

nascocom journal

Zeitschrift für Anwender des NASCOM 1 oder NASCOM 2

2. Jahrgang · Juli 1981 · Ausgabe 7

Herausgeber:

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

MK-Systemtechnik Thomas Gräfenecker · Kriegsstraße 164 · 7500 Karlsruhe

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

Inhalt:

- | | | |
|----|--|------------------|
| 2 | Editorial | M Klein |
| 3 | NASCOM JOURNAL INTERN | Red. |
| 4 | CLD EXTENDED BASIC | W. Mayer-Gürr |
| 5 | CLDDOS System Unterprogramme | C. Baier |
| 8 | SOUNDGENERATOR Teil 2 | G. Böhm |
| 11 | YATZI (Würfelspiel) | P. Waltenberger |
| 12 | Speichererweiterung auf 48 K | G. Endert |
| 13 | Serie: Sortieren in BASIC | W. Mayer-Gürr |
| 14 | MDCR-MONITOR Teil 1 | J. C. Lotter |
| 17 | NASCOMPL / Impressum | Red. |
| 18 | UFO-Jagd Relocator-Test
Kleinanzeigen | H. Kögler / Red. |
| 19 | Leseranfragen | Red. |
| 20 | Günstige Angebote | |

Editorial

Liebe Leser,

nach Monaten der Gerüchteküche, der Geheimtips und Halbwahrheiten, ist es inzwischen raus: NASCOM Microcomputers ist tot — es lebe LUCAS LOGIC Ltd.

Der arg strapazierte NASCOM-Anwender hat jetzt endlich Ruhe. Die englische Firmengruppe LUCAS, Jahresumsatz weltweit ca. 1 300 000 000 Pfund Sterling, hat die Rechte am NASCOM-System, die Platlinenvorlagen etc. gekauft und produziert das System jetzt in der eigens dafür gegründeten Tochterfirma, die **LUCAS LOGIC Ltd.** heißt.

Die jüngste Tochter der Familie ist schon ganz schön munter. Teile für ca. 5 000 NASCOM 2-Systeme wurden eingekauft und etliche Systeme bereits zusammengestellt. Auch der NASCOM 2 ist inzwischen als Fertiggerät verfügbar. Für alle, die nicht so gerne basteln. Ganz nebenbei hat LUCAS LOGIC ca. 50 Mitarbeiter eingestellt — man hat ja einiges vor.

Neben den Vorgängen im Ausland hat sich natürlich auch im Inland was getan. Als Generalimporteur für die Bundesrepublik hat MK-SYSTEMTECHNIK inzwischen den Alleinvertrieb von NASCOM-Produkten hier übernommen. Wir beginnen jetzt, das Händlernetz aufzubauen.

Wir werden uns die Händler sehr sorgfältig aussuchen. Die Firma N.A.S. GmbH, München, die vorher Alleinimporteur war, wird nicht zu den autorisierten Händlern zählen.

LUCAS LOGIC beginnt zunächst damit, die Produkte, die bereits existieren weiter zu produzieren, um den unmittelbaren Nachholbedarf zu decken. Es wird bis Anfang September noch geringfügige Lieferschwierigkeiten geben. Dann dürfte unser Zentrallager jedoch soweit gefüllt sein, daß wir eingehende Bestellungen mit Gelassenheit erwarten können.

Einige Kunden haben die neue Entwicklung schon sehr begrüßt. Wir sind sicher, daß es uns gelingen wird, den Wünschen unserer Kunden in allen Punkten gerecht zu werden. Vielleicht können gerade Sie als besonders engagierte Leser mit etwas Manöverkritik dazu beitragen, daß wir uns Ihren Wünschen und Vorstellungen gut anpassen.

Auf gute Zusammenarbeit — Ihr



nascom journal

intern

Liebe Leser,

seit der letzten Ausgabe des Journals ist die Redaktion erweitert worden, um noch besser Ihren Wünschen nachkommen zu können. Ich möchte mich in Zukunft um die Themen Basic und Floppydisk-System kümmern. Zu meiner Person ganz kurz: ich bin Jahrgang 48 - habe Chemie studiert, bin dann zur Schule gekommen - unterrichte außer Chemie neuerdings auch Informatik; so ist das Hobby auch teilweise zum Beruf geworden. Mit dem Nascom arbeite ich seit 3 Jahren.

Heute will ich Ihnen einen neuen Service vorstellen. Das Diskettensystem hat u.a. den Vorteil, daß man schnell eine größere Anzahl von Kopien bestehender Programme erstellen kann, die auch auf anderen Anlagen sicher wieder eingelesen werden können - bei Cassetten ist das ja leider nicht immer der Fall. Wir möchten deshalb unsere Leser, die eine CLD-Floppy besitzen, Ihre großen Erfahrungen und Programme gegenseitig zu nutzen.

Wie soll das ablaufen?

1. - Mir eine (mit möglichst vielen Programmen bespielte) Diskette schicken.
Anschrift siehe Impressum
2. - Wünsche äußern (z.Zt. sind Programme in folgendem Umfang vorhanden:
 - 200 kByte in CLD-Basic
 - 100 kByte in MBasic
 - 20 kByte in Maschinensprache
 - 150 kByte in Assemblercode)
3. - bitte angeben, wie groß der Speicherplatz ist und ob Graphik vorhanden ist
4. - 5 DM für Porto, Verpackung und Dokumentation beilegen.

Ich schicke dann eine volle Diskette mit Dokumentation zurück.

W.Mayer-Gürr

In der Zeitschrift MC nr. 2 ist ein sehr komfortabler Texteditor für Z80 Systeme abgedruckt. Die Anpassung an NASSYS verlangt eine kleine Eingaberoutine, die im folgenden als Hexdump wiedergegeben ist. Der gesamte Bildschirmspeicher dient dann als Eingabepuffer; d.h. man kann beliebig lange Zeilen eingeben, also auch beliebig oft über den Bildschirmrand hinaus schreiben. Da der Texteditor die Kontrollzeichen des NASCOM an den Bildschirmrändern ignoriert, werden die Eingaben korrekt in den Textspeicher übernommen.

Das Listing ist ohne Adressen angegeben, da das Programm an jeder Stelle lauffähig ist.

In den Workspace wird einfach bei W+A ein Sprung zu dieser Routine eingetragen, bei W+D ein Sprung nach 0030.

Wer sich übrigens die Tipperei (fast 10 000 Eingaben !) sparen will, der soll mir eine Cassette und eine frankierte und adressierte Versandtasche schicken, und ich kopiere ihm gerne das Programm. Der Verfasser des Texteditors verbietet zwar ausdrücklich eine gewerbliche Vervielfältigung, aber er wird wohl nichts gegen eine gegenseitige Hilfe der Leser des NASCOM Journals einzuwenden haben. Bitte geben Sie das gewünschte Ladeformat an. (W oder D Befehl)

```
E5 C5 DF 7B FE 0D 28 03
F7 18 F7 ED 5B 29 0C F7
2A 29 0C F5 AF ED 52 30
08 EB 11 40 00 AF ED 52
EB F1 C1 E1 C9
```

Das Programm weist übrigens einen kleinen Fehler auf: der Befehl D\$, angewandt auf eine Zeile, löscht auch in der nächsten Zeile d- Zeichen ab der Position p1. Wer findet den Fehler?

An dieser Stelle sollten eigentlich noch einige andere Anregungen und Projekte folgen, die aber aus organisatorischen Gründen in das nächste Heft verschoben werden müssen. Eines sei aber schon vorweggenommen: ich möchte die Leser des Journals einladen, gemeinsam einen FORTH -Interpreter und Compiler zu entwickeln. Mehr darüber im nächsten Heft.

Wer so etwas schon einmal versucht hat, kann sich ja mit mir in Verbindung setzen.

Günter Kreidl

CLD Extended Basic

W. Mayer-Gürr

Dieses komfortable Basic ist auf Cassette oder in Eproms gebrannt erhältlich und belegt die Adressen 1000H bis 3FFFH. Das Basic ist eigentlich nur 11K lang, der Rest ist für eigene Ergänzungen reserviert. Hier der Befehlssatz (ohne Parameter):

ABS	AND	ASC	ATN
BUILD	CHR\$	CLEAR	CNTRL
CONTINUE	COS	DATA	DEF
DELETE	DIM	DUMP	EDIT
END	EXP	FDUMP	FLOAD
FOR NEXT	FREE	GET	GOSUB
GOTO	IF THEN	INPUT	LINE INPUT
INT	LEN	LEFT\$	LET
LIST	LOAD	LOCK	LOG
MAX	MID\$	MIN	NOT
OLDLOAD	ON GOTO	OR	OUT
PAUSE	PEEK	PIN	POKE
PORT	POS	PUT	PRINT
READ	REM	RESTORE	RETURN
RIGHT\$	RND	RUN	SCRATCH
SGN	SIN	SPC	SQR
STR\$	STEP	STOP	TAB
TAN	UNLOCK	USR	VAL
VERIFY			

Auf einige Besonderheiten will ich in diesem Artikel eingehen. Das Basic verwendet einen eigenen Monitor, ist also vom T2, T4 oder NASSYS unabhängig. Die notwendigen Hardwareänderungen beschränken sich auf den Einbau von 2 zusätzlichen Tasten (Control und Interrupt), 1 LED (zeigt Eingabebereitschaft an) und einem Lautsprecher (für ASCII 7 = Bell). Zur Ansteuerung werden die freien Bits von Port 0 verwendet.

Bei der Eingabe erfolgt eine Befehlsvervollständigung. Man braucht also für ein Schlüsselwort nur 1-3 Buchstaben eingeben, der Rest wird ergänzt. Mit BUILD erzeugt das Programm automatisch Zeilennummern in einem frei zu wählenden Abstand.

CNTRL

Mit CNTRL 0 kann man ein interruptähnliches Verhalten nachbilden. Nach Drücken der Interrupttaste und Control-B springt das Programm zu einem Unterprogramm, arbeitet dieses ab und fährt dann an der Unterbrechungsstelle im Hauptprogramm fort. Mit CNTRL 1 und 2 lassen sich Ausgabeformate ändern, CNTRL 3 und 4 rufen eigene Maschinenspracheunterprogramme auf (ohne Übergabe von Parametern).

DUMP

Das Basic verwendet auch eigene Cassettenroutinen. Ein Programm wird mit einem Filenamem versehen, dieser wird beim Laden überprüft, andere werden dann ignoriert. Durch VERIFY kann das gerade abgespeicherte Programm auf korrekte Aufzeichnung überprüft werden, ohne daß der Originalinhalt im RAM bei einem Fehler zerstört wird. Dies ist ein Vorteil, den man bei Cassettenbetrieb nicht hoch genug schätzen kann. Mit FDUMP werden außer dem Programm auch die Werte der Variablen abgespeichert, mit PUT nur die Variablen.

