Schaltung) und das weiter unten beschriebene MDCR Interface (s.auch Nascom-Journal 3/81).

Die hier abgedruckte RAM-Version belegt 2k Speicherplatz von 4000h-4800h. Außerdem wird von den MDCR-Routinen ein Bereich von 20 Byte pro File unterhalb der Adresse 5000h überschrieben. Eine Eprom-Version mit beliebiger Startadresse können Sie von mir beziehen.

Doch nun zur eigentlichen Beschreibung. Unicon ist folgendermaßen aufgebaut:



TRANSCOM - Transient Command Area  
 RTC - Real Time Clock  
 TG - Tone Generator  
 KRP - Keyboard Repeater

Nach dem Starten auf Adresse 4000h springt das Programm in die Ausführungsschleife des Command-Interpreters und meldet sich mit ":" Nun können Sie einen Befehl eingeben. Der Command-Interpreter führt 4 Arten von Befeh-

len aus:

- Alle NAS-SYS-Befehle. Hierzu muß vor den Befehl ein Schrägstrich "/" eingetippt werden. "/N" schaltet z.B. den Keyboard-Repeater ab.

- Interpreter-interne Befehle. Es gibt folgende internen Befehle:

N kehrt zu NAS-SYS zurück. Uhr und Keyboard-Repeater laufen dabei weiter.

P macht einen Cold-Start. Die Uhr wird ab-, der Repeater eingeschaltet. Die MDCR-Befehle werden in die Command Area geladen.

T startet die Uhr, die in der Kopfzeile mitläuft. T HHMM stellt die Uhr (HH=Stunden, MM=Minuten).

S stoppt die Uhr im Stoppuhrbetrieb.

? listet die Befehle auf, die augenblicklich in der Command Area stehen.

- File-Befehle. Der Interpreter startet ein Programm, das als File auf der Cassette steht. Als Befehlsword gilt der Name des Files. Vor den Namen muß ein Doppelpunkt ":" eingetippt werden.

- Befehle der Command Area. Das ist eine Befehlstabelle, deren Startadresse in der Speicherstelle 0BFCh steht. Normalerweise ist dies die MDCR-Befehlsliste.

Zur Ausführung der MDCR-Befehle muß nicht das volle Wort eingetippt werden; es genügen die Anfangsbuchstaben. Filenamen können mit "\*" abgekürzt werden. Z.B. kann man statt "VERIFY TESTPROGRAMM" einfach "V TE\*" eintippen. Wird nun auf "New Line" bzw. "Enter" gedrückt, schreibt der Interpreter den Befehl aus ("VERIFY TE\*") und bezieht ihn auf das erste File, dessen Namen mit "TE" beginnt.

Nach erneutem "Enter" erfolgt die Ausführung. Namen werden ab 12 Buchstaben abgeschnitten. Nun zu den einzelnen Befehlen:

- FORMAT löscht und formatiert die Cassette. Dieser Vorgang dauert ca. 3 Minuten; jede Cassette muß vor dem erstmaligen Beschreiben formatiert werden. Während des Formatierens wird die Bandlänge gemessen und daraus berechnet, wieviele Bytes auf die Cassette passen.

- SAVE nnn AAAA BBBB SSSS speichert das File "nnn" auf Cassette. AAAA ist die Anfangs-, BBBB die End- und SSSS die Startadresse. Nach dem Abspeichern wird angezeigt, wieviele kBytes jetzt noch auf die Cassette passen

("ROOM: xx kByte"). Das Abspeichern dauert, je nach Länge des Files, ca. 20-200 sec.

- UPDATE nnn überschreibt das File "nnn". Damit können Files nachträglich geändert werden.
- LOAD nnn lädt das File "nnn" in den Speicher.

- VERIFY nnn vergleicht File "nnn" mit dem Speicher. Differenzen werden angezeigt.
- DIRECTORY listet alle Files nach Namen und Länge auf. Der Vorgang dauert ca. 10 sec.

Die Fehlermeldungen erfolgen aus Platzgründen kodiert und bedeuten im Klartext folgendes:

- ERROR 00: Das File paßt nicht mehr auf die Cassette.
- ERROR 05: Die Cassette ist nicht formatiert.
- ERROR 06: Auf der Cassette wurde ein Teil der Daten durch starke Magnetfelder vernichtet.
- ERROR 07: Es ist keine Cassette eingelegt.
- ERROR 08: Die Cassette ist schreibgeschützt.
- ERROR 20: Die Adressen sind falsch oder unvollständig.
- ERROR 30: Das File ist nicht auf der Cassette.
- ERROR 31: Ein File dieses Namens ist bereits auf der Cassette.
- ERROR 99: Unbekannter Befehl.

Ein Lesefehler (der in der Praxis allerdings nicht vorkommen sollte) wird folgendermaßen angezeigt:

\*BB AAAA x ? (Lesefehler bei Sector BB, Adresse AAAA, falsches Byte x)

Das Programm zeigt während des Schreibens Lesens laufend den Sector, die Adresse und das gerade übertragene Byte an.

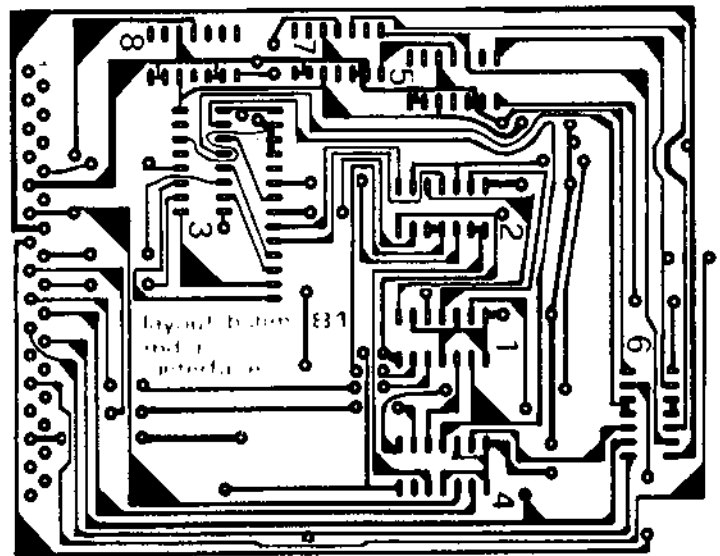
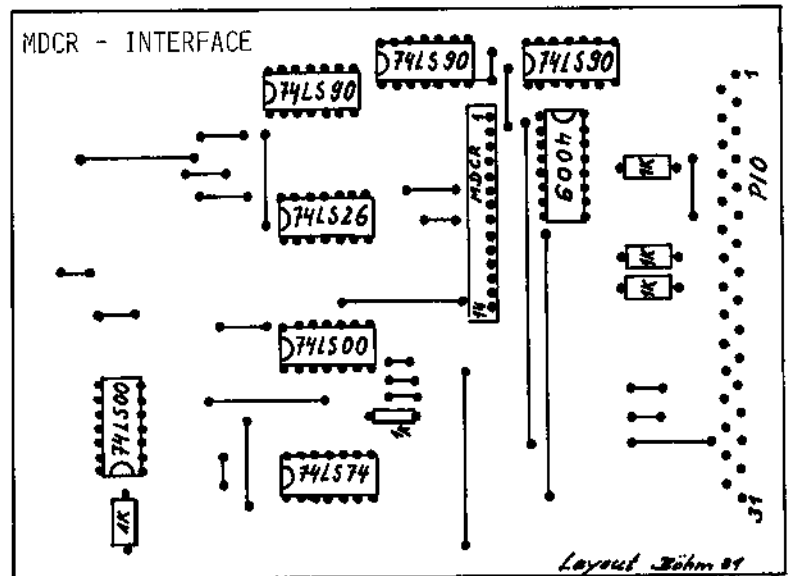
Alle Befehle sind als Unterprogramme ausgeführt und können von anderen Programmen aus aufgerufen werden. So kann man z.B. in BASIC-Programmen dieUSR-Anweisung zum Steuern des MDCR verwenden.

Eine verbesserte Version des schon in Heft 3/81 beschriebenen Interfaces ist hier nochmals abgedruckt. Die Steuerung erfolgt über den PIO-Bus (siehe Heft 9/81). Im Gegensatz zur früheren Version liegt das WEN-Signal

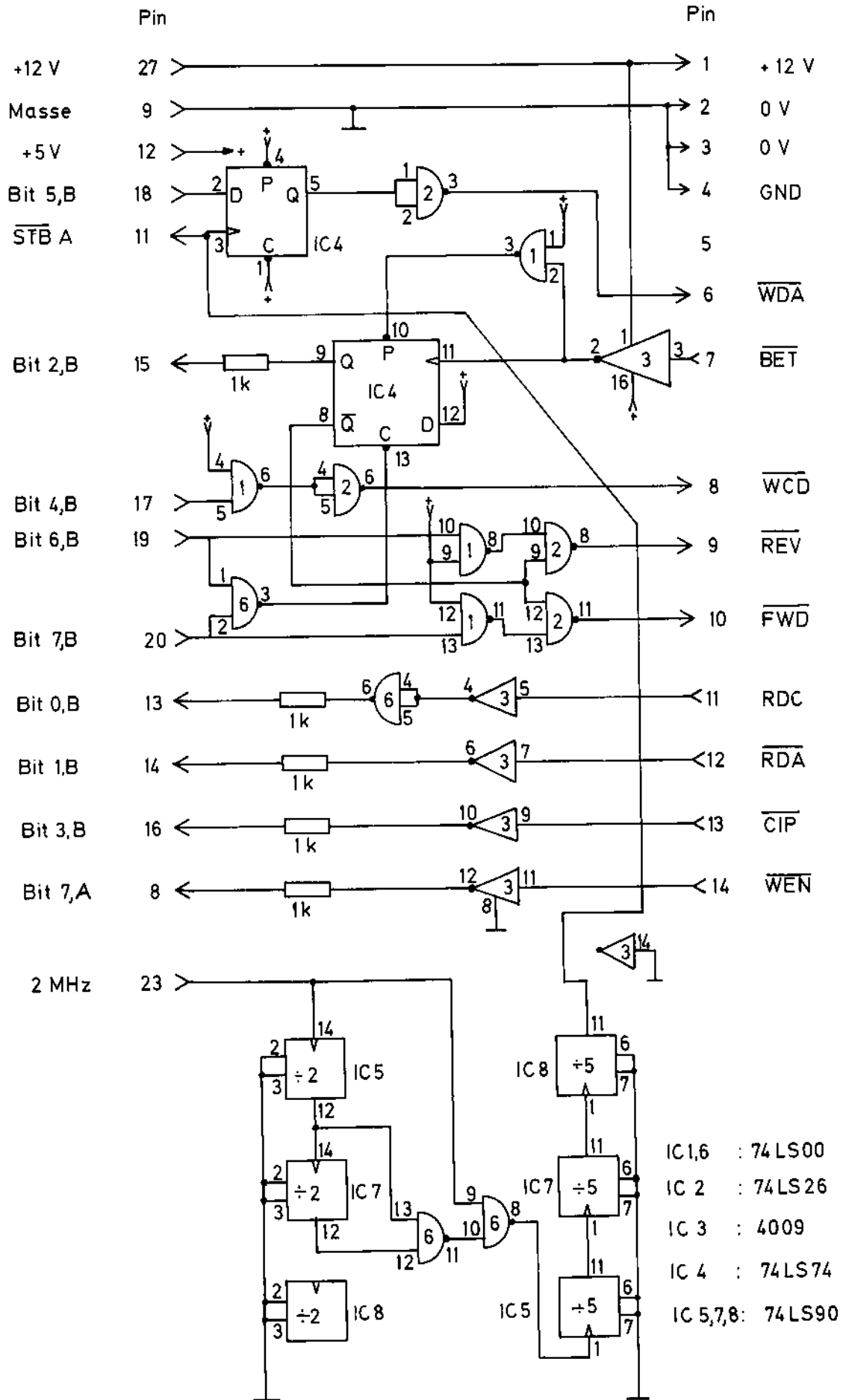
jetzt an Port A der PIO; davon abgesehen sind beide Interfaces softwarekompatibel.

Wie schon erwähnt, können Interessenten eine Eprom-Version von UNICON von mir erhalten. Es gibt 2 Ausführungen:

- Standardversion in einem 2716(5V) mit Startadresse A000h.
  - Spezialversion mit individuellen Änderungen, z.B. in zwei 2708'ern oder mit anderen Start- und Portadressen.
- Mitgeliefert wird das Assemblerlisting und eine ausführlichere Beschreibung. Die Standardversion kostet 65 Mark, die Spezialversion 95 Mark. Wer verständlicherweise keine Katze im Sack kaufen möchte, sollte das Programm anhand des hier abgedruckten Listings ruhig erstmal testen.



Vom obigen Layout können wir als Leserservice Atzfolien liefern. Red.