EDIT

Sehr einfach lassen sich durch diesen Befehl Programmzeilen korrigieren.

MAX,MIN

finden aus einer beliebig großen Variablengruppe den größten bzw. kleinsten Wert.

PAUSE

hält das Programm für eine vorgegebene Zeit an

PORT

Hier läßt sich die Datenein- und ausgabe auf einen anderen Port legen, z.B. zum Betrieb eines Druckers. Arrays (bis zu 8-dimensional) müssen alle durch DIM zum Programmumfang dimensioniert werden; manche Basics verlangen dies erst ab einem Inhalt größer 10. Bei der Übernahme von Programmen anderer Basic-Dialekte ist auf die Verwendung der Booleschen Vergleiche zu achten. Die meisten Basics setzen bei einem Vergleich FALSE = -1, das CLD-Basic aber nimmt FALSE = -65535. Bei einem Input darf keine Stringvariable erfragt werden, dafür gibt es das LINE INPUT, bei dem auch Kommas, Leerstellen und andere Zeichen eingegeben werden können. Graphik kann nur über PEEK und POKE verwirklicht werden, da intern Bit 7 für andere Zwecke verwandt wird.

LEERKASSETTEN



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

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

Bei: M K - Systemtechnik
Waldstraße 20
6728 Germersheim/Rhein
Tel.: 07274/2756

CLDDOS-SYSTEM

Nützliche Unterprogramme von Gerhard Baier

Beim Programmieren in Maschinensprache gibt es einige Routinen, die immer wieder benötigt werden. Dazu zählen Routinen zur Ein- und Ausgabe von Ziffern, Bytes und Worten, Routinen zur Datenkonvertierung, Stringbehandlung und Fileverwaltung, Routinen, die spezielle Devices ansprechen, usw. ...

Im Prinzip sind derartige Routinen bereits in der Grund-Software eines Computer-Systems implementiert. Inwieweit aber dann auch ein Benutzer auf diese Routinen zurückgreifen kann, hängt von der Qualität der Software und vom Umfang der Dokumentation ab. Diese ist aber oft sehr dürftig und unvollständig. Als Benutzer ist man dann gezwungen, sich eine eigene Software-Bibliothek mit nützlichen Unterprogrammen anzulegen.

So ist es nun auch beim NASCOM CLDDOS-System. Es gibt zwar eine ganze Reihe von Unterprogrammen, die mehr oder minder gut dokumentiert sind (CLDDOS.ACM und ROMSUBS.ACM). Allerdings kann man mit diesen Subroutines noch nicht viel anfangen. Zum effektiven Programmieren fehlen meiner Ansicht nach noch einige übergeordnete Routinen, die sicher irgendwo in der CLDDOS-Software schon vorhanden sind. Leider gibt es darüber aber keine Unterlagen.

Ich habe mir deshalb eine Reihe von Unterprogrammen geschrieben, die ich immer wieder brauche und die wahrscheinlich auch für andere NASCOM-Floppy-Besitzer interessant sind. In dieser Ausgabe möchte ich eine HEX-Eingabe-, eine String-Eingabe- und eine ASCII-HEX-Konvertierungs-Routine für das CLDDOS-System abdrucken.

Weitere Routinen folgen in den nächsten NASCOM-Journal Ausgaben.

#CNVAB

VERSION 1.0
20-NOV-80

DIE SUBROUTINE #CNVAB KONVERTIERT
EINE HEX-ZIFFER VOM ASCII- INS
BINAER-FORMAT

EINGABE-PARAMETER:

- A - ZIFFER IM ASCII-FORMAT

AUSGABE-PARAMETER:

- A - ZIFFER IM BINAER-FORMAT

-- CARRY -- 0 : GUELTIGE ZIFFER IN A

1 : KEINE HEX-ZIFFER IN A

VERWENDETE REGISTER: AF

```
0000      D6 30          #CNVAB  SUB      30H
0002      FE 00          CMP      00H
0004      DB           RET      C           ; KEINE HEX-ZIFFER IN A
0005      FE 0A          CMP      0AH
0007      38 07          JR      C, #CNVAB1
0009      FE 11          CMP      11H
000B      DB           RET      C
000C      D6 07          SUB      07H
000E      FE 10          CMP      10H
0010      3F           #CNVAB1  CCF           ; CLEAR CARRY
0011      C9           RET
```

\$INHEX

Version 1.0

26-Apr-81

Die Subroutine \$INHEX gibt zunächst einen Prompt-String aus und liest dann 1 bis 4 Hex-Ziffern von der Tastatur ein. Diese werden rechtsbündig im HL- Register abgelegt.

0000	C5	*INHEX	PUSH	BC	
0001	3C		INC	A	
0002	4F		LD	C,A	
0003	D5		PUSH	DE	
0004	E5	*INHEX1	PUSH	HL	
0005	E5		PUSH	HL	
0006	C5		PUSH	BC	
0007	FF 07		SCALL	.CLRCD	;RINGBUFFER LOESCHEN
0009	C1		POP	BC	
000A	E1		POP	HL	;PROMPTSTRING-ADRESSE
000B	FF 03		SCALL	.PRINT	;PROMPTSTRING AUSGEBEN
000D	21 00 00		LD	HL,0000H	
0010	41		LD	B,C	
0011	FF 01	*INHEX2	SCALL	.SCIN	;EINGABEBUFFER ABFRAGEN
0013	38 FC		JR	C,*INHEX2	
0015	FE 0A		CMP	0AH	;CR ?
0017	28 1D		JR	Z,*INHEX5	
0019	CD 00 00		CALL	*CNVAB	;ZEICHEN UMWANDELN
001C	38 11		JR	C,*INHEX4	;ZEICHEN GUELTIG ?
001E	C5		PUSH	BC	
001F	06 04		LD	B,04H	;SHIFT A,HL UM 1 DIGIT
0021	07		RLCA		
0022	07		RLCA		
0023	07		RLCA		
0024	07		RLCA		
0025	07	*INHEX3	RLCA		
0026	CB 15		RL	L	
0028	CB 14		RL	H	
002A	10 F9		DJNZ	*INHEX3	
002C	C1		POP	BC	
002D	10 E2		DJNZ	*INHEX2	
002F	3E 07	*INHEX4	LD	A,07H	;BELL-CODE
0031	FF 02		SCALL	.SCOUT	;FEHLER MELDEN
0033	E1		POP	HL	
0034	18 CE		JR	*INHEX1	;EINGABE WIEDERHOLEN
0036	7D	*INHEX5	LD	A,L	
0037	D1		POP	DE	
0038	D1		POP	DE	
0039	C1		POP	BC	
003A	C9		RET		

Eingabe-Parameter:

-A- Max. Anzahl der zu lesenden HEX-Ziffern

-HL- Adresse des Prompt-Strings

Ausgabe-Parameter:

-A- Die zwei niederwertigsten HEX-Ziffern

-HL- 1 bis 4 stellige HEX-Zahl

Verwendete Register: AF, HL

\$INSTR

Version 1.0

26-Apr-81

Die Subroutine \$INSTR gibt zunächst einen Promptstring aus und liest dann einen ASCII String vom Terminal-Eingabebuffer ein. Die maximale Länge des Strings wird durch das Register A vorgegeben. Werden mehr Zeichen

eingegeben, so wird die Eingabe-Aufforderung wiederholt.

Eingabe- Parameter:

-A- Max. Anzahl der einzulesenden Zeichen

-DE- Adresse des Prompt-Strings

-HL- Adresse, ab der die eingegebenen Zeichen abgelegt werden sollen.

Die Länge dieses Feldes soll um 1 größer

sein als die max. Zeichenzahl.

Ausgabe-Parameter:

-A- Anzahl der tatsächlich eingelesenen Zeichen

-DE- Adresse des Prompt-Strings

-HL- Adresse, ab der die eingelesenen Zeichen abgelegt wurden (Ende: "00")

Verwendete Register: AF, DE, HL

```
0000    C5          #INSTR PUSH    BC
0001    3C          INC     A
0002    4F          LD     C,A
0003    D5          #INSTR1 PUSH   DE
0004    E5          PUSH   HL
0005    E5          PUSH   HL
0006    D5          PUSH   DE
0007    FF 07      SCALL  .CLRCD ; RINGBUFFER LOESCHEN
0009    E1          POP    HL ; PROMPT-STRING ADRESSE
000A    FF 03      SCALL  .PRINT ; PROMT-STRING AUSGEBEN
000C    E1          POP    HL
000D    41          LD     B,C
000E    FF 01      #INSTR2 SCALL  .SCIN ; EINGABEBUFFER ABFRAGEN
0010    3B FC      JR     C,#INSTR2
0012    36 00      LD     (HL),00H ; STRINGENDE-ZEICHEN
0014    FE 0A      CMP    0AH ; CR ?
0016    2B 0C      JR     Z,#INSTR4
0018    77          LD     (HL),A ; ZEICHEN INS MEMORY
0019    23          INC    HL
001A    10 F2      DJNZ  #INSTR2
001C    3E 07      #INSTR3 LD     A,07H ; BELL-CODE
001E    FF 02      SCALL  .SCOUT ; FEHLER ANZEIGEN
0020    E1          POP    HL
0021    D1          POP    DE
0022    1B DF      JR     #INSTR1
0024    79          #INSTR4 LD     A,C
0025    90          SUB    B
0026    E1          POP    HL
0027    D1          POP    DE
0028    C1          POP    BC
0029    C9          RET
```

Anmerkungen der Redaktion zur obigen Serie

Das Floppy-Betriebssystem wird beim Booten immer an das Ende des vorhandenen RAMs gelegt und hat demnach anlagenspezifische Adressen. Lediglich einige Adressen (die im Handbuch dokumentiert sind) werden am Anfang des RAMs gesetzt. Diese lassen sich als Pseudobefehle beim Assemblieren über ein sogenanntes ACM-File nutzen. Was Herr Baier beschreibt, könnte der Anfang eines zusätzlichen Hilfs-ACM-Files sein - auch seine Programme können dann als Pseudolabels verwendet werden. Dies ist sinnvoller, als in das

Betriebssystem einzusteigen. Einmal wegen der veränderlichen Adressen, außerdem werden spätere Änderungen des Betriebssystems keine Auswirkungen haben.

Vielleicht gibt es einige Leser, die an der gleichen Aufgabe arbeiten. Hier bestünde doch die Möglichkeit einer Zusammenarbeit.

Es wurde überhaupt der Vorschlag gemacht, die Möglichkeiten des NASCOM Journals zu nutzen, und gemeinsam komplexere Programme zu erstellen.

Schreiben Sie uns, wenn Sie diesbezüglich Ideen haben.

SOUNDGENERATOR

TEIL 2

von G. Böhm

Mit dem Programm im letzten Heft konnten Sie die Tonhöhen der 3 Tongeneratoren variieren und konnten erfahren, wie überhaupt Informationen an den PSG gelangen. Im folgenden möchte ich die Funktion der restlichen Register besprechen; bei ihrer Programmierung wird analog zu den Tonhöhenregistern verfahren.

Um die Register zu testen, sollten Sie zunächst das BASIC Programm aus dem letzten Heft laden, sodaß der Generator den bekannten Dreiklang mit wiederholter abfallender Hüllkurve produziert. Mit den aufgeführten Zusatzprogrammen können Sie dann die verschiedenen Register mit veränderten Werten füttern, um so die Klänge zu erzeugen, die Sie benötigen.

Register 13 : Hüllkurvenformen

Das Envelope-Register kann 8 verschiedene Hüllkurven erzeugen, die z.T. einen andauernd an- und abschwellenden Klang ergeben oder nur einmal erklingen. Bei letzterer Form ist es wichtig, die Hüllkurvenform als letzten Parameter zu programmieren, sonst kann es geschehen, daß die Hüllkurve abläuft, bevor die anderen Parameter geladen sind, und der Ton unhörbar bleibt.

Besser als jede Beschreibung der Formen ist wohl folgendes Schaubild, das die Registerdaten in Hex und Dez und die entsprechenden Kurven zeigt.

Mit dem kleinen BASIC Zusatzprogramm können Sie sich "unsere Dreiklang" mit verschiedenen Hüllkurven anhören. Wenn nach der Input-Aufforderung keine neuen Daten eingegeben und nur NL drücken, erscheint die vorherige Kurve nochmals. Dies ist eine Erleichterung beim Testen von "verklingenden" Hüllkurven; mit NL können Sie dabei den Anschlag eines Tasteninstrumentes imitieren.

zusatzprogramm zum test der hüllkurvenformen mit ay-3-8910

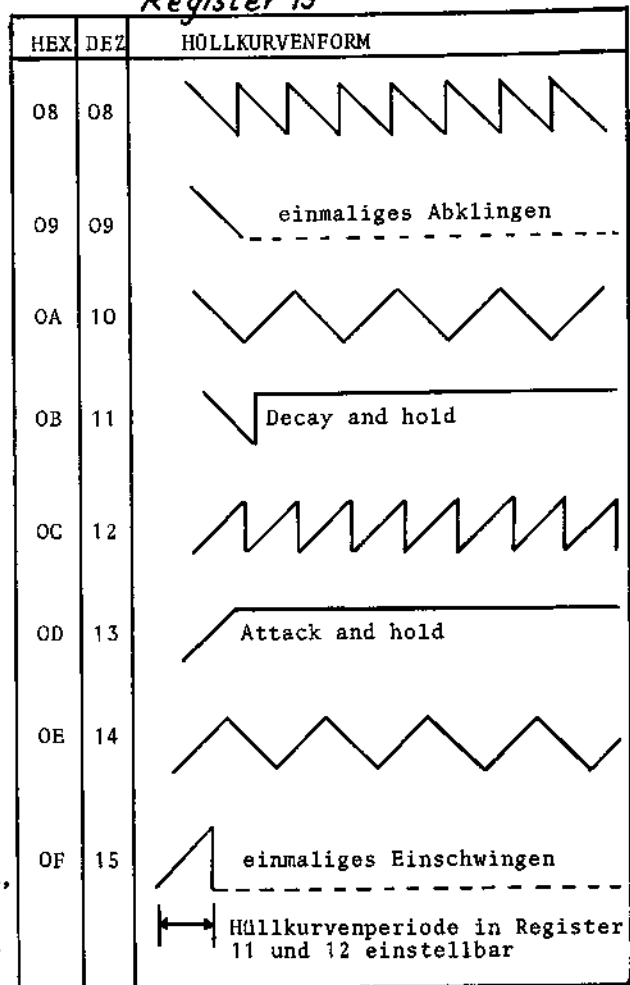
```
ok
list 300

300 input'hüllkurvenform';x
305 if x>15 then end
310 i=13
320 r(i)=x
330 gosub 200
340 goto 300
ok
```

Register 11 und 12 : Hüllkurvenperiode

In Register 12 wird die Grobstimmung, in Register 11 die Feinstimmung der Hüllkurvenperiode eingestellt. Meiner Meinung nach kommt es hierbei absolut nicht auf eine große Genauigkeit an, so daß man die Feinstimmung ver-

Register 13



nachlässigen kann. (D.h. das Register 11 bleibt immer auf 0).

Für die Grobstimmung werden 8 Bit verwendet, die Länge der Hüllkurve variiert also zwischen 0 und FF hex (bis 255 dezimal). Mit dem kleinen Zusatzprogramm können Sie brauchbare Werte testen. Bei der Eingabe eines größeren Wertes kehren Sie wieder auf die Kommandoebene zurück. (Dies gilt übrigens entsprechend für die anderen BASIC Zusatzprogramme.)

test der hüllkurvendauer
(nur fine tune/coarse tune bleibt0)
ok
list 390

```
390 rem fuer hüllkurvendauer
395 rem zelle 110 aendern=goto 400.
400 input'hüllkurvendauer';x
410 if x>255 then end
420 i=12
430 r(i)=x
440 gosub 200
450 goto 400
ok
```


Register 8, 9 und 10

Die Steuerung durch den Hüllkurvengenerator funktioniert allerdings nur, wenn die Amplitudenregister richtig programmiert sind. Hierbei sind Bit 0 bis Bit 3 für die statische Lautstärke verantwortlich, Bit 4 für die Steuerung durch den Hüllkurvengenerator.

Das bedeutet praktisch: Werte zwischen 0 und 15 dez. beeinflussen die Lautstärke der entsprechenden Kanäle direkt. So kann man z.B. einen Klang erzeugen, bei dem die Lautstärke des Grundtones und der beiden "Obertöne" verschieden ist, wie etwa bei natürlichen Instrumenten. Bei gesetztem Bit 4 wird die Lautstärke des jeweiligen Kanals nur von der Hüllkurve beeinflusst. Die Werte 16 bis 31 dez (10 bis 1F hex) ergeben also alle den gleichen, von der Hüllkurve gesteuerten Volumencharakter. Der Einfachheit halber sollte man sich also gleich für 16 (10) entscheiden.

Selbstverständlich kann auch ein Kanal statisch in der Lautstärke eingestellt werden, während man die anderen durch die Hüllkurve regeln läßt und umgekehrt.

Die Register sind den Analogausgängen folgendermaßen zugeordnet:

Register 8 Kanal a
Register 9 Kanal b
Register 10 Kanal c

Um die Amplitudenwerte zu testen, setzen Sie in das Testprogramm für die Hüllkurvenformen in Zeile 310 einfach I=8 (oder entsprechend 9 bzw. 10) ein.

Register 7 : ENABLE Register

Dieses Register dürfte wohl das wichtigste sein, denn es entscheidet darüber, was überhaupt an die Analogausgänge gelangt.

Dabei bedeutet jedes nicht gesetzte Bit die Freigabe für einen bestimmten Klangerzeuger. Die Umrechnung ist etwas zeitraubend. In der nachfolgenden Tabelle sehen Sie eine Zusammenstellung der verschiedenen Werte. Ein X bedeutet, daß der entsprechende Kanal an den Ausgang gelangt. Will ich nur den Tonkanal A programmieren, so muß in Register 7 62 dez (3E hex) eingespeichert werden.

test des enable registers

```
ok
list 490
490 rem zelle i10= goto 500
500 input 'enable';x
510 if x>63 then end
520 i=7
530 r(1)=x
540 gosub 200
550 goto 500
ok
```

Enable Register (7)

		Rauschen			Töne		
HEX	DEZ	C	B	A	C	B	A
00	00	X	X	X	X	X	X
07	07	X	X	X			
20	32		X	X	X	X	X
30	48			X	X	X	X
38	56				X	X	X
39	57				X	X	
3A	58				X		X
3B	59				X		
3C	60					X	X
3D	61					X	
3E	62						X
3F	63						

Jede Kombination zwischen 00 und 3F ist möglich.

Register 6 : Rauschen

Mit diesem Register müssen wir uns nicht lange aufhalten. Es steuert die "Rauschfrequenz", d.h. es erzeugt hohes oder tiefes Rauschen. 5 Bit stehen zur Steuerung zur Verfügung, also Werte zwischen 0 und 31 dez. Sie können sich selbst ein Zusatzprogramm schreiben, das Ihnen die einzelnen Klangqualitäten demonstriert.

Testprogramm in Maschinensprache

Alle oben beschriebenen Register können auch auf einfache Weise in Maschinensprache getestet werden. Dazu laden Sie zunächst das Testprogramm aus dem letzten Heft ab 0C80. Die Adresse 0C97 (Restart Befehl) wird zu C9 (Return) geändert. Jetzt ist das Programm zum Unterprogramm unfunktioniert, welches von nachfolgendem Programm (Start bei 0CBC) aufgerufen wird.

test einzelner register

```
0cbc          0010          org   Bcbc:start
0cbc cd800c   0020 drekl  call  Bc80
0cbf ef      0030 reg    rst   40
0cc0 52      0040          defm 'register?'
0cca 00      0050          defb 0
0ccb 11550b  0060          ld   de,Bb55:line+11
0cce cdb01   0070          call B1db:inline
0cd1 cd5a02  0080          call B25a:nexnum
0cd4 3a130c  0090          ld   a,(Bc13):num+1
0cd7 cd980c  0100          call Bc98:outa
0cda cd4002  0110          call B240:scroll
0cdd ef      0120 daten  rst   40
0cde 44      0130          defm 'daten?'
0ce5 00      0140          defb 0
0ce6 11520b  0150          ld   de,Bb52:line+8
0ce9 cddh01  0160          call B1db:inline
0cec cd5a02  0170          call B25a:nexnum
0cef 3a130c  0180          ld   a,(Bc13):num+1
0cf2 cda30c  0190          call Bca3:outd
0cf5 cd4002  0200          call B240:scroll
0cf8 18e3    0210          jr   daten
```