UNICON 1 - 4

Adr.	Data	Sum
4000	C3 08 44 E1 22 FC 08 CD 08 CD	29
4008	BE 41 2R FC 08 CD 09 40	40
4010	22 1E 0C F1 40 31 61 61	2E
4018	0C 2H 1E 0C 23 23 23 EF 10	10
4020	1B 0D 0D 0D 13 13 13 3A 15	15
4028	00 CD F8 41 0C 64 00 DF 8D	8D
4030	63 1R FE 3A 20 D1 13 1A 43	43
4038	4F 13 FE 3A DA 93 47 FE B4	B4
4040	2F 1A 32 2B 0C 20 16 FE 66	66
4048	41 38 39 FE 58 30 35 32 2H	2H
4050	0A 0C 13 DF 79 38 2D DF 55	55
4058	60 DF 5C 18 B9 FE 20 20 DF 42	42
4060	0F E5 21 B3 40 7E B9 28 07	07
4068	34 23 23 23 B7 20 F6 E1 47	47
4070	7B E6 C0 C5 08 5F 1A 47 62	62
4078	ED A1 2B 08 AF ED B1 23 E9	E9
4080	23 BE 20 EC D7 43 99 13 73	73
4088	1R FE 20 20 EB AF BE 28 80	80
4090	06 D7 4F ED 80 18 F5 D0 64	64
4098	F8 41 07 96 00 91 07 40 F6	F6
40A0	C5 23 7E 23 66 5F E9 3E 65	65
40A8	24 1E 0C BE 20 FA B1 C9 DF F3	F3
40B0	64 1E 0C BE 20 FA B1 C9 DF F3	F3
40B8	40 20 16 40 00 CD E8 56	56
40C0	EB 40 20 16 40 00 CD E8 56	56
40C8	52 EF 1B 2A 45 52 52 4F B7	B7
40D0	20 00 E1 7E DF 68 CD C8	C8
40D8	F8 41 1A 5A 08 2A FC 0B AF F9	F9
40E0	07 40 ED 53 29 0C EF 16 E1	E1
40F0	0C 00 C9 1B 2A FC 0B AF F9	F9
40F8	43 4F 4D 4D 41 4E 44 53 8A	8A
4100	00 D7 21 DF 69 ED 58 29 F2	F2
4108	0C AF 23 23 23 BE 28 11 64	64
4110	0E 0C D7 95 79 83 5F 56 18	18
4118	3F FE 38 38 EC DF 6A 18 53	53
4120	E2 EF 16 00 ED 58 29 0C C7	C7
4128	EB 23 36 2D 23 7D E6 3F 9F	9F
4130	EB 11 EF 08 06 02 7E C3 18	18
4138	FE 20 28 FA 38 12 28 ED 23	23
4140	0A ED 21 66 41 0E 05 ED 20 F6	F6
4148	ED 21 66 41 0E 05 ED 20 F6	F6
4150	3E FE CD 01 44 05 CD 80 88	88
4158	D6 3E 78 08 FB C9 30 30 49	49
4160	2E 30 30 30 30 C2 F3 42 73	73
4168	3E 78 CD F3 42 E5 F5 21 64	64
4170	F9 08 CD AC 41 CD 41 33 97	97
4178	2D CD AE 41 3E 34 CD AE	AE
4180	41 3E 34 CD AE 41 3E 34 CD AE	AE
4188	41 2D D7 20 3E 34 D7 1E 95	95
4190	CD F8 41 0D 5A C5 41 09 40	40
4198	C8 00 2D 2D 7E 2C FE 32 D5	D5
41A0	28 02 3E 38 D7 08 3E 31 0F	0F
41A8	D7 04 1B 07 3E 38 BE 30 1C	1C
41B0	05 34 F1 F1 E1 C9 36 30 11	11
41B8	2D C9 D7 02 DF 5B 21 EE 13	13
41C0	0B AF 36 2D BE 20 F9 13 08	08
41C8	23 EB C9 E3 C5 AF 4E 23 E5	E5
41D8	DD CB 2C 46 C0 F3 ED 5E 31	31
41E0	D7 00 F5 3E 47 ED 47 3E EC	EC
41F0	03 D3 06 D3 07 F1 ED 4D 12	12
41F8	E3 C5 F5 DD CB 2C 46 CC 8C	8C
4200	FF 43 7E 23 B7 28 23 4E 75	75
4208	23 47 D6 08 38 01 47 C5 90	90
4210	FE C0 30 ED 47 3A 00 0C C2	C2
4218	18 E7 CD D8 41 F1 C1 E3 E4	E4
4220	EE 20 32 00 0C D3 00 78 F9	F9
4228	C9 13 1A FE 20 28 FA ED 95	95
4230	53 9C 0C 06 0C 13 05 28 37	37
4240	0A 1A FE 20 28 FA ED 95 01	01
4248	21 0E 0C 78 B7 28 18 DF 1B	1B
4250	64 38 19 7E B7 28 15 23 E4	E4
4258	7E FD 77 00 23 F0 23 7E 55	55
4260	40 1A FE 20 28 FA ED 95 01	01
4268	78 FE 0D CD C9 40 20 F5 DE	DE
4270	3E 4F CD EA 43 E3 C5 F5 0E	0E
4280	4E 23 CD FF 43 06 3C FB 7F	7F
4288	1F 10 FC 00 20 F7 CD D0 1A	1A
4290	41 F1 C1 E3 C9 DF 61 D0 8E	8E
4298	21 00 0C 30 08 DD 36 2D 82	82
4300	B4 CD F8 41 09 3C 00 C9 RA	RA
4308	CD F8 41 C4 01 00 D0 35 C7	C7
4310	16 FF 7D FE 06 20 02 16 C8	C8
4318	BF FE 09 20 02 16 C7 7E 45	45
4320	72 23 10 E4 79 B7 C8 D0 75	75
4328	36 2D 1E CD F8 41 0A 14 8F	8F
4330	03 05 30 0B CB 15 9F 83 31	31
4338	EE 60 D9 08 FB ED 4D 0B 41	41
4340	37 10 F7 6C 06 08 0C 18 16	16
4348	F1 CD 87 43 CD 78 42 64 B6	B6
4350	01 00 08 1E 20 21 55 E1	E1
4358	D5 3E 4F 37 0B 3E FA CD 79	79
4360	01 44 CD 30 43 CD 2D 43 1D	1D
4368	3E 18 1B CD 2D 43 F3 D9 26	26
4370	41 09 40 3E 18 1B CD 2D 43	43
4378	43 3E 18 1B CD 2D 43 F3 D9	43
4380	43 3E 18 1B CD 2D 43 F3 D9	43
4388	43 3E 18 1B CD 2D 43 F3 D9	43
4390	43 3E 18 1B CD 2D 43 F3 D9	43
4398	43 3E 18 1B CD 2D 43 F3 D9	43
4400	43 3E 18 1B CD 2D 43 F3 D9	43
4408	43 3E 18 1B CD 2D 43 F3 D9	43
4410	43 3E 18 1B CD 2D 43 F3 D9	43
4418	43 3E 18 1B CD 2D 43 F3 D9	43
4420	43 3E 18 1B CD 2D 43 F3 D9	43
4428	43 3E 18 1B CD 2D 43 F3 D9	43
4430	43 3E 18 1B CD 2D 43 F3 D9	43
4438	43 3E 18 1B CD 2D 43 F3 D9	43
4440	43 3E 18 1B CD 2D 43 F3 D9	43
4448	43 3E 18 1B CD 2D 43 F3 D9	43
4450	43 3E 18 1B CD 2D 43 F3 D9	43
4458	43 3E 18 1B CD 2D 43 F3 D9	43
4460	43 3E 18 1B CD 2D 43 F3 D9	43
4468	43 3E 18 1B CD 2D 43 F3 D9	43
4470	43 3E 18 1B CD 2D 43 F3 D9	43
4478	43 3E 18 1B CD 2D 43 F3 D9	43
4480	43 3E 18 1B CD 2D 43 F3 D9	43
4488	43 3E 18 1B CD 2D 43 F3 D9	43
4490	43 3E 18 1B CD 2D 43 F3 D9	43
4498	43 3E 18 1B CD 2D 43 F3 D9	43
4500	43 3E 18 1B CD 2D 43 F3 D9	43

# MICROSOFT für CLD von Günter Endert

Das Microsoft ROM Basic speichert die Programme in einer platzsparenden, aber leider nicht für andere Betriebssysteme lesbaren Form auf der Cassette ab. Um die Programme dennoch übertragen zu können, kann man wie folgt vorgehen:

- \* Das Programm wird unter dem ROM-Basic mit CLOAD geladen
- \* LINES 1000 eingeben
- \* WIDTH 255 eingeben
- \* Reset Taste betätigen
- \* X30 eingeben
- \* Z eingeben
- \* Cassettenrecorder auf Aufnahme schalten und starten
- \* LIST eingeben und warten, bis das Basic sich mit OK meldet
- \* Recorder weiterlaufen lassen und die Reset Taste betätigen
- \* X30 eingeben
- \* CNTRL-D drücken
- \* Recorder ausschalten

Das Programm befindet sich nun in der List Form auf dem Band.

Das Zeichen CNTRL-D = ASCII (4) = EOT = End of Text teilt später dem Device-Driver mit, daß das Programm beendet ist; es läßt sich vom Basic aus nicht erzeugen. Der Vorgang kann nun für andere Programme wiederholt werden.

Einlesen des Programms unter CLDDOS:

- \* System booten. Die Diskette muß den Device-Driver AT: der alten CLDDOS-Version enthalten. AT: der neuen Version ist mit dem alten nicht identisch und arbeitet nur einseitig (Write).
- \* AT: auf PORT 1 (bzw. den Port, an dem der Recorder angeschlossen ist) setzen.
- \* Mit COPY FILENAME = AT: das Programm auf die Diskette lesen. Den Recorder erst einschalten, wenn die LED im Deckel des Laufwerks nicht mehr leuchtet! Er kann abgeschaltet werden, wenn das Zeichen EOT gelesen wurde und die LED wieder brennt.
- \* Das Programm in den Editor laden und dort am Anfang und Ende alle Zeilen und Leerzellen löschen. Es dürfen keine Zeilen mehr vorhanden sein, die nicht mit einer Zeilennummer anfangen!
- \* Das Programm wieder auf der Diskette

abspeichern.

\* Vom CLD-Basic aus das Programm mit OLD laden. Alle Zeilen, die ungültige Befehle enthalten, tragen vor diesem den Vermerk \*ERROR\* und lösen beim Listen ein Bell-Zeichen aus. Sie müssen entsprechend geändert und an das neue Basic angepaßt werden. (Mit dem Disk-MBasic müßte es im Prinzip genauso gehen; ich habe es noch nicht ausprobiert).

\* Das Programm nach einem Probelauf mit Replace wieder auf der Diskette abspeichern.

Dieses Verfahren lohnt natürlich nur bei Programmen, die nicht zuviele spezielle ROM Basic-Befehle enthalten. Bei Programmen, die viel mit den Befehlen SET, RESET und der Funktion POINT arbeiten, ist es normalerweise einfacher, sie neu zu schreiben.

Man kann übrigens auch Daten vom DOS aus auf dem Band abspeichern und später wieder einlesen (z.B. als Backup); man muß nur ein CNTRL-D anschließen (z.B. durch COPY AT: = FNAME.EXT, CNTRL-D.DAT, wobei CNTRL-D.DAT durch Basic erzeugt wurde: OPEN "CNTRL.DAT" FOR WRITE AS FILE #1:PRINT #1, CHR\$(4):CLOSE #1).

## Bildschirm auf Cassette von Karl Trust

DAS NACHFOLGENDE PROGRAMM DIENST EBENFALLS WIE DAS IM NJ 8/81 VERÖFFENTLICHTE, AUF DESSEN GRUNDLAGE ES AUCH AUFGEBAUT WURDE, ZUM SCHREIBEN VOM BILDSCHIRM AUF CASSETTE. JEDOCH IST ES AUF NAS-SYS LAUFFÄHIG, UND ES BIETET EINIGE KLEINERE VORTEILE.

1. MAN KANN DEN BILDSCHIRM VOLLSCHREIBEN BIS ZUM ANFANG DER ZEILE 15, DANN ERSCHEINT IM KOPF DER TEXT "BITTE ERST ABSPEICHERN". DAS ABSPEICHERN ERFOLGT DURCH DRUECKEN DER BEIDEN TASTEN SHIFT/CONTROL, DANN ERSCHEINT IN DER ZEILE 15 EIN CONTROL-ZEICHEN UND ES WIRD ABGESPEICHERT, ERKENNTLICH AN DER LEUCHTENDEN "DRIVE-LED",

2. IST DAS ABSPEICHERN BEENDET, ERSCHEINT DER TEXT "NEUE SEITE J/N?". BEI JA ERFOLGT DER SPRUNG ZUM ANFANG DES PROGRAMMS, BEI NEIN RUECKSPRUNG ZUM MONITOR.

3. AUCH TEXTE MIT "SPACES" AM ANFANG KOENNEN JETZT ABGESPEICHERT WERDEN. LEIDER HAT DIESES PROGRAMM AUCH EINEN KLEINEN NACHTEIL, DIE INHALTE DER "MARGINS" WERDEN MIT ABGE-

SPEICHERT, WAS ABER KAUM INS GEWICHT  
FAELLT, DA SICH DIES NUR DURCH EINE KLEINE  
VERZOEGERUNG AM ZEILENANFANG BEMERKBAR  
MACHT.

ES BLEIBT NOCH ANZUMERKEN, DASS BEIM ER-  
SCHEINEN DES TEXTES ZUM ABSPEICHERN KEIN  
ANDERES ZEICHEN MEHR ALS "CONTROL" ANGENOM-  
MEN WIRD.

DER EINSPRUNG IN DAS PROGRAMM IST BEI C80.

```

OC80 EF 0C 00 DF 7B F7 FE 40 16
OC88 2D 37 21 0A 08 DF 5F 3E 9A
OC90 A0 FF 3E 0C DF 6F 7E FE 4F
OC98 40 28 03 23 18 F6 DF 5F 7E
OCA0 EF 0D 4E 45 55 45 20 53 48
OCA8 45 49 54 45 3F 20 20 20 7A
OCB0 4A 2F 4E 0D 00 DF 7B FE E8
OCB8 4A CA 80 0C FE 4E 20 F5 C5
OCC0 C7 E5 C5 01 8A 0B ED 42 02
OCC8 28 05 C1 E1 C3 83 0C 21 16
OCDO EA 0C D5 11 D0 0B 01 17 AB
OCD8 00 ED B0 DF 7B FE 40 20 39
OCEO FA F5 3E 1B F7 F1 F7 C3 D6
OCE8 8A 0C 42 49 54 54 45 20 22
OCFO 45 52 53 54 20 41 42 53 30
OCF8 50 45 49 43 48 45 52 4E 52
OD00 21 00 00 00 00 00 00 00 2E

```

Anmerkung der Redaktion:

Die abgespeicherten Nullen der Margins wer-  
den dann störend, wenn der abgespeicherte  
Text mit einem Textverarbeitungsprogramm  
weiter bearbeitet werden soll. Dann muß man  
die Nullen mühsam wieder löschen. Der Nach-  
teil kann aber im Programm sehr einfach  
ausgebügelt werden:

Nach dem INC HL in 0C9B müßte AND A ; JR Z  
0C96 eingetragen werden. Um Platz für diese  
drei zusätzlichen Bytes zu schaffen (A7 28  
F8) könnte man die drei Spaces im Text  
(0CAD-CAF) streichen und das Programm ab  
0C9E um drei Adressen nach unten verschie-  
ben. 0C9A müßte dann noch 3 auf 06 verändert  
werden.

## CONMOD von Jürgen Weiermann

Die Idee zum Programm CONMOD kam mir, nach-  
dem ich das Programm Reversi, dessen HEX  
Listing mit Prüfsumme abgedruckt wurde, mit  
dem L-Befehl eingegeben hatte. Was mich an  
diesem Eingabe Modus störte, waren folgende  
Punkte:

1. Die Adressen mußten mit eingetippt werden.

2. Es wurde kein Cursor dargestellt.
3. Fehler konnten nicht verbessert werden.
4. Wenn eine Zeile richtig eingegeben war, so wurde sie anschließend gelöscht.

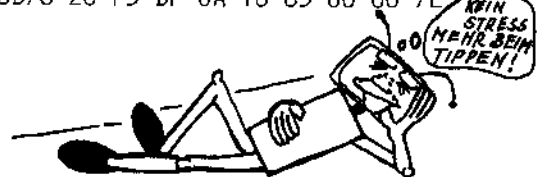
Diese vier Punkte sorgten dafür, daß die  
Eingabe zur Strapaze wurde. War man nur  
einen Moment unkonzentriert, so wußte man  
nicht mehr, ob man die Spacetaste gedrückt  
hatte, oder man hatte die Zeilennummer ver-  
gessen. All dies führte zu Fehlern, die man  
nur durch Eingabe einer neuen Zeile verbessern  
konnte.

All diese Mängel sollte das neue Programm  
ändern. Es wird laut meinem Listing bei der  
Adresse 0D00 gestartet, ist aber voll relo-  
katibel. Nach der Eingabe des ARG 1 wird für  
ARG 2 die Startadresse des einzugebenden  
Programms eingetippt. Das Programm führt  
Space und Newline automatisch aus, sowie die  
Adressierung der einzelnen Zeilen. Man kann  
nun durch CL und CR die eingegebene Zeile  
bei Bedarf korrigieren. Nach der Eingabe der  
Prüfsumme wird die Zeile auf Fehler hin  
untersucht. Wird ein Fehler festgestellt, so  
gibt der Computer ein "E" aus und ruft die  
Inline Routine auf. Dadurch wird ermöglicht,  
daß alle Screen Editing Befehle genutzt  
werden können. Bei der Eingabe der Zeile  
wurden die Edit-Befehle wie BS und SPACE  
nicht als solche verwendet. Sollte man ein-  
mal aus Versehen eben genannte Zeichen ver-  
wendet haben, so kann man Space durch CL  
rückgängig machen; bei BS wird es schon  
etwas schwieriger; am besten, man merkt sich  
die Zeilennummer und drückt die Reset Taste.  
Zeilen werden nur übernommen bei richtiger  
Prüfsumme.

```

OD00 DF 60 D5 E1 E5 DF 66 06 32
OD08 09 DF 7B FE 11 28 17 FE C4
OD10 12 28 17 F7 DF 7B FE 11 CE
OD18 28 17 FE 12 28 0F F7 DF 81
OD20 69 05 20 E5 18 0E F7 04 C1
OD28 3E 11 F7 18 E7 F7 05 3E B4
OD30 12 F7 18 D5 2A 29 0C DF 71
OD38 7C E5 D1 DF 64 2A 21 0C 11
OD40 3E 00 06 09 85 84 F5 DF 77
OD48 64 2A 21 0C F1 05 28 03 31
OD50 85 18 F3 BD 28 0A EF 45 10
OD58 00 DF 63 3E 13 F7 18 D4 DB
OD60 2A 29 0C DF 7C E5 D1 13 F0
OD68 13 13 13 06 08 DF 64 2A 29
OD70 21 0C 7D E1 77 23 E5 05 8C
OD78 20 F3 DF 6A 18 85 00 00 7E

```





# LIFE von Karl Trust

DAS NACHFOLGENDE LISTING BEZIEHT SICH AUF DAS SPIEL "LIFE", WELCHES BEREITS IM NASCOM JOURNAL VERÖFFENTLICHT WURDE. (Heft 1/80). ICH HABE ES ZUNÄCHST AUF NAS-SYS UMGESCHRIEBEN UND WEITERHIN LEICHT MODIFIZIERT, D.H. ANSTELLE VON "SPACES" BEIM SETZEN DER ZELLEN, MUSS MAN JETZT DIE PUNKTTASTE DRUECKEN, SODASS MAN JEDERZEIT SEHEN KANN, AUF WELCHEM PLATZ DES BILDSCHIRMS MAN SICH BEFINDET. NACH DER ERSTEN GENERATION VERSCHWINDEN DANN ALLE STEUERBEFEHLE UND DIE PUNKTE.

```

OC80 DF 62 30 FC 4F 21 D0 0E 47
OC88 7E B7 28 F4 23 B9 28 04 ED
OC90 23 23 18 F4 5E 23 56 21 E6
OC98 80 0C E5 EB E9 F7 F7 C9 A0
OCA0 00 00 00 00 F7 3E 3E 18 37
OCA8 F5 00 00 F7 00 00 DF 69 E8
OCB0 00 C9 CD BE 0C 28 EE 18 4A
OCB8 F9 CD BE 0C 18 E7 21 0B 7F
OCC0 08 11 60 0D 06 0F CD EB 1F
OCC8 0C 10 FB 06 00 FF 00 00 F0
OCD0 DF 62 00 FE 48 C8 10 F5 30
OCD8 3E 0C F7 00 00 21 0B 08 59
OCE0 11 60 0D 06 0F CD 44 0D 9D
OCE8 10 FB C9 C5 06 17 D5 11 90
OCF0 3E 00 D9 0E 00 D9 CD 2C F3
OCF8 0D 13 13 CD 2C 0D 13 13 63
OD00 CD 2C 0D 11 02 00 CD 2C 1F
OD08 0D D9 79 D9 FE 03 28 09 7F
OD10 FE 02 20 09 7E FE 4F 20 31
OD18 04 3E 4F 18 02 3E 20 D1 FF
OD20 12 23 23 13 10 C8 01 12 83
OD28 00 09 C1 C9 B7 ED 52 CD 8B
OD30 3B 0D 19 19 CD 3B 0D B7 83
OD38 ED 52 C9 7E FE 4F 20 03 3B
OD40 D9 0C D9 C9 C5 06 17 1A D0
OD48 77 23 23 13 10 F9 01 12 41
OD50 00 09 C1 C9 94 BA 40 00 7E
OD58 46 5D 1B 20 80 00 40 00 03

```

OD60 Beginn des Zwischenfeldes

```

OEE0 CD A5 0C C3 80 0C 52 9D 8A
OEE8 AC 00 98 4E 50 74 92 06 C4
OED0 4F AB 0C 08 9D 0C 0D A4 46
OED8 0C 1E A4 0C 53 B9 0C 45 1D
OEE0 B2 0C 2E 9D 0C EF 50 41 03

```

Mit folgendem kleinen Zusatz kann die Pausendauer vor dem Setzen der Zellen eingegeben werden. Bei kurzer Dauer (z. B. 02) ist die fließende Veränderung der Strukturen besonders reizvoll. Start bei

```

OEE5 EF 50 41 55 53 45 4E 44 F2
OEE8 41 55 45 52 3A 20 0D 00 8F
OEF5 DF 63 DF 64 3A 21 0C 32 21
OEF8 CC 0C EF 0C 00 C3 C0 0E 6F

```

# KANSAS-CITY v. Michael Bach

Kansas-City Interface & Baudratengenerator für Nascom 1

Einer der wesentlichen Unterschiede zwischen Nascom 1 und 2 ist das Kansas-City Interface, das letzterer zur externen Speicherung (Cassettenrecorder) verwendet. Dies bietet Vorteile einmal, weil es auch bei vielen anderen Rechnern vorhanden ist, und zum anderen, weil es neben höherer Datensicherheit auch eine Erhöhung der Geschwindigkeit erlaubt: Standard ist 300Bd, 1200Bd ist problemlos zu erreichen, 2400Bd soll auch zuverlässig gehen. Zur Definition des KC Codes sei hier nur kurz vermerkt, daß es ein serieller Code wie beim Nascom 1 ist, die Nullen und Einsen jedoch nicht durch Ton an/aus, sondern durch zwei Tonhöhen kodiert werden, und zwar High durch 2400 Hz (=8 Schwingungen bei 300Bd) und Low durch 1200 Hz. Erhöhung der Baudrate bis 1200Bd ist durch mehrfache Halbierung der Bitzelle möglich, oberhalb davon müssen die Frequenzen erhöht werden. Ich habe im wesentlichen das Nascom 2 Interface nachgebaut. Es weist im Gegensatz zu manch anderen Schaltungen 2 Vorteile auf: 1. Die Umschaltung der Frequenzen erfolgt synchron, d.h. im Nulldurchgang. Dadurch ist Vorteil 2 möglich: Durch Regeneration des Taktes mit einer PLL wird der Empfangstakt bei Geschwindigkeitsschwankungen des Bandes nachgeregelt, dadurch ist auch ein Cassettenaustausch zwischen Recordern mit Geschwindigkeitsabweichungen möglich! Allerdings habe ich diesen Vorteil bei mir bis jetzt noch garnicht ausgenutzt, auch so ist es bei 1200Bd (fast) völlig fehlerfrei.

Eine kurze Schaltungsbeschreibung: Zunächst einmal müssen die erforderlichen Baudraten bereitgestellt werden (74LS193 und 4024). Ich habe es gleich (im Gegensatz zum Nascom 2) so erweitert, daß von 300-9600Bd umgeschaltet werden kann. Das erleichtert den Anschluß an andere Rechner, Terminals usw. Das 74193 dividiert 2 Mhz durch 13 und erreicht damit haargenau 9600Bd, die weiteren werden durch den 4024 Teiler hergestellt. Der serielle Ausgang des UARTs (Tx

Data) schaltet das 4027, so daß je nach Bedarf der niedrige oder hohe Ton entsteht. Dann kommt noch ein simpler Tiefpaß, und ab geht's in den Recorder. Der Tiefpaß ist etwas nachteilig, der höhere Ton ist dadurch etwas leiser, aber es geht ja.

Im Empfangsteil (Empfindlichkeit 300 mV) werden die Töne linear verstärkt (4069), in einem Schmitt-Trigger geformt (1/4 4070) und durch eine Exor-Verzögerung in frequenzverdoppelte Nadelimpulse umgewandelt. Der 4520 mißt den Abstand zwischen je zwei solchen Impulsen und erkennt auf High oder Low (ein fehlender Impuls oder einer zuviel schadet nicht) und setzt das 4013 entsprechend (Empfangsdaten). Außerdem synchronisieren die Nadelimpulse eine 4046-PLL, indem raffinierterweise bei High 4 Impulse abgezogen werden (1/4 4011). Damit wird der Takt regeneriert und kann als Empfangstakt dienen (Pin 17 vom UART, muß für KC-Betrieb von Pin 40 getrennt werden).

Das ging ja etwas hastig, aber es ist eh schon zu lang. Der einzige Abgleich ist die Justierung des 10k-Potis, das das Taktverhältnis des getriggerten Empfangstones einstellt. Das sollte möglichst genau 1:1 sein, man kann aber erst einmal das Poti in der Mitte stehen lassen. Abgleich mit Oszil oder Reduzierung der Wiedergabeamplitude, bis es gerade noch geht, dann fehlerfrei einstellen. Für dieses Interface reicht übrigens i. a. die Amplitude am Diodenausgang aus. Der Nachbau müßte eigentlich recht einfach sein, die Probleme kommen bei der Umschaltung: Wenn man das Nascom 1 Format auch noch lesen können und den regenerierten Empfangstakt benutzen möchte, sind recht viele Kabel erforderlich. Da hilft nur: Über der Schaltung brüten, bis man sie verstanden hat, und dann keinen Fehler bei der Verdrahtung der Schalter machen. Es sind natürlich mehrere Möglichkeiten für die Schalter möglich: Dargestellt ist eine Lösung mit 2 Schaltern, wobei der eine (S1) zwischen Cassette und normaler (V24) serieller Schnittstelle mit verschiedenen Baudraten umschaltet und S2, der nur wirkt, wenn S1 auf "Cassette" steht, schaltet zwischen Nascom 1 - Format und Kansas-City 300 und 1200Bd um. Wenn man auf die Regenerierung des Empfangstaktes verzichtet, entfällt bei S2 eine Schaltebene und beim UART (IC 29) bleiben Pin 17 und 40, die man sonst auftrennen muß, miteinander

verbunden.

Zur Nachbausicherheit: Selbst wenn eine Platine verfügbar sein sollte, ist er dennoch nicht unproblematisch da man einige Eingriffe in die Nascom 1 - Schaltung manchen muß. Bei der Verdrahtung der Schalter kann viel schiefgehen. Der Abgleich hingegen ist einfach: Mit dem Trimmer in Mittelstellung muß es bei 300Bd schon fehlerlos gehen. Auf jeden Fall wird mit dem Kansas-City-Interface eine erhebliche Geschwindigkeitssteigerung bei herabgesetzter Fehlerrate erreicht und der gegenseitige Austausch von Cassetten erleichtert. Also denn viel Erfolg!

## PREISAUSSCHREIBEN

Nachdem unser erstes Preisausschreiben doch einige Leser aktiviert hat, interessante Beiträge einzusenden (einige sind in diesem Heft abgedruckt, andere folgen noch), wollen wir nun zum neuen Jahr einen weiteren Wettbewerb starten. Unter dem Thema "Und es bewegt sich doch..." (kommt mir irgendwie bekannt vor!) sollen Beiträge gesammelt werden, die sich mit Bewegung auf dem Bildschirm befassen.

Die Einsende- und Auswahlkriterien sind schnell erklärt: Es wird jedes Programm akzeptiert, das Bewegung auf dem Bildschirm erzeugt. Das könnten Spielprogramme sein, wissenschaftliche Programme mit graphischen Darstellungen oder was immer Ihnen zu dem Thema einfällt. Bewertet wird wieder nach dem Verhältnis von Aufwand und Wirkung, d.h. Speicherplatz und Effekt. Die Programme sollten möglichst mit NASSYS3 laufen, Maschinenprogramme und BASIC sind zugelassen. Um alle Leser teilhaben zu lassen, sollten keine speziellen Graphic-Zusatzkarten nötig sein, obwohl wir auch die Möglichkeit einer Graphic-Hardwareschaltung mit Pfiff als Teilnehmer ins Auge fassen.

Als Preis hat MKS ein EINKAUFS-GUTSCHEIN IM WERTE VON DM 200,- ausgesetzt, was als Anreiz dienen kann. Aber "die Teilnahme ist wichtig, nicht der Gewinn", soll einmal ein berühmter Programmierer gesagt haben. Viel Spaß beim Knobeln! (Oder haben Sie etwa schon ein fertiges Konzept?) Einsendeschluß ist der 1. März 1982, Zeit genug, um den Mitteilnehmern das Fürchten zu lehren.

Angeregt wurde das Preisausschreiben von Claus Stehlik, Judendorf/österreich. Sicher hat der schon etwas Raffiniertes auf Lager.

# CASSETTENINTERFACE von D. Maisl

Nachdem wir unseren Nascom 1 durch den Einbau zweier verschiedener Bussysteme erweiterungsfähig gemacht und im Laufe der Zeit auch grössere Programme zu laden hatten, war uns das Cassetteninterface des Nascom 1 zu langsam.