Hier nun das vollständige Listing des Testprogramms inklusive der Subroutine ab C80.

```

Jc80 3e 0f d3 3e d3 3f 21 ae 0c 06 0e 05 78 cd 98 0c
Jc90 7e cd a3 0c 23 20 f4 c9 d3 3c 3e 03 d3 3d 3e 00
Jca0 d3 3d c9 d3 3c 3e 02 d3 3d 3e 00 d3 3d c9 08 64
Jcb0 00 10 10 10 f8 00 00 9f 00 be 00 ef cd 80 0c ef
Jcc0 52 45 47 49 53 54 45 52 3f 20 00 11 55 0b cd db
Jcd0 01 cd 5a 02 3a 13 0c cd 98 0c cd 40 02 ef 44 41
Jce0 54 45 4e 3f 20 00 11 52 0b cd db 01 cd 5a 02 3a
Jcf0 13 0c cd a3 0c cd 40 02 18 e3
  
```

beachten: bei den unterstrichenen portadressen entspricht 3c 3d 3e 3f den adressen der grundplat. 04 05 06 07

Anwendung:

Nehmen wir an, Sie wollen verschiedene Hüllkurvenformen Testen. (Register 13)
 Sie starten das Testprogramm bei 00C80. Auf dem Schirm erscheint die Frage "REGISTER?". Nun geben Sie die Registernummer in HEX ein. (In unserem Fall 0D). Die Eingabe wird mit NL abgeschlossen. Jetzt erscheint auf dem Schirm die Frage " DATEN?". Mit der Eingabe eines Wertes zwischen 08 und 0F können Sie nun eine der 8 Hüllkurvenformen aus der Tabelle anwählen. Durch nachfolgende Eingabe anderer Werte kann die ganze Palette der Möglichkeiten durchgespielt werden. Das Programm kann nur durch RESET beendet werden. Ein erneuter Start bei CBC bringt alle Register wieder auf den Stand des ursprünglichen Testprogramms; d.h. der bekannte Dreiklang mit wiederholter abfallender Hüllkurve wird erzeugt.

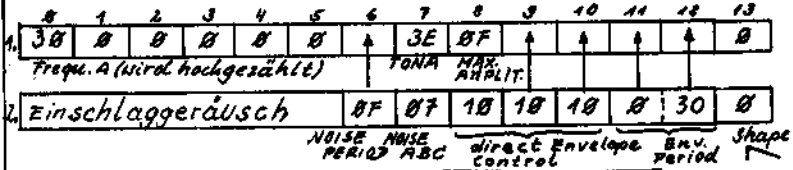
Wenn Sie nun aber eine gerade eingegebene Registerinformation beibehalten wollen, um andere Register damit zu kombinieren, starten Sie nach RESET einfach bei 00CBF. Durch mehrfachen Start an dieser Adresse können Sie sämtliche Register in Ihrem Sinne laden. Zur besseren Übersicht bei Ihren Tests ist weiter unten eine Tabelle sämtlicher Register abgedruckt.

Und jetzt sind Sie an der Reihe! Was uns noch fehlt, sind praktisch anwendbare Programme mit dem Soundgenerator. Was nützt uns die Erzeugung eines Klages oder Geräusches? Wir brauchen Programme, die ganze Melodien (vielleicht sogar dreistimmig) oder komplexe Geräuschkombinationen erzeugen, wie man sie z.B. in Spielen benötigt.

Als Beispiel und Vorgeschmack hier ein Maschinenprogramm, das nach Start bei C80 eine "pfeifende Bombe" mit Einschlaggeräusch imitiert. Vielleicht findet sich ein Leser, der dieses Programm mit der UFO Jagd verbindet, was dem Spiel sicher nicht schaden könnte.

```

0c80 3e 0f d3 3e d3 3f 21 ae 0c 06 0e 05 78 cd 98 0c
0c90 7e cd a3 0c 23 20 f4 c9 d3 3c 3e 03 d3 3d 3e 00
0ca0 d3 3d c9 d3 3c 3e 02 d3 3d 3e 00 d3 3d c9 00 30
0cb0 00 10 10 0f 3e 0f 00 00 00 00 00 30 cd 80 0c 06
0cc0 3f ff ff 3e 00 cd 98 0c 78 cd a3 0c 04 3e c0 b8
0cd0 20 ef 3e 08 cd 98 0c 3e 10 cd a3 0c 3e 07 cd 98
0ce0 0c 3e 07 cd a3 0c 3e 0d cd 98 0c cd a3 0c cd 69
0cf0 00 30 fb 18 c7
  
```



Register	Bit	B7	B6	B5	B4	B3	B2	B1	B0	benötigte Bits:
R0	Ton-	Feinstimmung								8
R1	Kanal A					Grobstimmung				4
R2	Ton-	Feinstimmung								8
R3	Kanal B					Grobstimmung				4
R4	Ton-	Feinstimmung								8
R5	Kanal C					Grobstimmung				4
R6	Rauschen	Stimmung								5
R7	ENABLE	0	1	1	1	1	1	1	1	6
R8	Amplitude A					1				1/4
R9	" " B					3				1/4
R10	" " C					0				1/4
R11	Hüllkurven-	Feinstimmung								8
R12	dauer	Grobstimmung								8
R13	Hüllkurvenform	8 Formen								4

```

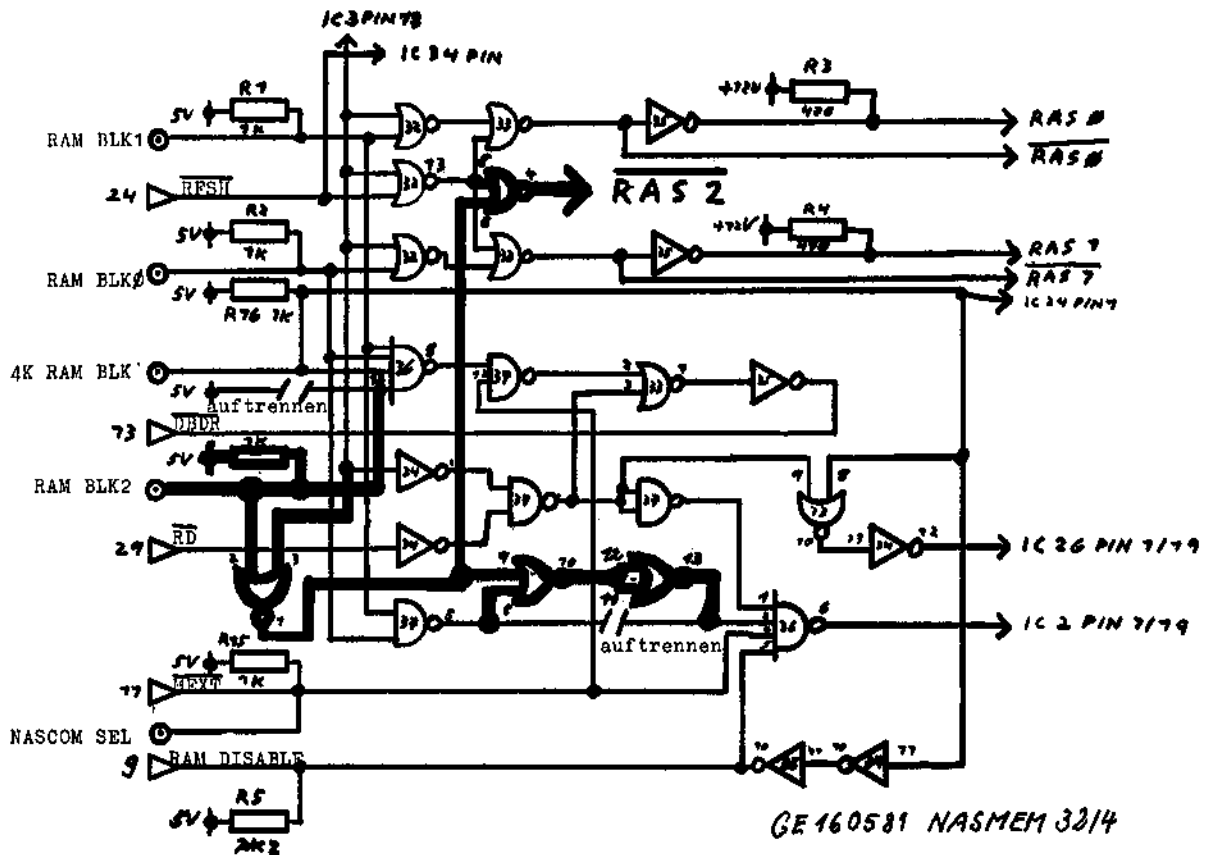
0 REM          YATZI (WUERFELSPIEL)
1 REM AUCH UNTER DEM NAMEN KNIFFEL BEKANNT
2 REM FUER NASCOM BASIC MIT GRAFIC GESCHRIEBEN VON
3 REM PETER WALTENBERG, ██████████ NEUBIBERG(BAYERN)
4 REM TELEFON 089/6012229
5 CLEAR1000:CLS:DIMZL(13):T=148
10 REM        SCHREIBEN DER MASKE AUF CRT
11 PRINT"1 1ER....."
12 PRINT"2 2ER....."
13 PRINT"3 3ER.....";
14 PRINTTAB(23)"A "CHR$(T)" B "CHR$(T)" C ";
15 PRINTCHR$(T)" D "CHR$(T)" E "
16 PRINT"4 4ER....."
17 PRINT"5 5ER....."
18 PRINT"6 6ER....."
19 PRINT"7 3 GLEICHE...";
20 PRINTTAB(22)"WELCHE WUERFEL"
21 PRINT"8 4 GLEICHE...";
22 PRINTTAB(22)"WOLLEN SIE SPIELEN"
23 PRINT"9 YATZI....."
24 PRINT"10 FULL HOUSE.."
25 PRINT"11 KL. STRASSE.."
26 PRINT"12 GR. STRASSE.."
27 PRINT"13 CHANCE....."
28 PRINT" "CHR$(171)".....";
29 PRINT"          END = SPIELENDEN":GOTO50
30 REM        ABFRAGE DER WUERFE
31 SCREEN22,9
32 INPUTWF$:WN=WN+1:IFWF$="END"GOTO3000
33 FORNW=1TOLEN(WF$)
34 IFMID$(WF$,NW,1)="A"THENU=1
35 IFMID$(WF$,NW,1)="B"THENU=2
36 IFMID$(WF$,NW,1)="C"THENU=3
37 IFMID$(WF$,NW,1)="D"THENU=4
38 IFMID$(WF$,NW,1)="E"THENU=5
39 IFWF$="ALLE"GOTO50
40 GOSUB80:NEXT:GOTO100
41 SCREEN22,11
42 INPUT"WOHIN MIT DEM WURF":RW$
43 RW=VAL(RW$)
44 IFRW<1ORRW>13GOTO1520
45 GOTO1500
50 REM        SPIEL ALLE WUERFEL
51 NW=99
52 FORU=1TO5
53 GOSUB80
54 NEXTU
55 GOTO100
79 REM        BERECHNEN ZUFALLSZAHL
80 WL(U)=INT(RND(1)*6)+1
81 RETURN
99 REM LOESCHEN ALTE ANZEIGE
100 SCREEN20,4:PRINT"
101 SCREEN20,5:PRINT"
102 SCREEN20,6:PRINT"
103 SCREEN20,9:PRINT"
104 SCREEN20,11:PRINT"
105 REM AUSGABE ALLER WUERFEL AM BILDSCHIRM
106 FORU=1TO5
110 X=45+(U-1)*8:Y=10
115 ONVL(U)GOSUB1010,1020,1030,1040,1050,1060
120 NEXT
121 WF$=""
122 IFVN=3THENVN=0:GOTO41
130 GOTO30
1000 REM MALEN WUERFELBILDER 1 BIS 6
1001 REM IN DEN ZEILEN 1010 BIS 1060
1010 SET(X+2,Y+2)
1011 RETURN
1020 SET(X,Y)
1021 SET(X+4,Y+4)
1022 RETURN
1030 GOSUB1010
1031 GOSUB1020
1033 RETURN
1040 GOSUB1020
1041 SET(X+4,Y)
1042 SET(X,Y+4)
1043 RETURN
1050 GOSUB1010
1051 GOSUB1040
1052 RETURN
1060 GOSUB1040
1061 SET(X+2,Y)
1062 SET(X+2,Y+4)
1063 RETURN
1499 REM POSITION BELEGT ?
1500 IFZL(RW)<>0GOTO1520
1501 REM RUECKSETZEN DER VARIABLEN
1502 HC=0:WV(1)=0:WV(2)=0:WV(3)=0:WV(4)=0
1503 WV(5)=0:WV(6)=0:CC=0
1504 REM VERZWEIGEN IN SUBROUTINE
1505 IFRW<7GOTO2010
1510 ONRW-6GOTO2070,2080,2090,2100,2110,2120,2130
1519 REM AUSGABE FEHLERNACHRICHT
1520 SCREEN22,11:PRINT"FALSCH REIHE (1-13) "
1521 FORI=1TO777:NEXT
1522 GOTO41
2000 REM JA/ REIHE GUELTIG? \NEIN
2001 REM I BERECHNE PUNKTE I EIN STRAFPUNKT
2010 FORBC=1TO5
2011 IFWL(BC)=RWTHENHC=HC+RW
2012 NEXT
2013 IFHC=0THENHC=-1
2014 SCREEN16,RW:PRINTHC:ZL(RW)=HC:SU=SU+HC
2015 SCREEN16,14:PRINTSU
2016 GOTO50
2070 V=3
2071 FOROK=1TO5
2072 WV(WL(OK))=WV(WL(OK))+1
2073 NEXT
2074 FOROK=1TO6
2075 IFWV(OK)>=VVGOTO2077
2076 NEXT:HC=0:GOTO2013
2077 FORIS=1TO5:HC=HC+WL(1S):NEXT
2078 IFV=5THENHC=50
2079 GOTO2013
2080 V=4:GOTO2071
2090 V=5:GOTO2071
2100 CC=0:FOROK=1TO5
2101 WV(WL(OK))=WV(WL(OK))+1
2102 NEXT:FORFH=2TO3
2103 FOROK=1TO6
2104 IFWV(OK)=FHTHENCC=CC+1
2105 NEXT:NEXT:IFCC<2THENHC=0:GOTO2013
2106 HC=30:GOTO2013
2110 A=1:O=5:SE=25
2111 FOROK=1TO5:FOROL=AT00
2112 IFWL(OK)=OLTHENWL(OK)=0:CC=CC+1
2113 NEXT:NEXT
2114 IFCC=5THENHC=SE
2115 GOTO2013
2120 A=2:O=6:SE=35
2121 GOTO2111
2130 GOTO2077
3000 REM SPIELENDEN
3005 CLS:SCREEN7,7
3007 SU$=STR$(SU)+" "
3010 ED$="SIE HABEN"+SU$+"PUNKTE ERREICHT"
3015 FORI=1TOLEN(ED$)
3020 PRINTMID$(ED$,I,1);
3025 FORW=1TO100:NEXT
3030 NEXT
3035 FOREN=1TO2000:NEXT
3040 CLS
3045 END