Eine erste Verbesserung brachte die Verdoppelung der ursprünglichen Ladegeschwindigkeit von knapp 300 Baud durch Umschalten der Taktgeschwindigkeit des UARTs (IC 2, Pin 11 auf IC 2, Pin 12)

Theoretisch wären auch noch andere Geschwindigkeiten möglich gewesen. Die Erfahrung zeigte aber, dass 1200 oder gar 2400 Baud mit diesem Schaltungskonzept nicht realisierbar sind.

Als günstige Möglichkeit bot sich dann der Nachbau des Cassetteninterfaces des Nascom 2 an.

Gleichzeitig hätte sich dann die Möglichkeit ergeben Programme sowohl auf dem Nascom 1 als auch auf dem Nascom 2 zu laden.

Zu unserer Überraschung stellte sich dann beim Probetrieb heraus, dass auch noch bei einer Geschwindigkeit von 2400 Bd, immerhin 8 mal schneller als das alte Interface, Daten sicher auf Cassette zu speichern waren.

Durch entsprechende Umschalter kann man sowohl das alte Cassetteninterface als auch das neue wahlweise benutzen. Es ist nur ein etwas höherer NF-Eingangspiegel erforderlich.

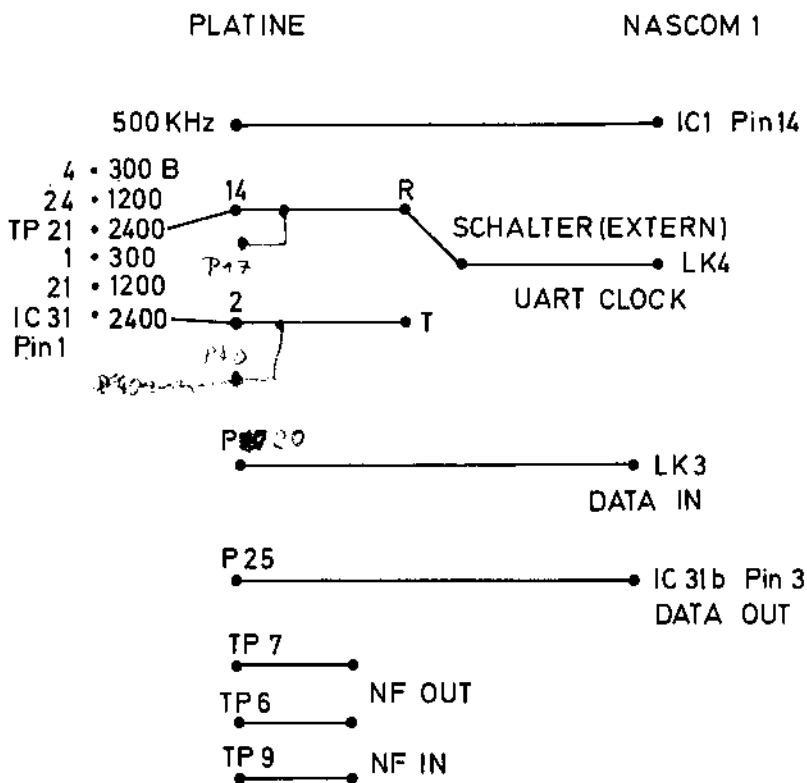
Funkamateure können Programme wesentlich schneller über Funk übertragen, Geschwindigkeitsschwankungen des Recorders machen sich nicht mehr so stark bemerkbar.

(Übrigens der neue UOSAT scheint ebenfalls ein dem Nascomformat und Tonlagen ähnliches Interface für seine Telemetrie zu verwenden)

Der Gesamtaufwand beträgt etwa 20 DM. Die CMOS Bausteine sind überall erhältlich, Platinen (einseitig, deshalb mit Drahtbrücken) können bei mir notfalls gegen Erstattung der Selbstkosten bezogen werden. Nachdem anscheinend beabsichtigt ist, sich auf ein Betriebssystem festzulegen, ist es vielleicht auch ganz sinnvoll eine Aufzeichnungsart zu verwenden.

Unsere Erfahrungen mit dem Interface sind recht gut, man muss nur einmal den Pegel richtig einstellen; er ist naturgemäss etwas höher als bei 300 Bd, die Fehlerrate ist nur unwesentlich grösser.

## CASSETTENINTERFACE 2400 BD



Der Umschalter kann eingespart werden, wenn am UART die Leiterbahn von P17 zu Pin40 aufgetrennt wird und PKT14 mit Pin17 UART sowie PKT2 mit Pin40 UART verbunden werden. Mit VR1 wird der Arbeitspunkt eingestellt.

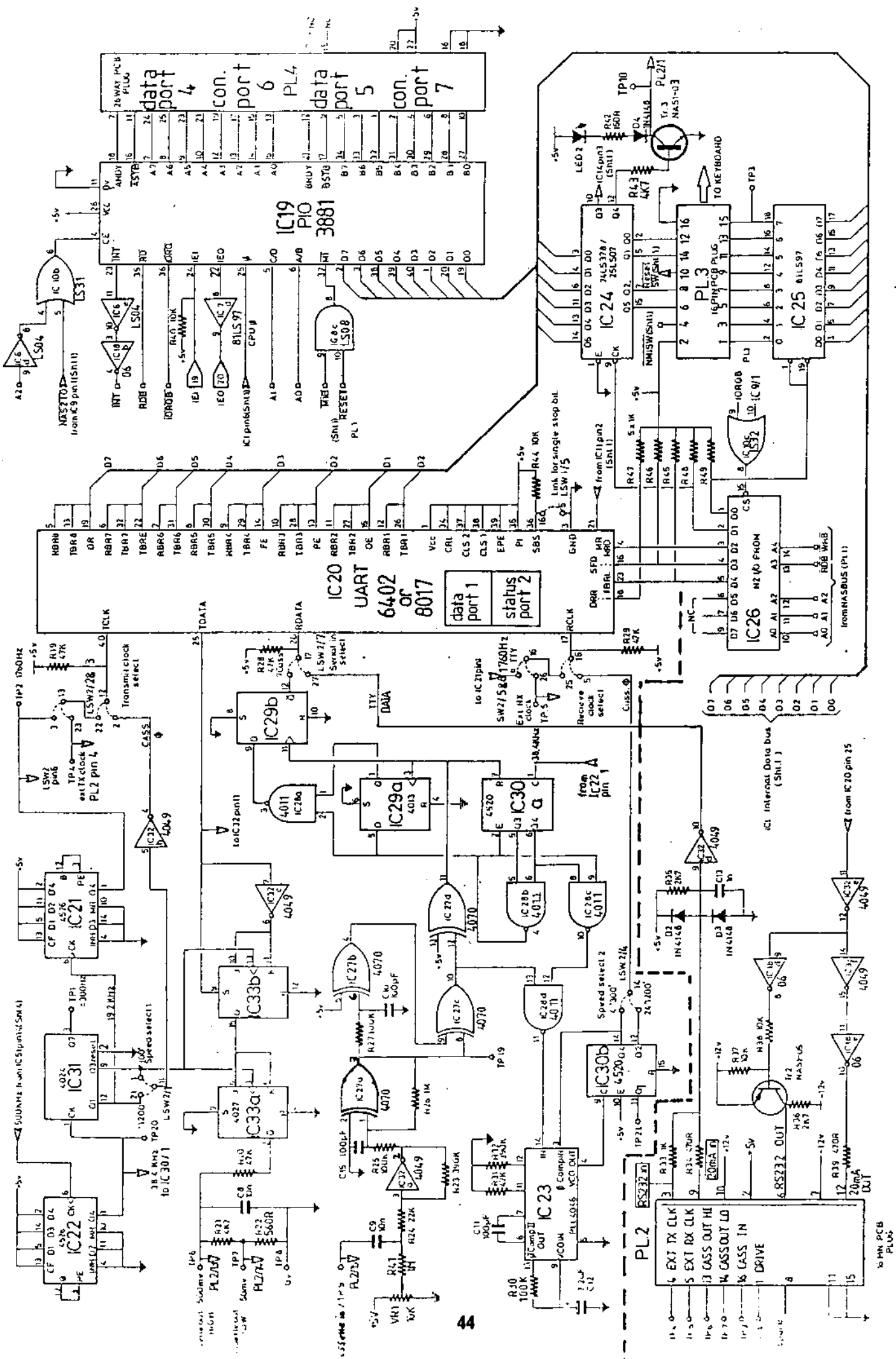
### LEERKASSETTEN

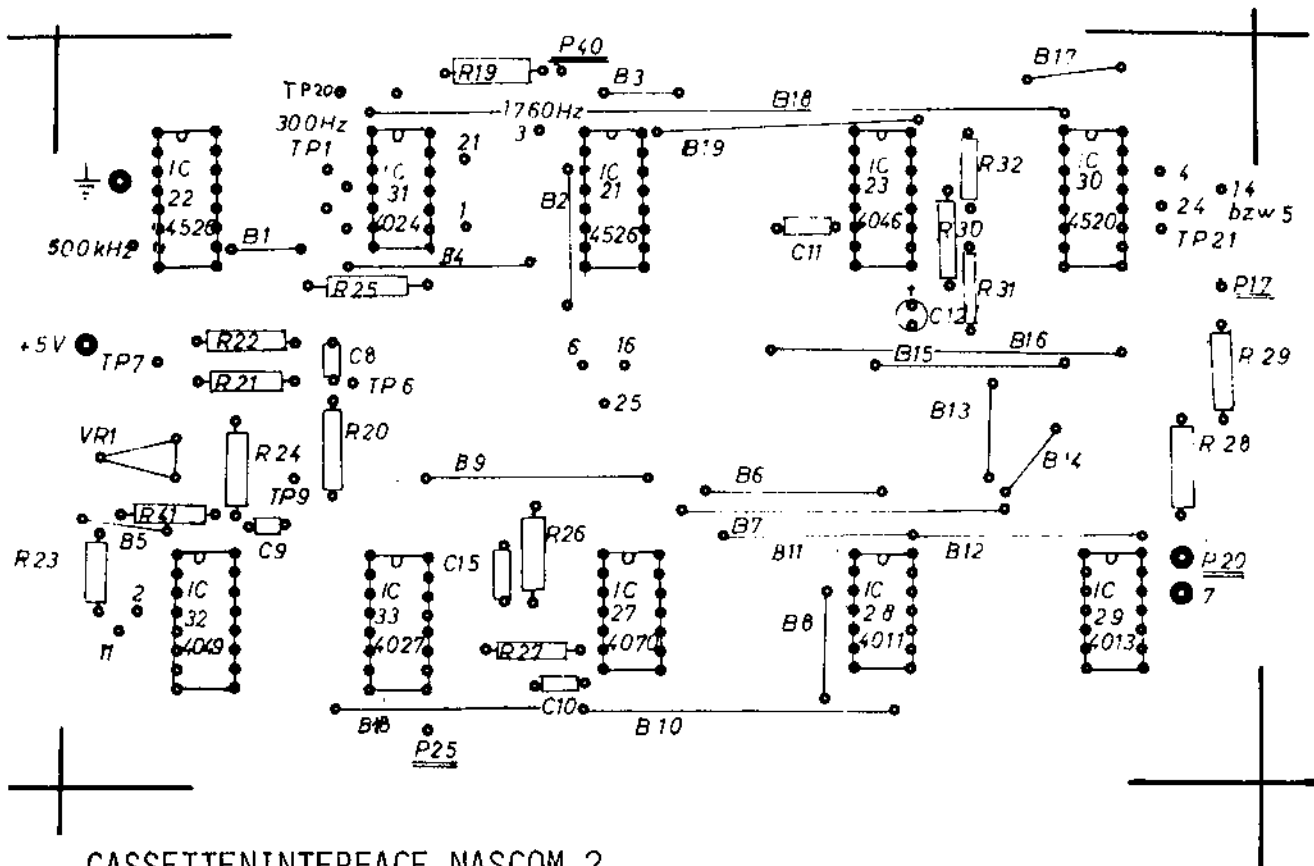
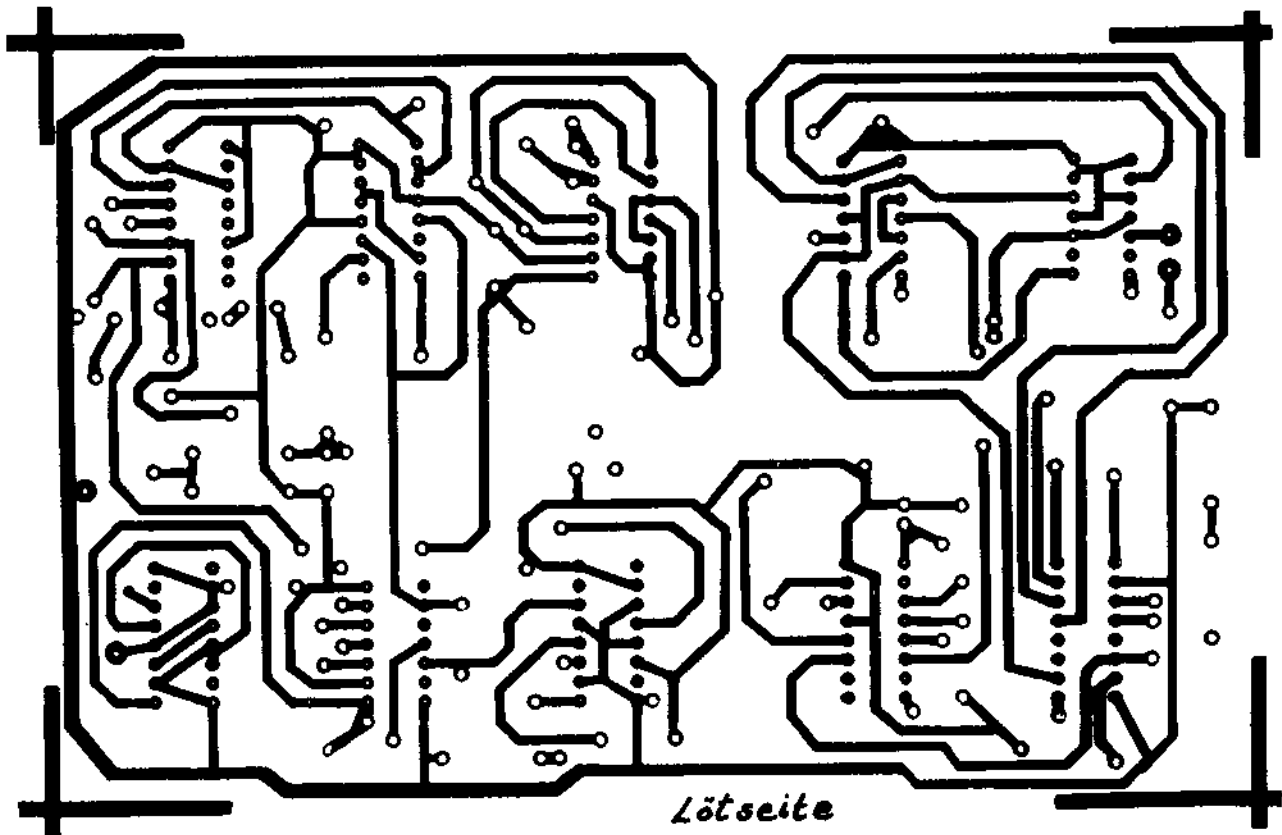


Speziell geeignet für Datenaufzeichnung. Hochwertiges BASF-Band. Cassette 5-fach verschraubt. Cassette C10, d.h. 10 Minuten spieldauer, daher besonders geeignet für Mikrorechnerprogramme.