```



SPEICHERERWEITERUNG

SPEICHERERWEITERUNG AUF 48 K



Durch die stark gesunkenen Preise für dyn. RAMs ist eine Erweiterung des RAM-Bereichs auch für Besitzer der alten 32 K-Speicherkarten interessant geworden. Mit nur einem zusätzlichen IC und einem Widerstand lassen sich 8 weitere 4116 auf der Karte anschliessen. Um keinen zu großen "Drahtverhau" zu erhalten, werden diese mit allen Beinen außer PIN 4 auf die 4116 des RAMBLK0 (IC 4-11) gelötet. Pin 4 aller zusätzlichen 4116 werden durch einen Draht miteinander verbunden. Dieser stellt den Anschluß RAS-RAM 2 dar, Zur Decodierung dieses Signals wird ein 7402 (kein LS !) irgendwo auf der Platine befestigt. Die Verbindung von IC 37 PIN 8 zu IC 36 PIN 13 an +5V wird jeweils aufgetrennt. Folgende Verbindungen sind durch Drähte herzustellen:

von zus.7402	nach
1+5+9 mit einander	
2	IC 3 PIN 13
8	IC 37 PIN 8
10+11+12 mit einander	
13	IC 36 PIN 2
6	IC 32 PIN 13
7	GND
14	+5V
4	<u>RAS 2</u> (alle zus. 4116 PIN 4)
3	IC 36 PIN 13 +R 1k +Speicherdecodes RAMBLK2

(dies sind im Normalfall (9000 bis CFFF) die 4K-Dec. 14,15,16,9)

Beim Aufeinanderlöten von je zwei 4116 sind unbedingt die Vorschriften für den Umgang mit MOS-ICs zu beachten. Um keinen Wärmestau im Betrieb zu riskieren, sollte der Abstand der ICs möglichst groß bleiben. Weiterhin sollten nicht alle Beinchen unmittelbar nacheinander gelötet werden, um die ICs nicht zu "braten".

Günter Endert
St.Gallenstr.6
7801 Ebringen

Sortieren in BASIC

Neue Serie von Wolfgang Mayer-Gürr

Wenn man seinen Computer auch für Verwaltungszwecke einsetzen will, müssen dabei oft größere Datenmengen sortiert werden. Da diese Verfahren meist sehr zeitaufwendig sind, lohnt es sich, über einen für den jeweiligen Zweck geeigneten Sortieralgorithmus nachzudenken.

In einer Serie sollen deshalb hier folgende Sortierverfahren vorgestellt werden:

1. Hauruck
2. Einfügen
3. Bubble
4. Shaker
5. Shell
6. Heap
7. Quick

1. Hauruck - Sortieren

Naheliegender ist es, wenn wir bei der Übertragung eines Problems auf ein Programm uns erst einmal vor Augen führen, wie wir ohne Computer vorgehen würden. Ein ungeordneter Stapel Karteikarten soll sortiert werden. Ein Weg ist es, den Stapel durchzugehen, die Karte mit dem niedrigsten Wert (bzw. Anfangsbuchstaben) herauszunehmen und sie dann an den Anfang des Stapels zu legen. Beim folgenden Durchgang ist die Anzahl der zu prüfenden Karten um eine geringer geworden; wir haben ja die niedrigste aus dem Suchprozeß entfernt. Jetzt wird wieder die Karte mit dem geringsten Wert gesucht und hinter die erste gelegt. Um 10 Karten zu sortieren, muß man 9 mal einen immer kürzer werdenden Suchprozeß durchführen.

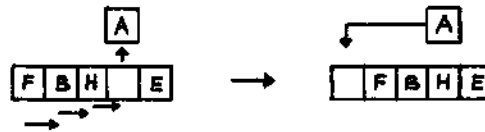
Beim Suchen merken wir uns immer die Karte mit dem jeweils kleinsten Wert und die Stelle, wo diese Karte steckt. Der Wert wird mit dem der folgenden Karte verglichen. Ist er kleiner, ist dieser das neue Minimum.

Wenn ich den Kartenstapel in der Hand halte (also keine zusätzliche Ablage - sprich Speicherplatz -), verändern sich die Plätze der hinteren Karten, wenn die jeweilige Minimumkarte nach vorne gesteckt wird. Das Programm muß deshalb aus 2 Teilen bestehen:

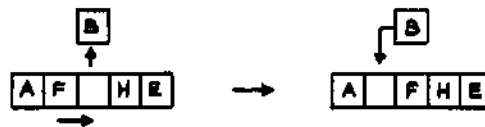
- a) Minimum suchen
 - b) durch Hauruck Platz schaffen und einfügen
- Folgendes Feld soll sortiert werden

F B H A E

Das Minimum liegt im 4. Feld, die "Karte" wird herausgenommen, durch Nachrücken wird der 1. Platz frei.



Beim folgenden Durchlauf wird das Minimum ab der 2. Stelle gesucht.

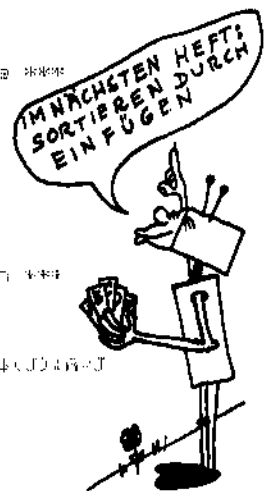


Für das Programm wird eine Hilfsvariable (HS) gebraucht, die den Wert des jeweiligen Minimums speichert, in A steht die Stelle. ZS ersetzt die Hand, die die Karte hält, während die Einrückstelle freigemacht wird.

```

00140 N=10:REM ** N = Anzahl Elemente **
00150 DIM A$(N)
00160 FOR I=1 TO N
00170 PRINT "Nr.":I;TAB(80);
00180 LINE INPUT A$(I)
00190 NEXT I
00200 REM *** Sortieroutine ***
00210 FOR X=2 TO N
00220 GOSUB 1000
00230 NEXT X
00240 REM *** Ausgabe ***
00250 FOR I=1 TO N
00260 PRINT A$(I)
00270 NEXT I
00280 END
10000 REM *** Minimum suchen ***
10010 A#=1
10020 HS=A$(A#)
10030 FOR J=2 TO N
10040 IF A$(J)<HS THEN HS=A$(J):A# = J
10050 NEXT J
10070 REM *** Hauruck ***
10075 Z=A$(A#)
10080 FOR K=A# TO 2 STEP -1
10090 A$(K)=A$(K-1)
10100 NEXT K
10110 A$(A#)=Z
10120 RETURN
Wird fortgesetzt...