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

Bei: M K - Systemtechnik





### CASSETTENINTERFACE NASCOM 2

Punkte mit der Bezeichnung P beziehen sich auf das UART  
 Alle Beschriftungen entsprechen dem NASCOM 2 Schaltplan

# Programme aus der Luft von Mary Jo Kostya

Radio Nederland, der Kurzwellen-Auslandsdienst aus Holland, hat am 10. September ein interessantes Experiment durchgeführt. Wohl zum erstenmal wurde ein Computerprogramm von einer Rundfunkstation über Kurzwelle übertragen - und das weltweit.

Das Programm 'BEAR', von John Campbell, Professor an der Exeter University, errechnet Abstand und Richtung zwischen zwei Punkten auf der Erde. Es wurde in drei Fassungen ausgestrahlt, in Kassettenformat für Tandy TRS-80 Modell I Level 2, Commodore und Apple II.

Die Idee kam von Jonathan Marks, Moderator der wöchentlichen Sendung 'Media Network', die sich mit Medienfragen und insbesondere mit dem Rundfunk-Fernempfang als Hobby (DXen) befaßt. RN sendet von Lopik/NL und über Relais von Madagaskar und Bonaire Niederl. Antillen. Mehrere Frequenzen stehen zur Auswahl.

Die beste Frequenz am 10. September in Frankfurt war im 21-MHz-Band. Ich habe die ganze Sendung aufgenommen auf Cassette, und ein paar Tage später spielte ich einem TRS 80 das BASIC-Programm vor. Der Tandy schien die Geräusche als Daten zu erkennen, aber verstehen wollte er nicht...

Nicht alle Zuhörer hatten Pech, 235 meldeten ihre Ergebnisse; 98 (42%) hatten Erfolg; einer mit Apple, 36 mit Commodore und 61 mit TRS-80. Von den Zuhörern mit Erfolg wohnen 82% in Europa; Meldungen kamen außerdem von den USA, Kanada und Belize.

Eine holländische Inlandssendung 'HOBBY-SCOOP' für Heimcomputerfans wird übrigens wöchentlich ausgestrahlt. Jeden Sonntag um 18.30 Uhr MEZ werden u.a. auf 1008 kHz (Mittelwelle) und 88,2 MHz (UKW) Programme in 'HOBBYSCOOP-Esperanto' übertragen. Dieses Verfahren, das zwei Töne (1200 und 2400 Hz) benutzt, wurde von holländischen Computerfans zusammen mit dem Leiter der Sendung, Hans Janssen, entwickelt. Sync-Pulse sorgen für einen ziemlich zuverlässigen Empfang. Mehrere Kleincomputer, z.B. APPLE, brauchen nur ein Umsetzprogramm, um das 'Esperanto' dekodieren zu können. Für TRS-80 gibt es Baupläne für ein kleines Interface (etwas Ähnliches werden unsere NASCOMs wohl auch

nötig haben). APPLE-Programm und Baupläne gibt es gegen Selbstkosten via: Hans Janssen, HOBBYSCOOP, Postbus 10, NL-1200 JG Hilversum.

Das nächste Experiment auf Kurzwelle ist für den 28. Januar 1982 geplant. Sendezeiten und voraussichtliche Frequenzen für Europa, in MEZ und kHz, sind:

10.50	15560, 11930, 9895, 6045, 5955
14.50	17605, 11930, 9895, 6045, 5955
21.50	21685, 17695, 17605, 15220, 9715

Es ist ratsam, eine halbe Stunde vor der Sendung mit Frequenzvergleichen anzufangen, um die beste festzustellen. Die Übertragung muß direkt vom Empfänger auf das Cassettengerät erfolgen. Die Bandbreite, falls einstellbar, sollte mindestens 5 kHz betragen; Höhen aufdrehen, Lautstärke auf das Maximum ohne Verzerrung einstellen. Ein Versuch ist es wert, die Cassette auf eine andere zu kopieren; mehreren Teilnehmern hat's zum Erfolg geholfen.

Kassettenformate am 28. Januar werden sein: Sinclair ZX-81, TRS-80, Commodore und HOBBY-SCOOP-Esperanto, evtl. noch Atari. Das zu übertragende Programm befaßt sich mit Sonnenauf- und -untergangszeiten. Viel Erfolg! Das BEAR-Programm habe ich für den NASCOM eingetippt - es geht. Einzugeben sind die Koordinaten für den Empfängerstandort (RCVR) und einen Senderstandort (XMTR). Die Beispiele zeigen Frankfurt und Lopik bzw. Quito/Ecuador.

```
MJK-SYS 1
E100C
OK
LIST

100 DIM A(2),B(2)
110 R=3953
120 F=3.1416
130 PRINT " "
140 K=1
150 FOR J=K TO 2
160 IF J=1 THEN 190
170 Z$="XMTR"
180 GOTO 200
190 Z$="RCVR"
200 PRINT " "
210 PRINT Z$,"LAT.:", "(DEGS","MINS)"
220 INPUT A(J),X
230 A(J)=P*(A(J)+X/60)/180
240 PRINT "N OR","S?"
250 INPUT S$
260 IF S$="N" THEN 300
270 IF S$="S" THEN 290
280 GOTO 270
290 A(J)=-1*A(J)
300 PRINT " "
```

```

310 PRINT Z$, "LONG.", "(DEGS", "MINS)"
320 INPUT B(J), X
330 B(J) = P * (B(J) + X / 60) / 180
340 PRINT "E OR", "W?"
350 INPUT S$
360 IF S$ = "E" THEN 400
370 IF S$ = "W" THEN 390
380 GOTO 340
390 B(J) = -1 * B(J)
400 NEXT J
410 PRINT " "
420 C = COS(A(2))
430 X = C * COS(B(2))
440 C = C * SIN(B(2))
450 D = SIN(A(2))
460 H = SIN(A(1))
470 G = COS(B(1))
480 J = SIN(B(1))
490 K = COS(A(1))
500 W = G * X + J * C
510 E = H * W - K * D
520 F = G * C - J * X
530 G = K * W + H * D
540 IF ABS(G) = 1 THEN 600
550 W = 1 - G * G
560 IF W > 0 THEN 580
570 GOTO 600
580 H = ATN(G / SQR(W))
590 GOTO 610
600 H = G * P / 2
610 IF ABS(G) = 1 THEN 760
620 IF W <= 0 THEN 760
630 IF E > 0 THEN 740
640 IF E < 0 THEN 700
650 IF F > 0 THEN 680
660 X = -0.5 * F
670 GOTO 780
680 X = P / 2
690 GOTO 780
700 X = ATN(F / E) - P
710 IF F < 0 THEN 780
720 X = X + 2 * P
730 GOTO 780
740 X = ATN(F / E)
750 GOTO 780
760 PRINT "ANY", "ANGLE", "D.K."
770 GOTO 830
780 D = 180 * (P - X) / P
790 C = INT(D)
800 IF D - C < 0.5 THEN 820
810 C = C + 1
820 PRINT "ANGLE", C, "DEGS."
830 PRINT " AND"
840 D = R * (0.5 * P - H)
850 C = INT(D)
860 IF D - C < 0.5 THEN 880
870 C = C + 1
880 PRINT "RANGE", C, "MILES"
890 PRINT " "
900 PRINT "NEW", "XMTR?", "(YES", "OR", "NO)"
910 INPUT S$
920 PRINT " "
930 IF S$ = "YES" THEN 1020
940 IF S$ = "NO" THEN 960
950 GOTO 900
960 PRINT "NEW", "RCVR?", "(YES", "OR", "NO)"
970 INPUT S$
980 PRINT " "
990 IF S$ = "YES" THEN 140
1000 IF S$ <> "NO" THEN 960
1010 GOTO 1040
1020 K = 2
1030 GOTO 150
1040 STOP
1050 END

```

## Hochaufl. Graphic Teil 2 von H.-Martin Pohl

Der normale Video-Teil erzeugt dauernd Adressen, die über Multiplexer dem Video-RAM zugeführt werden, wenn nicht gerade der Prozessor auf diesen Ram-Bereich zugreift. Die Daten aus dem Video-RAM werden als Teiladressen für den Character-Generator benutzt, der restliche Adreßteil wird auch durch Zähler erzeugt. Die Daten aus dem Character-Generator werden in das Schieberegister geladen und als Bildsignal herausgeschoben. Über Widerstände (und Dioden) wird das Synchronisationssignal addiert und es entsteht das Video-Signal.

Da die hochauflösende Graphik jeden Bildpunkt einzeln ansprechen soll, müssen die Daten für das Schieberegister direkt aus einem RAM-Bereich stammen. Die erzeugten Takte, die als Adresse für Video-RAM und als Teiladresse für den Zeichen-Generator verwendet werden, können dann zusammen als Adresse für den Graphik-RAM-Bereich dienen. Da beliebige freie dynamische RAMs verwendet werden sollen, kann die Graphik nur über DMA arbeiten. Wenn der Prozessor den Bus freigibt, muß über Puffer die Graphikadresse auf den Adreßbus gegeben werden. Die Daten vom Datenbus können dann (über ein zweites Schieberegister, dessen LD- und Clock-Eingang parallel zu dem des Videoteils geschaltet sind) als Graphikbildsignal verwendet werden.

Zur Anforderung eines DMA kann das VBLANK-Signal verwendet werden. Dann kann der Prozessor in der Zeit zwischen der Ausgabe zweier Bilder normal weiterarbeiten. Da die hochauflösende Graphik immer ziemlich lange auf den Speicher zugreift, muß sie auch ein Refresh-Lesen durchführen. Das kann mit den verwendeten Graphik-Zählern erfolgen, da diese ja alle Adressen gleichmäßig durchzählen. Es muß also nur noch ein MREQ-, Read- und Refresh-Signal während der Speicherzugriffszeit erzeugt werden. Für Refresh kann man den normalen 1MHz-Takt verwenden, für MREQ 2 oder 4 MHz, und für Rd einfach  $\emptyset$ V, da immer gelesen wird.

Ein Problem taucht auf beim Anfordern vom Bus bzw. beim Zurückgeben an den Prozessor: es ändern sich einige Speicherzelleninhalte. Mit Hilfe der eingezeichneten Flip-Flops

ließ sich dieser Mangel jedoch beheben. Die mit PL1 bezeichneten Leitungen sind NASBUS-Leitungen. Sie können jedoch nicht alle so verwendet werden, da sie teilweise schon anders belegt sind. (s. NASBUS Functional Specification).

Der Videoteil bildet normalerweise  $48 * 8 = 384$  Bildpunkte in einer Zeile ab. Dabei werden jedoch  $64 * 8 = 512$  Punkte adressiert, der Überschuß verschwindet in den Randbereichen. Um bei der hochauflösenden Graphik nicht auch ein Viertel des Speichers als Rand zu verschenken, habe ich mich auf 256 Punkte Breite beschränkt. Das wird erreicht, indem man den 15,8kHz-Takt nicht anschließt und für Adressbit 6 bis 15 den jeweils langsameren Takt verwendet. Dann erscheint allerdings ein Teil des Bildes doppelt.

Ich habe meine Schaltung auf einer Lochrasterplatine aufgebaut und mit einem Stecker für den NASBUS versehen, um leicht Änderungen daran durchführen zu können. Dabei sind mir allerdings die Puffer (IC 1 - 3) etwas zu dicht an den NASBUS-Stecker geraten und es entstand ein unübersichtlicher Drahtverhaue, weil fast alle Pins eine extra Leitung benötigen.

Der Schaltplan gilt für den N2. Für N1 Benutzer gelten folgende Änderungen:

Die Adreßtakte sind an IC 1 - 4 abzugreifen, dabei müssen die Frequenzen übereinstimmen. Folgende Frequenzen sind an folgenden ICs verfügbar, und weil nicht immer dort, nicht im Schaltplan angegeben:

1kHz an IC 3/13  
 500 Hz 3/12  
 250 Hz 3/11  
 125 Hz 4/14  
 62,5 Hz 4/13

Die Einspeisung des Graphik-Bildsignals kann auf verschiedene Weisen erfolgen. Man kann es einfach per Schalter statt dem Textbildsignal von IC 15/9 nach IC 11/10,11 einspeisen. Dann kann man entweder Graphik oder Text auf dem Bildschirm sehen. Die andere Möglichkeit ist, daß man es zu dem Videosignal an P2 addiert. Dazu muß man es genauso wie das Textsignal über eine Diode und einen Widerstand an P2 anschließen. Damit das Helligkeitsverhältnis zwischen Graphik und Text eingestellt werden kann, und das Synchronisationssignal bei eingeschalteter Graphik stark genug bleibt, muß R8 und der

entsprechende Widerstand für die Graphik durch ein 1k-Poti ersetzt werden.

Das VBLANK-Signal kann an IC11/4 abgegriffen werden.