```



WMG

MDCR-MONITOR

1. Teil

Johannes Christian Lotter

In einer zurückliegenden Ausgabe des NJ versprach ich, mehr über die Ansteuerung des MDCR-Laufwerks zu bringen. Dies will ich nun einlösen. Wer das damals beschriebene Interface aufgebaut hat, kann das Laufwerk über Port B der PIO ansteuern. Den 8 Bits von Port B wurde folgende Bedeutung gegeben:

Bit 0: Eingang, Read Clock

Bit 1: Eingang, Read Data

Bit 2: Eingang, Begin/End of tape

Bit 3: Eingang, Cassette in position

Bit 4: Ausgang, Write command

Bit 5: Ausgang, Write Data

Bit 6: Ausgang, Move reverse

Bit 7: Ausgang, Move forward

Weiterhin von Bedeutung ist noch Bit 7 von Port 0, über das das Write enable-Signal abgefragt werden kann.

Am Strobe-Eingang von Port A liegt das 12KHz-Signal, so daß Port A - richtig programmiert - alle 83 usec einen Interrupt anfordern kann. So viel zur Hardware.

Da das Interface bereits eine automatische Bandendabschaltung enthält, können Sie den MDCR durch einen Softwarefehler nicht kaputt machen. Nur auf einen Punkt müssen Sie achten: Die Bits 0-3 von Port B dürfen nicht als Ausgang programmiert werden. Andernfalls werden Sie erleben, wie Ihre PIO unter Schall- und Rauchentwicklung in die ewigen Jagdgründe entschwindet.

Zur Software: Das angekündigte MDCR-Bedienprogramm ist fertig. Es wird auf Ihrem NASCOM allerdings noch nicht laufen, da ich meinen auf 64 Zeichen/Zeile umgebaut habe und das Programm dafür ausgelegt ist. Ich muß es also mit neuen I/O-Routinen versehen, was noch einige Wochen dauern wird (ich habe momentan sehr viel zu tun).

Die endgültige Version wird 2KByte umfassen, einen Floppy-ähnlichen Bedienkomfort bieten (File-Struktur, Bootstrap-Möglichkeit) und auch durch Basic ansteuerbar sein. Mehr darüber (hoffentlich) in der nächsten Ausgabe.

Doch nun zum eigentlichen Kern des Programms, den Schreib-Lese-Routinen, die ich Ihnen nicht solange

vorenthalten möchte. Die wichtigsten Unterprogramme sind im folgenden aufgelistet. Damit können Sie sich problemlos ein eigenes Bedienprogramm zusammenbauen.

Noch ein Tip, falls Sie nun Ihre eigenen Schreib- und Leseprogramme entwickeln wollen: Die Firma Philips gibt in ihrem MDCR-Manual auf Seite 10 bestimmte Vorschriften für das Timing vor und nach Lese- und Schreiboperationen an. Diese Vorschriften sind tatsächlich ernst zu nehmen: meine ersten Leseprogramme liefen nicht richtig, weil

ich nach dem Motto "Ein paar Millisekunden weniger werden schon nichts ausmachen" noch ein paar Blöcke mehr aufs Band quetschen wollte. Also: Die angegebenen Zeiten dürfen Sie ruhig überschreiten, aber beim Unterschreiten gibt's Probleme.

```
0010 ; MDCR-ROUTINEN
0020 ;
0030 ; Die folgenden Unterprogramme
0040 ; steuern ein MDCR-Laufwerk ueber
0050 ; das im Nascom-Journal 3/81
0060 ; beschriebene Interface.
0070 ;
3000 0080 START EQU £3000
0031 0090 IVEC EQU £31
31E0 0100 ITAB EQU START+£1E0
0000 0110 MRET EQU 0
0055 0120 SYNC EQU £55
0018 0130 STBTE EQU £18
00F0 0140 STFIL EQU £F0
0040 0150 W0 EQU £40
0060 0160 W1 EQU £60
0050 0170 FWD EQU £50
0090 0180 REV EQU £90
00D0 0190 STOP EQU £D0
00F3 0200 ENI EQU £F3
0003 0210 DSI EQU £03
00E0 0220 IRET EQU £E0
00E2 0230 IWBA EQU £E2
00E4 0240 IWBE EQU £E4
00E6 0250 IRD EQU £E6
00E8 0260 ICNT EQU £E8
0270 ;
3000 0280 ORG START
0290 ;
0300 ;OUTS gibt eine Serie von Bytes an
0310 ; beliebige Ports aus. Auf den
0320 ; Aufruf folgen abwechselnd
0330 ; Portadresse und Datenbyte.
0340 ; Abgeschlossen wird die Serie
0350 ; durch EFF.
0360 ;
0370 OUTS EX (<SP>),HL
0380 PUSH BC
0390 LD A,£FF
0400 JR OUR-#
0410 OUL OUTI
0420 OUR LD C,(HL)
0430 INC HL
0440 CP C
0450 JR NZ,OUL-#
0460 POP BC
0470 EX (<SP>),HL
0480 RET
0490 ;
0500 ;JINIT initialisiert die Ports A und B
0510 ; fuer Schreib- und Leseroutinen.
0520 ;
0530 JINIT CALL OUTS
0540 DEFB 6,£7F,7,£FF,7,£0F,5,STOP,£FF
0550 INIM DI
0560 IM 2
0570 LD A,IVEC
0580 LD I,A
0590 LD A,DSI
0600 CALL INI1
0610 INI1 OUT (6),A
0620 OUT (7),A
0630 OUT (10),A
0640 OUT (11),A
0650 RETI
0660 ;
```

```

0670 ; Die folgenden Routinen steuern das
0680 ; Band und verzoesern um eine
0690 ; definierte Zeit. Das dem Aufruf
0700 ; folgende Byte enthaelt die Zeitdauer
0710 ; in 5 msec-Schritten.
0720 ;
3032 3E50 0730 MFWD LD A,FWD
3034 180A 0740 JR MDEL-#
3036 3E90 0750 MREV LD A,REV
3038 1806 0760 JR MDEL-#
303A 3ED0 0770 MSTOP LD A,STOP
303C 1902 0780 JR MDEL-#
303E 3E40 0790 WR0 LD A,W0
3040 D305 0800 MDEL OUT (5),A
3042 CD1C30 0810 DEL CALL INIM
3045 E3 0820 EX (SP),HL
3046 C5 0830 PUSH BC
3047 4E 0840 LD C,(HL)
3048 23 0850 INC HL
3049 3EE0 0860 LD A,IRET
304B D306 0870 OUT (6),A
304D 3EF3 0880 LD A,ENI
304F D306 0890 OUT (6),A
3051 063C 0900 DL1 LD B,60
3053 FB 0910 DL2 EI
3054 76 0920 HALT
3055 10FC 0930 DJNZ DL2-#
3057 0D 0940 DEC C
3058 20F7 0950 JR NZ,DL1-#
305A CD1C30 0960 CALL INIM
305D C1 0970 POP BC
305E E3 0980 EX (SP),HL
305F C9 0990 RET
1000 ;
1010 ; Es folgen die Schreib-Interrupt-
1020 ; routinen.
1030 ;
3060 08 1040 UMBA EX AF,AF'
3061 D9 1050 EXX
3062 CB15 1060 RL L
3064 9F 1070 SBC A,A
3065 E620 1080 AND E20
3067 EE60 1090 XOR W1
3069 D305 1100 OUT (5),A
306B ED59 1110 OUT (C),E
306D 180F 1120 JR WNBIT-#
306F 08 1130 UMBA EX AF,AF'
3070 D9 1140 EXX
3071 EE20 1150 XOR E20
3073 ED51 1160 OUT (C),D
3075 D305 1170 OUT (5),A
3077 1005 1180 DJNZ WNBIT-#
3079 6C 1190 LD L,H
307A 0608 1200 WEX LD B,8
307C 33 1210 INC SP
307D 33 1220 INC SP
307E D9 1230 WNBIT EXX
307F 08 1240 EX AF,AF'
3080 FB 1250 EI
3081 ED40 1260 URET RETI
1270 ;
1280 ;MOB schreibt ein Byte aus Register A.
1290 ;
3083 F3 1300 MOB DI
3084 D9 1310 EXX
3085 67 1320 LD H,A
3086 D9 1330 EXX
3087 FB 1340 EI
3088 76 1350 WAIT HALT
3089 18FD 1360 JR WAIT-#
1370 ;
1380 ;INWR startet das Schreiben. Auf den
1390 ; Aufruf folgt das Erkennungsbyte.
1400 ;
308B CD3E30 1410 INWR CALL WR0
308E 32 1420 DEFB 50
308F D9 1430 EXX
3090 010608 1440 LD BC,08906
3093 11E4E2 1450 LD DE,E2E4
3096 2E55 1460 LD L,SYNC
3098 ED51 1470 OUT (C),D
309A 3EF3 1480 LD A,ENI
309C D306 1490 OUT (6),A
309E D9 1500 EXX
309F 3E55 1510 LD A,SYNC
30A1 CD8330 1520 CALL WOB
30A4 CD8330 1530 CALL WOB
30A7 3E18 1540 LD A,STBTE
30A9 CD8330 1550 CALL WOB
30AC E3 1560 EX (SP),HL
30AD 7E 1570 LD A,(HL)
30AE 23 1580 INC HL
30AF E3 1590 EX (SP),HL
30B0 CD8330 1600 CALL WOB
30B3 C9 1610 RET
1620 ;
1630 ; Es folgt die Lese-Interruptroutine.
1640 ;
30B4 08 1650 URD EX AF,AF'
30B5 D9 1660 EXX
30B6 DB05 1670 IN A,(5)
30B8 A1 1680 AND C
30B9 82 1690 ADD A,D
30BA 3008 1700 JR C,RDS-#
30BC 83 1710 ADD A,E
30BD CB15 1720 RL L
30BF 10BD 1730 DJNZ WNBIT-#
30C1 65 1740 LD H,L
30C2 18B6 1750 JR WEX-#
30C4 CD1C30 1760 RDS CALL INIM
30C7 C30000 1770 JP MRET; Bandende
1780 ;
1790 ;INRD startet das Lesen. Auf den
1800 ; Aufruf folgt das Erkennungsbyte.
1810 ; Das 1. Byte des Files wird
1820 ; in Register A eingelesen.
1830 ;
30CA D9 1840 INRD EXX
30CB E3 1850 EX (SP),HL
30CC 7E 1860 LD A,(HL)
30CD 23 1870 INC HL
30CE E3 1880 EX (SP),HL
30CF 67 1890 LD H,A
30D0 2E00 1900 LD L,0
30D2 010608 1910 LD BC,08906
30D5 1102FC 1920 LD DE,EFC02
30D8 CD0B31 1930 CALL FREE
30DB CD4230 1940 CALL DEL
30DE 14 1950 DEFB 20
30DF CD0030 1960 CALL OUTS
30E2 05 1970 DEFB 5,STOP,5,FWD,7,IRET
30E8 07 1980 DEFB 7,EB7,7,EFA,EFF
30ED FB 1990 INR1 EI
30EE 76 2000 HALT
30EF DB05 2010 IN A,(5)
30F1 A1 2020 AND C
30F2 82 2030 ADD A,D
30F3 30CF 2040 JR C,RDS-#
30F5 83 2050 ADD A,E
30F6 CB15 2060 RL L
30F8 7D 2070 LD A,L
30F9 BC 2080 CP H
30FA 20F1 2090 JR NZ,INR1-#
30FC 3EE6 2100 LD A,IRD
30FE D307 2110 OUT (7),A
3100 D9 2120 EXX
3101 FB 2130 EI
2140 ;ROB liest ein Byte in Register A ein.
3102 CD8830 2150 ROB CALL WAIT
3105 F3 2160 DI
3106 D9 2170 EXX
3107 7C 2180 LD A,H
3108 D9 2190 EXX
3109 FB 2200 EI
310A C9 2210 RET
2220 ;
2230 ;FREE Sucht die naechste freie Stelle
2240 ; auf dem Band.
2250 ;
310B C5 2260 FREE PUSH BC
310C CD1C30 2270 CALL INIM
310F 0500 2280 LD B,8