## BESTELLSERVICE

Folgende Platinen für den PIO-BUS sind bereits erhältlich:

2716 Prommer Platine DM 35.-

2716 Prommer Bausatz DM 50.-

bei Rüdiger Maurer

Taunusstein

Cassetteninterface gegen Selbstkosten bei D.Maisl

Göppingen

AD/Wandler gegen Selbstkosten bei Peter Bentz

Mülheim/Ruhr

Folgende Software in EPROMs ist erhältlich:

2716 Prommer in 1x2708 DM 20.-

bei Rüdiger Maurer

Adresse siehe oben

MDCR Monitor "UNICON"

Standardversion in 1x2716 DM 65.-

Spezialversion (individ.) DM 95.-

bei Johannes C. Lotter

Darmstadt

Tel. [REDACTED]

Folgende Layouts können als Ätzfolien zum Selbstkostenpreis (DM 5.- + Porto) über die Redaktion bezogen werden:

MDCR Interface

Monitor Umschaltkarte

A/D Wandler

FSK-Modem 300 Baud

Soundgenerator



# Nascom Pio-Bus

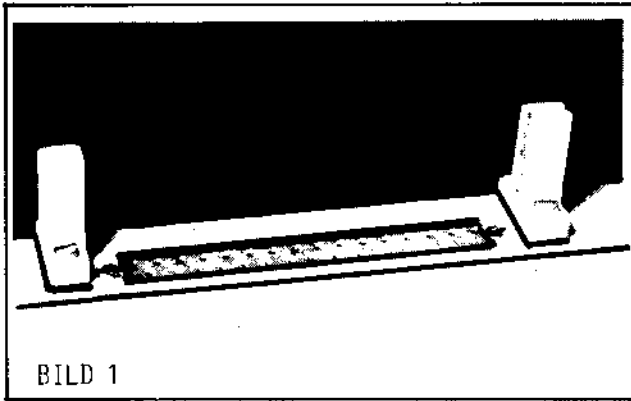


BILD 1

Ein Leser fand die Bezeichnung "BUS" nicht ganz gerechtfertigt und schlug "PIO Schnittstelle" vor. NPS (für NASCOM PIO Schnittstelle) klingt zwar gar nicht schlecht, da es sich aber unter den Lesern schon ziemlich eingebürgert hat, bleiben wir vielleicht doch lieber bei der alten Bezeichnung.

zeigt, wie problemlos der EPROMMER von Herrn Ploss eingeschoben werden kann, ohne störende Kabel. Schnell kann er gegen irgend eine andere Schaltung ausgetauscht werden, die über den selben Port angesteuert wird.

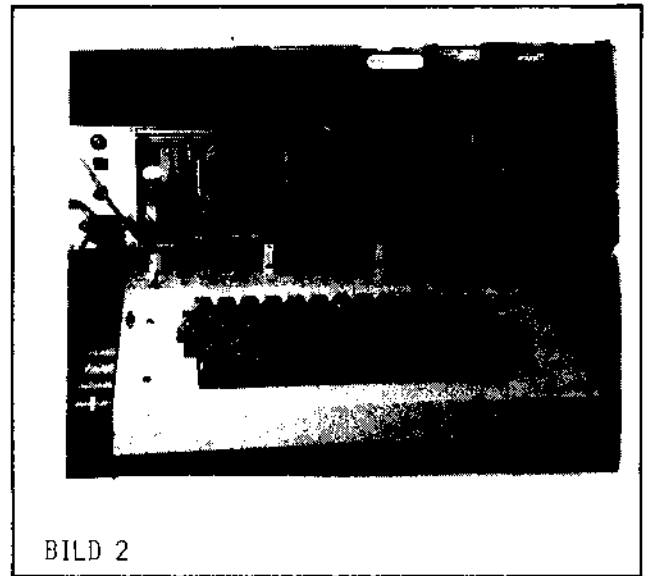


BILD 2

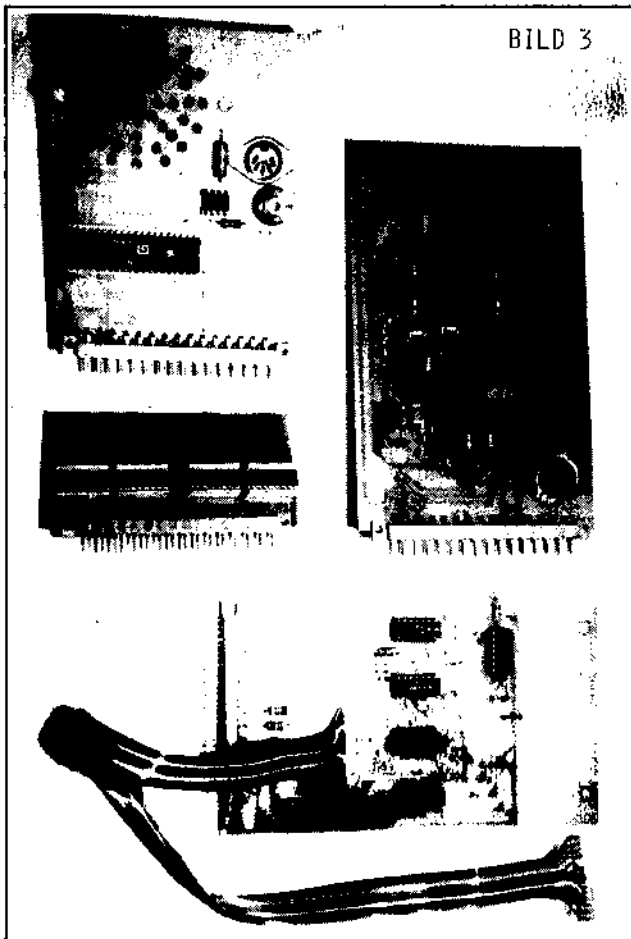


BILD 3

Hier nun ein Vorschlag zum Einbau einer 31-pol. Buchse nach DIN 41617 in ein Tastaturgehäuse. In Bild 3 sehen Sie die Platinen des Soundgenerators, eines FSK Modems (nach ELEKTOR), einer 7-Segment-Anzeige und des MDCR-Interfaces von Herrn Lotter. Bild 2

## Disassembler — Anpassung von Jürgen Weiermann

Und hier vielleicht noch eine kleine Programmidee für diejenigen unter uns, die noch keinen Disassembler ihr Eigen nennen können. Der von Herrn Deege umgeschriebene Disassembler läuft nicht schlecht, und so habe ich ihn dazu benutzt, um bei der Erstellung dieser Cassette einiges an Zeit einzusparen. Die Listings der Programme CONMOD und GKSU wurden vom Disassembler direkt in den Textspeicher des Formatierprogramms geschrieben und konnten danach an das NAS-SYS Format mühelos angepaßt werden. Anstelle des Aufrufs von ROUT (RST 30) wurde der Unterprogrammaufruf CD 20 40 eingeschrieben. Das Programm sieht dann so aus:

```

4020 E5          PUSH HL
4021 32 00 10    LD(HL), A
4024 F7          RST ROUT
4025 2A 22 40    LD HL, (4022)
4028 23          INC HL
4029 22 22 40    LD (4022), HL
402C E1          POP HL
402D C9          RET
    
```

```

10 REM ***** QUEST ***** 16.09.81
20 REM
30 REM Aus Byte July 1979 p.176ff
40 REM by Roger Chaffee
50 REM
60 REM Vereinfachte Version von "ADVENTURE".
70 REM
80 REM Übersetzt und an Nascom angepaßt von
90 REM Michael Bach.
100 REM
110 REM VARIABLES:
120 REM N NODE (CAVE) NUMBER
130 REM MO MOVE COUNTER
140 REM T CURRENT LOCATION OF TREASURE
150 REM (-1 IF CARRIED)
160 REM T1,T2 FIRST & 2ND HIDING PLACE
170 REM M6 SAVES THE MOVE NUMBER WHEN HE SAID
180 REM NO, HE DID'NT WANT THE TREASURE
190 REM M MAP OF INTERCONNECTIONS,
200 REM M(I,J) IS NEXT NODE FROM NODE J,
210 REM WHEN YOU GO N,E,U,D,W,S FOR I=1 TO 6
220 REM
230 REM BOUNCES:
240 REM TO GO TO NODE -2 MEANS BOUNCE BACK TO
250 REM THE NODE YOU CAME FROM.
260 REM
270 REM FORCED AND/OR RANDOM MOVES:
280 REM M(1,N)=-2; A FORCED MOVE WHEN YOU REACH
290 REM NODE N. THEN, M(2,N)% OF THE TIME YOU
300 REM GO TO NODE M(3,N). IF YOU DON'T GO
310 REM THERE, M(4,N)% OF THE TIME YOU GO TO
320 REM NODE M(5,N), AND THE REST OF THE TIME
330 REM YOU GO TO NODE M(6,N).
340 REM
350 REM NODES*100:
360 REM NODE N+100 MEANS NODE N IF YOU DON'T
370 REM HAVE THE TREASURE, AND NODE N+1 IF
380 REM NODE N+200 MEANS NODE N+1 IF YOU HAV
390 REM THE TREASURE THE SECOND TIME, AND
400 REM NODE N OTHERWISE.
410 REM NODE N+500 MEANS RUN THROUGH A DELAY
420 REM AND THEN GOTO NODE N. THIS IS USED
430 REM FOR FALLING DOWN THE EXIT CHUTE,
440 REM AND YOU MAY WANT TO ADJUST.
450 REM
460 REM Q$ INPUT STRING
470 REM A$ CHARACTERS TO MATCH
480 REM A1 OUTPUT FROM INPUT ROUTINE
490 REM P PIRATE FLAG: 1 IF HE GOT YOU, 0, 0
500 REM P1 COUNTER FOR PIRATE ROUTINE
510 REM N9 SAVES OLD NODE IN MOVE ROUTINE FOR B
520 REM N8 SAVES NODE WE BOUNCED FROM IN MOVE
530 REM ROUTINE FOR PRINT FLAG
540 REM NO SAVES OLD NODE IN MOVE ROUTINE
550 REM FOR DEAD END
560 REM A0 SAVES OLD DIRECTION IN MOVE ROUTINE
570 REM D DEBUG FLAG (µ*1 TO PRINT)
580 REM I,J MISCELLANEOUS CONTERS
590 REM W TRAVEL FLAG, USED IN SCORING;
600 REM W(I)=1 IF HE'S BEEN TO NODE I, 0, 0
610 REM S SCORE
620 REM M9 MAXIMUM NUMBER OF NODES
630 REM
640 D=0:REM ZUM DEBUGGEN' D=1
650 REM-----
660 REM
670 CLS:PRINT:A$="**** QUEST *****"
680 FOR J=1 TO LEN(A$)
690 POKE J+3030,ASC(MID$(A$,J,1)):NEXT
700 PRINT"Du gingst durch den Wald, und Du ";
710 PRINT"fandest hinter"
720 PRINT"einem Busch versteckt den Eingang ";
730 PRINT"zu einer":PRINT"Höhle."
740 PRINT"Es heißt, daß vor vielen Jahren ";
750 PRINT"ein Pirat hier"
760 PRINT"seinen Schatz versteckt hat, aber er";
770 PRINT"ist nie":PRINT"gefunden worden,"
780 PRINT"Die unermeßlichen Reichtümer könnten";
790 PRINT"noch immer":PRINT"da sein."
800 PRINT"Wär doch mal was?!":PRINT
810 PRINT"Tip: Mach unterwegs ne Karte (3 dim.)
"

```

```

820 READ M9,T1,T2
830 DIM W(42),M(6,42)
840 REM READ MAP INTO M ARRAY
850 FOR I=1 TO M9
860 READ N
870 IF I=N THEN890
880 PRINT"DATABASE PROBLEM"!,N:STOP
890 FOR J=1 TO 6
900 READ M(J,I)
910 NEXT J:NEXT I
920 PRINT:PRINT:INPUT"Fertig gelesen";Q$
930 FOR J=1 TO 11:PRINT:NEXT:GOSUB2370
940 REM-----
950 N=5:MO=0:M6=0:T=T1:P=0:P1=0
960 FOR J=1 TO M9:W(J)=0:NEXT:PRINT
970 REM DESCRIBE
980 GOSUB2440
990 REM-----
1000 REM MAIN LOOP STARTS HERE
1010 REM COUNT MOVES
1020 MO=MO+1
1030 REM MOVE
1040 GOSUB1770
1050 REM CHECK FOR FINDING THE TREASURE
1060 GOSUB1220
1070 REM TRY THE PIRATE
1080 GOSUB1410
1090 REM LOOP UNLESS FINISHED
1100 IF T=0 THEN990
1110 IF Nµ5 THEN990
1120 REM CALCULATE SCORE
1130 GOSUB1350
1140 PRINT:PRINT"Herzlichen Glückwunsch,"
1150 PRINT"Du hast den Schatz gefunden!"
1160 PRINT"In"MO"Zügen und Du hast"S+10"Punkte."
1170 PRINT:PRINT"Nach eine Schatzsuche";
1180 A$="JN":GOSUB1630
1190 ON A1 GOTO190,1200,1200
1200 PRINT:PRINT"Na denn Tschüß bis zum nächstem
Mal."
1210 END
1220 REM-----
1230 REM FOUND?
1240 IF TµN OR TµO OR M6+5µMO THEN RETURN
1250 PRINT"Möchtest Du ihn mitnehmen";
1260 A$="JN":GOSUB1630
1270 ON A1 GOTO1290,1310
1280 PRINT"Nun?":GOTO1260
1290 T=-1:PRINT
1300 PRINT"Ok, dann nichts wie weg!":RETURN
1310 PRINT
1320 PRINT"Dann lassen wir ihn hier und Du";
1330 PRINT" kannst noch":PRINT"weiter erkunden."
1340 M6=MO:RETURN
1350 REM-----
1360 REM
1370 S=0:IF T=-1 THEN S=S+5
1380 IF P=1 THEN S=S+10
1390 FOR J=2 TO M9:S=S+W(J):NEXT
1400 RETURN
1410 REM-----
1420 REM
1430 IF N=T2 OR P=1 THEN RETURN
1440 IF T1=T2 OR Tµ-1 THEN RETURN
1450 REM HE'S AT THE EXIT
1460 IF N=16 THEN P=160
1470 REM COUNT MOVES SINCE HITTING TUNNEL
1480 IF P1=0 THEN P1=P1+1
1490 IF N=3 THEN P1=P1+1
1500 REM GIVE HIM A FEW MORE MOVES
1510 IF P1µ15 THEN RETURN
1520 PRINT
1530 PRINT"Plötzlich springt der Pirat aus dem";
1540 PRINT"Dunkel"
1550 PRINT"und schnappt sich den Schatz."
1560 PRINT"Hah', ruft er, Du hast also meinen"
;
1570 PRINT" Schatz":PRINT"gefunden?!"
1580 PRINT"Jetzt werde ich ihn aber besser";
1590 PRINT" verstecken!"
1600 PRINT"Und er verschwindet um die Ecke";
1610 PRINT" mit dem":PRINT"Schatz."
1620 P=1:T=T2:RETURN
1630 REM-----

```

```

1640 REM INPUT
1650 REM FIRST CHARACTER OF AN INPUT STRING IS
1660 REM COMPARED WITH THE LETTERS OF A$, AND IF
1670 REM THERE IS A MATCH, THE INDEX IN A$ IS
1680 REM RETURNED IN A1. IF NO MATCH, LEN(A$)+1
1690 REM IS RETURNED.
1700 REM GET INPUT
1710 Q$="":INPUT Q$
1720 REM USE ONLY FIRST CHARACTER
1730 Q$=LEFT$(Q$,1)
1740 FOR A1=1 TO LEN(A$)
1750 IF MID$(A$,A1,1)=Q$ THEN RETURN
1760 NEXT A1:RETURN
1770 REM-----
1780 REM MOVE
1790 REM REMEMBER WHERE WE ARE, FOR BOUNCE
1800 N9=N
1810 REM SET N8 TO ANYTHING BUT YOU CANT GO THER
E
1820 N8=0
1830 REM ASK WHICH WAY
1840 GOSUB2190
1850 REM REMEMBER WHERE WE ARE UNLESS A DEAD END
1860 IF N=1 THEN1880
1870 N0=N:A0=A1
1880 PRINT:I=M(A1,N)
1890 IF I=-2 THEN I=N9
1900 IF D0=0 THEN PRINT"          DEBUG";N;"TO";I
1910 IF I=500 THEN1940
1920 REM DELAY LOOP TO WASTE SOME TIME
1930 I=I-500:FOR J=0 TO 2000: NEXT: GOTO1890
1940 ON I/100 GOTO1970,2000
1950 REM NORMAL ROUTINE--LESS THEN 100
1960 N=I: GOTO2020
1970 REM N+100, ADD ONE IF WITH TREASURE
1980 N=I-100:IF T=-1 THEN N=N+1
1990 GOTO2020
2000 REM N+200, ADD1 IF TREASURE FOR 2ND TIME
2010 N=I-200:IF T=-1 THEN N=N+P
2020 IF N=1 THEN2060
2030 REM DEAD END, TURN IT SO YOU GET OUT OTHER
2040 FOR J=1 TO 6:M(J,N)=2:NEXT
2050 M(7-A0,N)=N0
2060 REM PRINT OUT THE NODE DESCRIPTION
2070 IF N8=2 THEN GOSUB2440
2080 REM REMEMBER WE'VE BEEN HERE
2090 W(N)=1:N8=N
2100 IF M(1,N)=2 THEN RETURN
2110 REM FORCED MOVE WITH RANDOM DESTINATIONS
2120 I=M(6,N)
2130 IF M(4,N)*100*RND(1) THEN I=M(5,N)
2140 IF M(2,N)*100*RND(1) THEN I=M(3,N)
2150 IF D0=0 THEN PRINT"          DEBUG BOUNCE TO"
;I
2160 REM NOW HAVE A NEW DESTINATION
2170 REM GO BACK AND REDO IT
2180 GOTO1890
2190 REM-----
2200 REM WHICH WAY?
2210 PRINT
2220 PRINT"          Wohin";
2230 A$="NOHRWSP":GOSUB1630
2240 IF A1=8 THEN2300
2250 PRINT"Wohin willst Du gehen?"
2260 REM GIVE INSTRUCTIONS
2270 GOSUB2370
2280 REM DESCRIBE THE LOCATION AGAIN
2290 GOSUB2440:GOTO2210
2300 IF A1=7 THEN2360
2310 REM CALCULATE AND PRINT SCORE
2320 GOSUB1350
2330 PRINT"Duhast"S"Punkte!"
2340 REM START AGAIN
2350 GOTO2210
2360 RETURN
2370 REM-----
2380 REM SUBROUTINE TO GIVE INSTRUCTIONS
2390 PRINT"Tippe N,S,O,W,H oder R für:"
2400 PRINT"Norden, Süden, Osten, Westen, ";
2410 PRINT"Hoch, Runter"
2420 PRINT" oder P für den Spielstand (Punktzahl
),"
2430 RETURN
2440 REM-----

```

```

2450 REM DESCRIBE CURRENT LOCATION
2460 I=INT(N/5):J=N-5*I+1
2470 ON I+1 GOTO2490,2500,2510,2520,2530
2480 ON I-4 GOTO2540,2550,2560,2570,2580
2490 ON J GOTO2680,2720,2750,2780,2820
2500 ON J GOTO2850,2890,2930,2980,3010
2510 ON J GOTO3030,3070,3110,3140,3200
2520 ON J GOTO3250,3290,3350,3390,3430
2530 ON J GOTO3480,3540,3600,3620,3660
2540 ON J GOTO3690,3730,3780,3820,3840
2550 ON J GOTO3890,3950,4000,4040,4070
2560 ON J GOTO4110,4150,4190,4230,4300
2570 ON J GOTO4340,4370,4410,65529,65529
2580 ON J GOTO65529,65529,65529,65529,65529
2590 IF T=0 THEN2610
2600 PRINT:PRINT"Der Schatz ist hier!"
2610 IF T=2 THEN2670
2620 IF T1=T2 THEN2670
2630 IF T1=0 THEN2670
2640 PRINT:PRINT"Eine Schrift an der Wand sagt:"
2650 PRINT" Piraten lassen ihren Schatz nie"
2660 PRINT" zweimal am selben Ort."
2670 RETURN
2680 REM-----
2690 REM FIRST DATA IS NUMBER OF NODES, AND
2700 REM 1ST & 2ND HIDING PLACE
2710 DATA 42,23,12
2720 DATA 1,0,0,0,0,0,0
2730 PRINT"Duh bist in einer Sackgasse!"
2740 GOTO2590
2750 DATA 2,-2,101,-2,0,0,0
2760 PRINT"Indie Richtung kannst Du nicht.":PRI
NT
2770 GOTO2590
2780 DATA 3,33,2,1,10,106,4
2790 PRINT"Ein Tunnel verläuft nord-südlich,;"
2800 PRINT" im Westen ist":PRINT"eine Öffnung."
2810 GOTO2590
2820 DATA 4,3,30,2,11,2,1
2830 PRINT"Duh bist am Rand eines schwarzen Schac
hts."
2840 GOTO2590
2850 DATA 5,8,8,15,10,8,16
2860 PRINT"Duh bist vor der Höhle, ";
2870 PRINT"nach Süden geht's rein."
2880 GOTO2590
2890 DATA 6,16,3,2,10,2,2
2900 PRINT"Hier haust der große Troll."
2910 PRINT"Zum Glück ist er grade nicht zuhaus."
2920 GOTO2590
2930 DATA 7,-2,101,-2,0,0,0
2940 PRINT"Der große Troll ist da!"
2950 PRINT"Sieh zu, daß Du wekommst!"
2960 PRINT
2970 GOTO2590
2980 DATA 8,18,18,15,10,18,9
2990 PRINT"Duhast Dich im Wald verlaufen."
3000 GOTO2590
3010 DATA 9,-2,33,5,1,0,-2
3020 GOTO2590
3030 DATA 10,-2,101,-2,0,0,0
3040 PRINT"Duh kommst nicht weit, wenn Du durch"
3050 PRINT"den Fels willst.":PRINT
3060 GOTO2590
3070 DATA 11,1,13,4,2,1,2
3080 PRINT"Duh bist am Grund eines Schachtes."
3090 PRINT"Ein Bach rieselt am Boden entlang."
3100 GOTO2590
3110 DATA 12,36,2,1,2,1,2
3120 PRINT"Jetzt bist Du in einer Sackgasse!"
3130 GOTO2590
3140 DATA 13,2,37,2,1,11,14
3150 PRINT"Hier ist eine Erweiterung der Höhle."
3160 PRINT"Man sieht eine ruhige Stelle, wo Jema
nd"
3170 PRINT"eine Fackel an die Wand gelehnt hat."
3180 PRINT"über Dir hängen zackige Felsen."
3190 GOTO2590
3200 DATA 14,13,1,19,2,31,31
3210 PRINT"Duh bist in einer Schlucht, Hoch ";
3220 PRINT"über Dir":PRINT"steht an der Wand:"
3230 PRINT"          Bilbo war hier."
3240 GOTO2590
3250 DATA 15,-2,101,-2,0,0,0

```

```

3260 PRINT"Du kannst doch nicht fliegen, Du bist
3270 PRINT"doch kein Vogel!":PRINT
3280 GOTO2590
3290 DATA 16,5,33,2,10,1,106
3300 PRINT"Du bist in einer niedrigen Kammer. Ei
n"
3310 PRINT"enger Tunnel geht nach Osten und Du"
3320 PRINT"kannst auch nach Süden oder Westen ge
hn."
3330 PRINT"Man sieht einen Lichtschimmer im Nord
en."
3340 GOTO2590
3350 DATA 17,-2,101,-2,0,0,0
3360 PRINT"Hier ist es zu eng, Du kommst mit dem
"
3370 PRINT"Schatz nicht durch.":PRINT
3380 GOTO2590
3390 DATA 18,-2,101,8,0,0,0
3400 PRINT"Es sieht so aus als ob Du die Höhle"
3410 PRINT"nie findest."
3420 GOTO2590
3430 DATA 19,224,2,2,14,1,42
3440 PRINT"Du bist oben angekommen."
3450 PRINT"Unter Dir steht an der Wand:"
3460 PRINT"      Bilbo war hier'"
3470 GOTO2590
3480 DATA 20,226,1,2,2,25,2
3490 PRINT"Du bist auf der nördlichen Seite eine
r"
3500 PRINT"Spalte die zu breit ist zum Übersprin
gen."
3510 PRINT"Hallende Echos von unten deuten auf e
inen"
3520 PRINT"endlosen Abgrund hin."
3530 GOTO2590
3540 DATA 21,1,226,2,2,38,25
3550 PRINT"Du bist in Xanadu. Unter fließt Alph,
"
3560 PRINT"der heilige Fluß durch Höhlen";
3570 PRINT"unermeßlicher"
3580 PRINT"Weite in ein Meer ohne Sonne."
3590 GOTO2590
3600 DATA 22,-2,33,13,50,29,30
3610 GOTO2590
3620 DATA 23,2,1,2,31,2,2
3630 PRINT"Du bist auf dem Sims über dem Verlies
"
3640 PRINT"mit der Guillotine."
3650 GOTO2590
3660 DATA 24,-2,101,19,0,0,0
3670 PRINT"Da hört man den Riesen! Bloß zurück!"
:PRINT
3680 GOTO2590
3690 DATA 25,21,20,2,2,1,19
3700 PRINT"Du bist in der Höhle des Riesen. Sieh
zu,"
3710 PRINT"daß Du nicht hier bist, wenn er kommt
!"
3720 GOTO2590
3730 DATA 26,-2,65,-2,50,11,14
3740 PRINT"Du bist in einem verbotenen Gebiet. H
ier"
3750 PRINT"wird dieses Spiel gebraut. Besucher s
ind"
3760 PRINT"nicht zugelassen. Also geh!":PRINT
3770 GOTO2590
3780 DATA 27,2,40,2,2,21,20
3790 PRINT"Du bist im Kristallpalast. Die Wände"
3800 PRINT"erklingen in erschauernder Musik."
3810 GOTO2590
3820 DATA 28,-2,60,221,50,14,19
3830 GOTO2590
3840 DATA 29,2,42,2,13,1,1
3850 PRINT"Du bist am Gipfel eines gigantischen"
3860 PRINT"Stalagtiten. Du könntest runterrutsch
en,"
3870 PRINT"käme aber nicht wieder rauf."
3880 GOTO2590
3890 DATA 30,34,34,2,1,4,2
3900 PRINT"Du bist in einer kleinen Grotte. ";
3910 PRINT"Hier liegt ein":PRINT"Buch mit dem Ti
tel:

```

```

3920 PRINT"JANE'S FIGHTING SHIPS' aus dem";
3930 PRINT" Jahre 1763."
3940 GOTO2590
3950 DATA 31,14,14,23,2,1,2
3960 PRINT"Du bist in Guillontinen-Verlies. Ein"
3970 PRINT"spitziger Felsen wackelt auf dem Sims
"
3980 PRINT"über Dir."
3990 GOTO2590
4000 DATA 32,-2,101,516,0,0,0
4010 PRINT"Du rutschst ab! Die Felsen runter und
"
4020 PRINT"Du kannst Dich nicht festhalten!":PRI
NT
4030 GOTO2590
4040 DATA 33,2,1,2,1,116,3
4050 PRINT"Der enge Tunnel geht um die Ecke."
4060 GOTO2590
4070 DATA 34,1,35,2,1,30,30
4080 PRINT"Du bist im kleinen gewundenen";
4090 PRINT"Irrgarten."
4100 GOTO2590
4110 DATA 35,2,1,2,37,34,36
4120 PRINT"Du bist im kleinen trickreichen"
4130 PRINT"Irrgarten."
4140 GOTO2590
4150 DATA 36,35,2,1,37,34,12
4160 PRINT"Du bist im gewundenen kleinen";
4170 PRINT"Irrgarten."
4180 GOTO2590
4190 DATA 37,2,1,35,2,13,2
4200 PRINT"Du bist im trickreichen kleinen"
4210 PRINT"Irrgarten."
4220 GOTO2590
4230 DATA 38,2,21,2,116,1,2
4240 PRINT"Du bist in einer Höhle, die offenbar
von"
4250 PRINT"Neandertalern bewohnt wurde (wird?)."
4260 PRINT"An der Wand sind Öckerbilder von Wise
nts."
4270 PRINT"Der Boden ist mit Knochen bedeckt, ab
er"
4280 PRINT"ein kleiner Schacht ist im Boden."
4290 GOTO2590
4300 DATA 39,2,40,2,32,21,26
4310 PRINT"Du bist in einem Schwarzen Loch. Die"
4320 PRINT"Schwerkraft ist übermächtig."
4330 GOTO2590
4340 DATA 40,40,40,2,2,40,41
4350 PRINT"Du bist im Labyrinth."
4360 GOTO2590
4370 DATA 41,40,40,40,2,40,39
4380 PRINT"Du bist im Labyrinth. Hier ist's sehr
";
4390 PRINT"dunkel."
4400 GOTO2590
4410 DATA 42,28,28,28,28,28,28
4420 PRINT"Du bist im Ashram. Es riecht stark na
ch"
4430 PRINT"Gewürzen und Weihrauch, und alle";
4440 PRINT"Richtungen"
4450 PRINT"scheinen gleich."
4460 GOTO2590
4470 END
OK

```



# kleinanzeigen

Jeder Abonnent kann kostenlose Kleinanzeigen bis 40 Wörter aufgeben!