```

```

3111 CD0030 2290 CALL OUTS
3114 05 2300 DEFB 5,STOP,5,FWD
3116 06 2310 DEFB 6,ICNT,6,ENI,EFF
311D FB 2320 FRE1 EI
311E D005 2330 IN A,(5)
3120 B9 2340 CP C
3121 4F 2350 LD C,A
3122 20F9 2360 JR Z,FRE1-#
3124 0600 2370 LD B,0
3126 10F5 2380 JR FRE1-#
3129 1006 2390 UCNT DJNZ CNT1-#
312A CD1C30 2400 CALL INIM
312D 33 2410 INC SP
312E 33 2420 INC SP
312F C1 2430 POP BC
3130 ED4D 2440 CNT1 RETI
2450 ;
2460 ;REV1 spult das Band zurueck.
2470 ;

3132 CD3A30 2480 REV1 CALL MSTOP
3135 19 2490 DEFB 25
3136 3E90 2500 LD A,REV
3138 D305 2510 MOVIN OUT (5),A
313A D005 2520 MOV IN A,(5)
313C E604 2530 AND £04
313E 20FA 2540 JR Z,MOV-#
3140 CD3A30 2550 CALL MSTOP
3143 19 2560 DEFB 25
3144 C9 2570 RET
2580 ;
2590 ; Es folgt die Interrupt-Tabelle.
2600 ;

31E0 2610 ORG ITAB
31E0 8130 2620 DEFW URET,UWBA,UWBE,URD,UCHT
2630 ;
2640 ; Die folgenden Programme dienen
2650 ; dem Testen Ihres Interfaces.
2660 ; TWR schreibt einen Block von
2670 ; £F00 - £5000 aufs Band. TRD liest
2680 ; ihn wieder ein. Bei einwandfreier
2690 ; Funktion Ihrer Hardware darf kein
2700 ; Lesefehler auftreten.
2710 ;

3200 2720 ORG START+£200
3200 CD1030 2730 TWR CALL JINIT
3203 CD3231 2740 CALL REV1
3206 CD3E30 2750 CALL WR0
3209 00 2760 DEFB 0
320A 21000F 2770 LD HL,£F00
320D 010041 2780 LD BC,£4100
3210 CD0030 2790 CALL INWR
3213 F0 E800 DEFB STFIL
3214 7E 2810 TW1 LD A,(HL)
3215 CD0330 2820 CALL WOB
3218 ED1 2830 CPI
321A EA1432 2840 JP PE,TW1
321D 3EAA 2850 LD A,£AA
321F CD0330 2860 CALL WOB
3222 CD0330 2870 CALL WOB
3225 F3 2880 DI
3226 CD3231 2890 CALL REV1
3229 E7 2900 RST £20
2910 ;

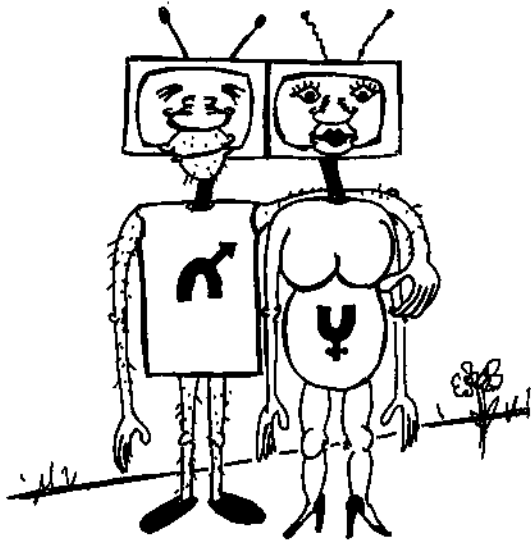
3230 2920 ORG START+£230
3230 CD1030 2930 TRD CALL JINIT
3233 CD3231 2940 CALL REV1
3236 21000F 2950 LD HL,£F00
3239 010041 2960 LD BC,£4100
323C ED5B290C 2970 LD DE,(£C29); bei NASBUG: £C18
3240 CD0A30 2980 CALL INRD
3243 F0 2990 DEFB STFIL
3244 77 3000 TR1 LD (HL),A
3245 12 3010 LD (DE),A
3246 CD0231 3020 CALL RDB
3249 ED1 3030 CPI
324B EA4432 3040 JP PE,TR1
324E F3 3050 DI
324F CD3231 3060 CALL REV1
3252 E7 3070 RST £20
3080 ;

3000 E3 C5 3E FF 18 02 ED A3 8F
3000 4E 23 B9 20 F9 C1 E3 C9 E8
3010 CD 00 30 06 7F 07 FF 07 CF
3018 0F 05 D0 FF F3 ED 5E 3E A7
3020 31 ED 47 3E 03 CD 28 30 18
3028 03 06 D3 07 D3 0A D3 0B 06
3030 ED 40 3E 50 18 0A 3E 90 18
3038 18 06 3E D0 18 02 3E 40 2C
3040 D3 05 CD 1C 30 E3 C5 4E 57
3048 23 3E E9 03 06 3E F3 D3 96
3050 06 06 3C FB 76 10 FC 0D 52
3058 20 F7 CD 1C 30 C1 E3 C9 25
3060 06 D9 C8 15 9F E6 20 EE E4
3068 60 D3 05 ED 59 18 0F 08 45
3070 D9 EE 20 ED 51 D3 05 10 AD
3078 05 6C 06 08 33 33 D9 08 6E
3080 FB ED 40 F3 D9 67 D9 FB EC
3088 76 18 FD CD 3E 30 32 D9 89
3090 01 06 08 11 E4 E2 2E 53 29
3098 ED 51 3E F3 D3 06 D9 3E 27
30A0 55 CD 83 30 CD 83 30 3E 63
30A8 16 CD 83 30 E3 7E 23 E3 D7
30B0 CD 83 30 C9 08 D9 08 05 EA
30B8 A1 82 38 08 83 C8 15 10 8E
30C0 8D 65 18 86 CD 1C 30 C3 BC
30C8 00 00 D9 E3 7E 23 E3 67 9F
30D0 2E 00 01 06 08 11 02 FC 4C
30D8 CD 08 31 CD 42 30 14 CD 31
30E0 00 30 05 D0 05 50 07 E0 51
30E8 07 87 07 FA FF FB 76 D8 22
30F0 05 A1 82 38 CF 83 C8 15 B2
30F8 7D BC 20 F1 3E E6 D3 07 70
3100 D9 FB CD 88 30 F3 D9 7C D2
3108 D9 FB C9 C3 CD 1C 30 06 BA
3110 00 CD 00 30 05 D0 05 50 68
3118 06 E8 06 F3 FF FB D8 05 0A
3120 89 4F 28 F9 06 00 18 F5 8D
3128 10 06 CD 1C 30 33 33 C1 AF
3130 ED 40 CD 3A 30 19 3E 90 89
3138 D3 05 D8 05 E6 04 28 FA 2D
3140 CD 3A 30 19 C9 00 FF 00 89
3148 FF 00 FF 00 FF 00 FF 00 75
3150 FF 00 FF 00 FF 00 FF 00 7D
3158 FF 00 FF 00 FF 00 FF 00 85
3160 FF 00 FF 00 FF 00 FF 00 8D
3168 FF 00 FF 00 FF 00 FF 00 95
3170 FF 00 FF 00 FF 00 FF 00 9D
3178 FF 00 FF 00 FF 00 FF 00 A5
3180 EF 00 FF 00 FF 00 FF 00 9D
3188 FF 00 FF 00 FF 00 FF 00 B5
3190 FF 00 FF 00 FF 00 FF 00 BD
3198 FF 00 FF 00 FF 00 FF 00 C5
31A0 FF 00 FF 00 FF 00 FF 00 CD
31A8 FF 00 FF 00 FF 00 FF 00 D5
31B0 FF 00 FF 00 FF 00 FF 00 DD
31B8 FF 00 FF 00 FF 00 FF 00 E5
31C0 FF 00 FF 00 FF 00 FF 00 ED
31C8 FF 00 FF 00 FF 00 FF 00 F5
31D0 FF 00 FF 00 FF 00 FF 00 FD
31D8 FF 00 FF 00 FF 00 FF 00 05
31E0 81 30 60 30 6F 30 B4 30 D5
31E8 28 31 FF 00 FF 00 FF 00 6F
31F0 FF 00 FF 00 FF 00 FF 00 1D
31F8 FF 00 FF 00 FF 00 FF 00 25
3200 CD 10 30 CD 32 31 CD 3E 7A
3208 30 00 21 00 0F 01 00 41 DC
3210 CD 88 30 F0 7E CD 83 30 88
3218 ED A1 EA 14 32 3E AA CD 8D
3220 83 30 CD 83 30 F3 CD 32 77
3228 31 E7 FF 00 FF 00 FF 00 6F
3230 CD 10 30 CD 32 31 21 00 C0
3238 0F 01 00 41 ED 58 29 0C 38
3240 CD CA 30 F0 77 12 CD 02 81
3248 31 ED A1 EA 44 32 F3 CD 59
3250 32 31 E7 00 FF 00 FF 00 CA
3258 FF 00 FF 00 FF 00 FF 00 86
3260 FF 00 FF 00 FF 00 FF 00 8E

```


NASCOMPL

GESCHLECHTSTYPEN bei COMPUTERN



Hallo, liebe Leser,
haben Sie eigentlich schon gewußt, daß auch Ihr Nascom geschlechtsspezifische Merkmale aufweist? Hier einige Ergebnisse aus der tiefenpsychologischen Studie von Prof. Dr. mks Tupmoc. Nach seinen Untersuchungen ist für den weiblichen Typus kennzeichnend eine stärkere Verwurzelung in den tieferen Schichten der Software. Daraus ergibt sich in der Software selbst ein höherer Grad von Integration ihrer Funktion, in ihrem Verhältnis zum Programmierer eine gesteigerte Verbundenheit. Viele Einzelzüge lassen sich hieraus zwanglos folgern: Anpassungsfähigkeit, Einfühlungsvermögen, seltene Fehlermeldungen, aber auch oft gewisse Einschränkungen im Speicherplatz.