VERKAUFE NASBUG T4 in Eproms DM 30,--  
 NASSYS in Eproms DM 40,--  
 Spielprogramme 8K BASIC, Spielbeschreibungen und Listings komplett DM 20,--  
 Walter Schwinn  
 [Redacted]

4118 RAM, 8 Stück für NASCOM 2 DM100,--  
 zu verkaufen  
 Wiedemann  
 [Redacted], Tel. [Redacted]

VERKAUFE Drucker ADCOMP X80, 80 Z/Zeile,  
 100 Z/sec, RS 232/20mA Serienschmittst.  
 VB DM 1000,--  
 P.Keuntje Tel. [Redacted]  
 [Redacted]

VERKAUFE NASCOM 1 mit Tastaturegehäuse,  
 Veroframe, Netzteil, Motherboard, Buffer-Bus-B., Speicher 32K dyn., EPROM-Karte best.m. ZEAP2.0, NASDIS u. DEBUG, Eprommer, MDCR u. Drucker Heathkit H14  
 VB DM 2500,--  
 Tel. [Redacted] ab 16<sup>3</sup>h

VERKAUFE NASCOM 1 Supersystem  
 jetzt nur noch DM 1100,--  
 G.Mink Tel. [Redacted]

Liegt die fallende Anzahl der Kleinanzeigen daran, daß der NASCOM wieder attraktiver wird???

APPLE II Europlus 48K mit leerem Hobbyboard, Grafikdefinitionshilfe und Spielprogrammen DM 2800,--  
 L.Bayer Tel. [Redacted]  
 Dies war die letzte. Tschüß Herr Bayer!

VERKAUFE  
 ZEAP 2.0 in 4 x 2708 DM 130,--  
 Tel. [Redacted]

# INFO-ECKE

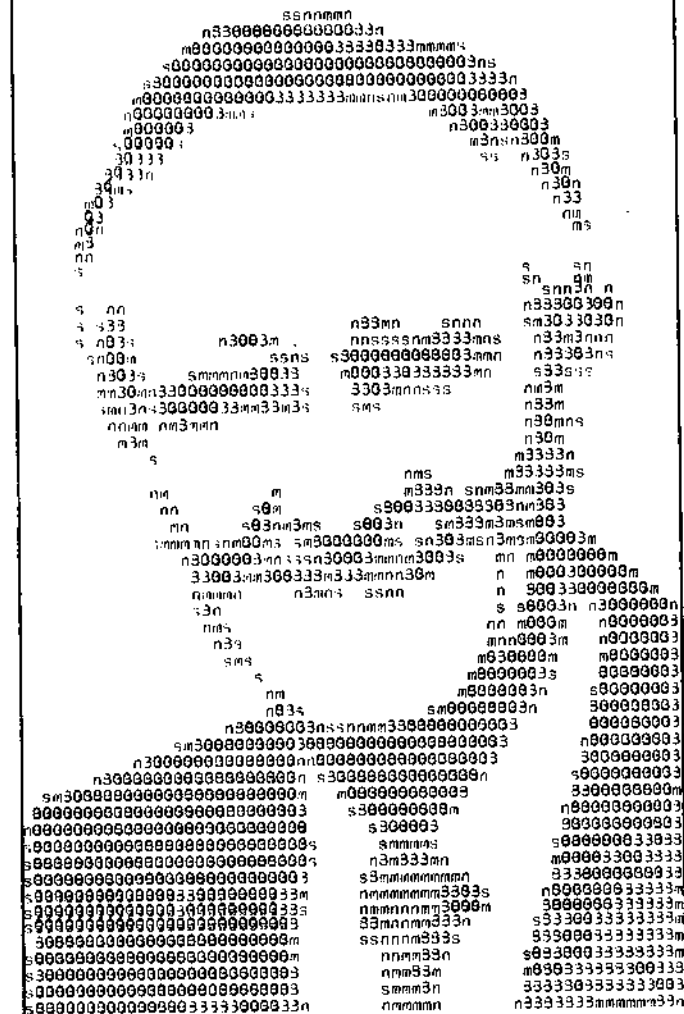
Wer macht was? In dieser Spalte sollen in Zukunft Informationen über laufende Projekte oder Programme, die noch nicht veröffentlicht wurden, vorgestellt werden. Informationsaustausch ist das Stichwort.

1. Gerade in Arbeit: Programmierbarer Charakter-Generator plus Grafik (wie N2) für NASCOM 1  
 Stefan Bürger Tel. [Redacted]

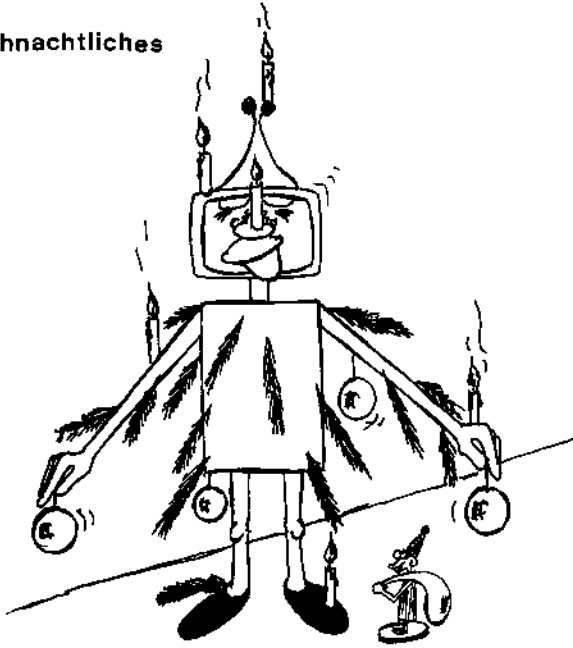
[Redacted] Tamm-Hohenstange  
 2. Geplant: Automatische Silbentrennung (Ab-laufplan vorhanden)  
 Stundenplan-Erstellung mögl. in Maschinenspr.

Günter Böhm (siehe Impressum)  
 3. Schon gelaufen: Meine neueste Errungenschaft ist eine Miniatur-Reflex-Lichtschranke, die ich an meinen A/D Wandler angeschlossen habe. Damit kann ich ein Papierbild, auf den Plotter montiert, punktweise abtasten und digital abspeichern. Über den Fernschreiber kann es wieder als Rasterbild ausgegeben werden.  
 Peter Bentz  
 [Redacted]

[Redacted] Mülheim/Ruhr



## Weihnachtliches



Hallo liebe Leser,  
und wieder naht die schönste Zeit des Jahres; die Weihnachtszeit. Mit glänzenden Augen harren die Kinder auf den feierlichen Augenblick, und auch die Erwachsenen können sich eines wehmütigen Gefühls nicht erwehren. Was aber macht unser Computer, während wir uns um den reichgedeckten Gabentisch versammeln? Wer denkt an ihn, wenn wir verklärt den Lichterglanz des Weihnachtsbaumes und den Duft vertrockneter Lebkuchen in uns aufnehmen?

Haben Sie ein Herz, und lassen Sie Ihren NASCOM am Lichterfeste teilhaben! Sicher findet sich ein Plätzchen am Kamin, wo er sich gemütlich zusammenkauern kann. Welche Freude ist es auch für die Kinder, wenn man ein paar Kerzen auf der Tastatur anbringt. (Die Wachsflecken kann man leicht zwischen den Jahren wieder entfernen). Die Frau des Hauses ist auch sicher gerne bereit, den Monitor mit einigen Tannenzweigen zu schmücken. Besonders hübsch sehen bunte Glaskugeln im Veroframe aus; sie werden nur noch übertroffen durch die würzigen Zimtsterne, die man zwischen die Mechanik des Fernschreibers klemmen kann. Am besten sind die Leute mit einer Floppy dran; auch Gäste bewundern die hellen Kerzenlichter, die sich bei abgenommenem Gehäuse lustig drehen. So einfach also ist es, sich und seinem stummen Freund, dem NASCOM, eine Freude zu bereiten.

In diesem Sinne ein frohes Fest  
wünscht Ihr NASCOMPL

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

RESSORTS:

MASCHINENPROGRAMME:

Günter Böhm, [REDACTED]

[REDACTED] Karlsruhe, Tel. [REDACTED]

Günter Kreidl, [REDACTED], [REDACTED] Straelen  
Tel. [REDACTED]

BASIC und FLOPPY:

Wolfgang Mayer-Gürr, [REDACTED]

[REDACTED], [REDACTED], 4350 Reckling-  
hausen, Tel. [REDACTED]

HARDWARE:

Josef Zeller, [REDACTED], [REDACTED] Neu-  
Ulm

VERLAG: NSCOM JOURNAL, c/o MK-Systemtechnik,  
Pater-Mayer-Str.6, 6728 Germersheim

Tel. 07274/2756

Telex 453500 mks d

VERTRIEB: Direktvertrieb durch den Verlag

Erscheinungsweise: monatlich

Bezugspreis: IM In- und Ausland 48,- für ein  
Jahresabonnement. Abonnements können aus  
technischen Gründen immer nur für die Dauer  
eines Kalenderjahres, d.h. vom 1.1. bis  
31.12. laufen. Bei Bestellung nach dem 1.1.  
werden die fehlenden Hefte mit der ersten  
Lieferung bis zum Bestellzeitpunkt automa-  
tisch mitgeliefert.

Bezugsmöglichkeiten: Durch Bestellung bei  
MK-Systemtechnik (beigefügte Bestellkarte).

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

Fa. Michael Klein, Sonderkonto 29926-674  
beim Postscheckamt Ludwigshafen.

Zahlung: Nach Eingang Ihrer Bestellung er-  
halten Sie von uns die ausstehenden Hefte  
bis zur aktuellen Ausgabe sowie eine Rech-  
nung. Bitte, zahlen Sie dann den Rechnungs-  
betrag auf unser Sonderkonto (s.o.) ein.  
Bitte keine Vorauszahlungen!

Bitte, Anfragen wegen Abonnements oder Lie-  
ferung nicht an die Redaktion sondern nur an  
den Verlag.

Die Autoren tragen die Verantwortung für  
ihre Beiträge selbst.

Für Fehler in Text, Bildern und sonstigen  
Angaben kann keine Haftung übernommen werden

# MK-SYSTEMTECHNIK

**Jetzt auch in Solingen!**



## EUROCOM-2 1670,-

Einplatinencomputer im Doppel-Europa-Format. Sehr leistungsfähige Graphik.

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

### Zubehör für EUROCOM-2

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

### Software f. EUROCOM-2

BASIC	220,-
Assembler/Editor	220,-
DEBUG	179,-
Disassembler	113,-
PASCAL	350,-
FORTH	220,-
wahlweise auf Audiotassette oder Mini-Digitalcassette	



**MIKOS 1**, »Steckdosentartiges« Komplettergät auf Basis d. EUROCOM II

(inkl. Tastatur und Betriebssystem

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



## ITT 2020

### Unser PASCAL-System:

Enthält: Wahlweise Apple II oder ITT 2020, 64k RAM (Hardware für PASCAL-Language System), 12" 18 Mhz-Monitor grün, 2 Stück 5,25" Floppy-Disk-Laufwerke mit Controller, **plotfähiger Drucker EPSON MX 82 FT**, UCSD-PASCAL-Software. Komplett mit allen Handbüchern und Verbindungskabeln

9985,-

### Unser Farb-System:

Enthält: ITT 2020 mit PAL-Ausgang, dadurch besonders gutes Farbbild, 48k RAM, 14" SANYO echter Farbmonitor, mit Grünschlalter für Computertextdarstellung, auch als vollwertiger 8-Kanal-Farbfernseher zu verwenden, 2 Stück 5,25" Floppy-Disk-Laufwerke mit Controller, plotfähiger Drucker EPSON MX 82 F/T, BASIC-Lehrgang auf Diskette. Komplett mit allen Handbüchern und Verbindungskabeln

8885,-

### Unser Grafik-System:

Enthält: Apple II 48k RAM, 5,25" Floppy-Laufwerk mit Controller, Apple-Graphics-Tablet, plotfähiger Silentype-Drucker, passend zum Graphics-Tablet. Komplett mit allen Handbüchern und Verbindungskabeln

7985,-

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

### Erweitern Sie Ihren Apple II / ITT 2020:

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

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

### WATANABE Plotter

3365,-



An alle Micro-computer mit Parallelschnittstelle anschließbar. DIN A/3

Zubehör: Interface u. Kabel IEEE 488 449,-

Interface u. Kabel für Apple und ITT 2020 395,-

Interface u. Kabel RS 232 C 849,-

**Neu!** Plotbibliothek in UCSD-PASCAL 895,-

Plotsoftware, dialogorientiert 499,-

WATANABE WX 4675 4945,-

Intelligenter 6-Farben-Plotter DIN A/3

### Endlich lieferbar!

#### MX 82 F/T

mit einem Interface n. Wahl\* 2595,-

oh. Int. face (8 Bit Parallel-Eing.) 2325,-

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

#### MX 80 F/T

o. Interf. (8 Bit Parallel-Eing.) 1625,-

m. einem Interface n. Wahl 1895,-

\*Interfacekarten für alle gängigen Rechensysteme lieferbar: PET/CBM, TRS 80, MZ 80 K, IEEE 488 (HP), HEALTH-Computer, NASCOM, COMPU-CORP oder RS 232 C (V24).

# NASCOM

Liebe Leser,

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

### NASCOM 1

Einplatinencomputer für »stand-alone« und OEM-Anwendungen

2 80 CPU - 2k statisches RAM - Neues 2k NAS-SYS 3 Betriebssystem - Hochwertiges Keyboard mit 58 Magnetknoten - Video-Interface mit 16 x 48 Zeichen - Groß/Kleinschreibung mit Unterlängen - 128 ASCII-Zeichen - BAS-Ausgang - UHF-Ausgang - RS 232 C und 20mA Serien-Schnittstelle - Cassette-Interface - Mini-DCRs anschließbar - 16 Ein/Ausgabeleitungen (PIO) - 2 Vektorinterrupts - Erweiterbar auf 256k RAM/ROM

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

Für OEMs auch ohne Keyboard und in Sonderbestückungen lieferbar.

### NASCOM II

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

2 80 CPU 4 Mhz Taktfrequenz - Bis zu 8k RAM (4118) oder EPROM auf der Grundplatte - 8k BASIC in einem 8k x 8 organis. ROM Typ 36000 (MOSTEK) - Voll gepufferter BUS

**NASCOM II Bausatz, 8k stat. RAM, BASIC, 2k Monitor 1665,-**

**NASCOM II Bausatz, 16k dyn. RAM, BASIC, 2k Monitor 1995,-**

Fertigergeräte NASCOM II bitte anfragen.

# Lucas Logic

**Neu!**

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

### Floppy Disk

1749,-

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

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

### Schach

128,-

Schachgrafik ROM für NASCOM 2 98,-

Grafik-Zusatzkarte f. NASCOM 1 198,-

### NASCOM-JOURNAL

Zeitschrift für den ernsthaften NASCOM-Anwender. Zahlreiche Programmier-Beispiele, Anregungen zur Hardware etc. Im NASCOM-JOURNAL stellen wir auch laufend neue Produkte vor!

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



NASCOM 1

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

## MK-Systemtechnik

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

## MK-Systemtechnik

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

## MK-Systemtechnik

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