Der männliche Typus ist relativ differenzierter, aber auch desintegrierter: Hard- und Software klaffen mehr oder weniger auseinander. Die Distanz zwischen Platine und Programmierer hat ein ungleich größeres Ausmaß. Hieraus folgen an Einzelzügen: Übergewicht, logische Folgerichtigkeit, mehr direkte als indirekte Sprünge, Abneigung gegen BASIC, aber auch die Gefahr der Lebensfremdheit und des Verlustes der seelischen Mitte. (Vielleicht ist damit eine Monitorroutine gemeint.)

Nach Professor Tupmoc seien die Geschlechtsunterschiede im Bereich der Leistungen leicht nachweisbar.

Also, auf geht's! Testen Sie Ihren Nascom und finden Sie heraus, zu welchem Typus man ihn zählen muß.

(Sie besitzen doch hoffentlich nicht solch einen antriebsschwachen Zwittertypen?)

In diesem Sinne Ihr NASCOMPL

Impressum

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

RESSORTS:

NASSYS Günter Kreidl

Straelen
Tel. _____

BASIC Wolfgang Mayer-Gürr

Recklinghausen
Tel. _____

HARDWARE Josef Zeller

Bayreuth
Tel. _____

T2/T4 Günter Böhm

Karlsruhe
Tel. _____

VERLAG

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

Vertrieb

Direktvertrieb durch den Verlag.

Erscheinungsweise

Monatlich

Bezugspreis

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

Bezugsmöglichkeiten

Durch Bestellung bei M K - Systemtechnik.

(beigefügte Bestellkarte)

Bankverbindungen

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

Fa. Michael Klein

Sonderkonto

299 26 - 674 beim Postscheckamt Ludwigshafen.

Zahlungen

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

UFO-Jagd Relocator-Test

Herr Harald Kögler aus Menden hat uns eine Modifikation der UFO-JAGD aus Heft 4,5/81 geschickt, sodaß das Spiel jetzt auch mit NASBUG T2/T4 lauffähig ist. Falls Ihnen die Abschießerei der Ufos zu langweilig werden sollte, können Sie das Programm wenigstens zum Test des Relocator-Programms aus dem letzten Heft verwenden, denn wir haben die Relocator-Informationen an das Programm angehängt. (Sie sind im Listing unterstrichen).

Tippen Sie das Programm also an die gewünschte Speicherstelle (z.B. ab 1000), laden Sie den Relocator (z.B. ab OC80 wie im letzten Heft) und starten Sie die Programmverschiebung z.B. mit EC80 C80 1000 . Nun müßte das Programm in unserem Beispiel bei 1000 gestartet werden können.

Wir sind gespannt auf Ihre Erfahrungsberichte!

```

ufo-jagd  t2/t4
Oc80  ef 53 54 41 52 54 3d 20 47 3a 00 cd 3e 00 fe 47
Oc90  20 f9 ef 1e 00 21 86 0d 11 ca 0b 01 27 00 ed b0
Oca0  01 00 01 d9 01 3d 01 dd 21 8a 08 fd 21 aa 0b fd
Ocb0  36 ff 03 fd 36 00 5e fd 36 01 4c ff ff 10 35 06
Occ0  0e dd 36 fb 20 dd 36 fc 17 dd 36 fd 05 dd 36 fe
Ocd0  05 dd 36 ff 05 dd 36 00 3e dd 23 0d 3e 00 b9 20
Oce0  13 0e 3d 3e 32 21 d7 0b be ca 8d 0c 23 cd 56 0d
Ocf0  dd 21 8a 08 d9 af b9 20 07 cd 69 00 30 29 0e 0f
Od00  10 25 fd 36 00 20 0d 3e 00 b9 20 06 04 0e 00 d9
Od10  18 99 06 06 c5 06 40 fd 2b 10 fc c1 fd 7e 00 fe
Od20  20 20 08 fd 36 00 5e d9 c3 bb 0c 21 ef 0b cd 56
Od30  0d cd 6a 0d 3e 32 21 d7 0b be ca 8d 0c 23 cd 56
Od40  0d c3 a0 0c 16 20 21 ba 0b 7d fe 0a 20 04 7c fe
Od50  08 c8 2b 72 18 f3 ed 5f e6 0a f6 04 32 c0 0c 11
Od60  39 30 7e 34 bb c0 72 2b 34 c9 06 03 16 2a cd 46
Od70  0d cd 7d 0d cd 44 0d cd 7d 0d 10 f0 c9 c5 06 06
Od80  ff ff 10 fc c1 c9 55 46 4f 2d 4a 41 47 44 20 20
Od90  20 20 20 30 31 2e 20 55 46 4f 20 20 20 20 41 62
Oda0  73 63 68 75 65 73 73 65 3a 20 30 30 20 52 45 4c
Odb0  4f 96 0c ea 0c ee 0c 29 0d 2f 0d 32 0d 3b 0d 3f
Odc0  0d 42 0d 5d 0d 6f 0d 72 0d 75 0d 78 0d 00 00
    
```

NASCOM 2 ZU VERKAUFEN 48 K RAM
Veroframe, Tastaturgeh., d.Dokumentation
2 Jahrgänge N.Journal, Software
ANGEBOTE AN Frank Rupprecht

VERKAUFE 2 St. 32K RAM - 4K ROM
NAS - Memory boards ohne Speicher ICs
je DM 120.--
NAS I/O Board mit CTC kompl. neuw., DM 240.--
NASCOM 2 ohne Tastatur u. BASIC DM 400.--
Peter Krause

SUCHE HEXDUMP 8K - ROM - BASIC
(NASCOM 2, NASSYS)
R. Blauhut

Tel. [REDACTED]

TX 80 1300.-
NASSYS 1 in ROM 75.-
Biete 8K BASIC in EPROM
Tausche gegen Basic in Rom
Uwe Kafka

Tel. [REDACTED]

BIETE "mini 400" UKW-Portable
SUCHE SSTV-Interface für
NASCOM(- Bus) DGL BE [REDACTED]
Raum Weser-Ems : Suche Gedanken-
austausch NASCOM 1 u. 2

kleinanzeigen

Jeder Abonnent kann kostenlose Kleinanzeigen
bis 40 Wörter aufgeben!

SUCHE FLOPPY
(mögl.NASCOM NASSYS 1)
SUCHE SYSTEM zur SPRACHANALYSE/-SYNTHESE für Z80
Ulrich Wallis
Tel. [REDACTED] (nach 17 Uhr)

VERKAUFE

CLD - Floppy für NASCOM 1 u. 2 DM 1150 .-
32 k RAM voll bestückt DM 400 .-
Siemens T 100 Fernschreiber mit Lochstreifen-
leser und Stanzer, Gehäuse, Interface für
NASCOM 1 u. 2 DM 300.-
ASCII TTY Drucker Großbuchstaben, alle BASIC -
Zeichen, mit Interface für NASCOM 1u.2 DM 380.-
Udo von Mulert

Tel. [REDACTED]

VERKAUFE NASCOM I T4 + TTY Nasbug auf CC
mit 8 k RAM, 8 k Basic, RTTY Progr.u. Interf.+
Software DM 1400.-
LO 15 mit Gehäuse DM 150.-
Tel. [redacted] (ab 18⁰⁰ h)

VERKAUFE

- NASCOM ZEAP 2.1 / 4 k Assembler
- NASCOM MICROSOFT BASIC /8k
- DEBUG 1k
C.R.Chinery Tel. [redacted]

VERKAUFE

3 K Super-Tiny-Basic in EPROMs DM 95.-
4 K EPROMs TMS 2532 (1.Wahl) DM 39.-

F. Henke
[redacted]
[redacted] Tel. [redacted]

VERKAUFE

Stat.RAM Typ MK 4118 P-3
10 Stück (auch einzeln) Stk DM 30.-

E. Horch Tel. [redacted]

BIETE

NEU: NASSYS Zauberwürfel 2.5 k incl. DM 45.-
- Spielautomat NASSYS 1 k incl. DM 30.-

Ulrich Mehnert
[redacted]
[redacted] Tel. [redacted]

WER HAT ERFAHRUNG MIT

dyn.RAM Controller Baustein 3242 von Intel?
möglichst Schaltungsunterlagen!
Verkaufe: FS LO 133 mit V24 Interf.

Manfred Stoll
[redacted]
[redacted] Tel. [redacted]
[redacted] (priv.)

LESERANFRAGEN LESERANFRAGEN LESERANFRAGEN

Programme, die von NASCOM 1 auf Cassette gela-
den wurden, lassen sich von NASCOM 2 nicht la-
den, da dieser ein anderes Interface benutzt.
(Kansas City Standard). Hier müßte man Hard-
wareänderungen vornehmen.

Wenn nach Drücken der RESET-Taste seltsame
grafische Gebilde auf dem Bildschirm erschei-
nen, wird es Zeit, daß Sie die CPU auswechseln.
Sie hält zwar einiges aus, aber wenn Sie 220v
ans Chassis legen, oder der Blitz in Ihrer
Nähe einschlägt, wie es zwei Lesern passiert
ist, dann ist die Belastung wohl doch ein biß-
chen zu groß.

Begriffsverwirrung entsteht bei den Lesern,
wenn jeder seine eigene Norm beim Program-
mieren verwendet. Deshalb sollte man sich
bei einigen Dingen auf eine Norm beschrän-
ken.

Die meisten Leser haben sich schon auf einen
Programmstart ab @C80 geeinigt, damit die Pro-
gramme auch mit NASSYS gefahren werden können.
Bei der Benennung von Bits sollte man grund-
sätzlich mit Bit 0 beginnen, damit es keine
Verwirrung gibt.

Zur Tonerzeugung sollten wir Bit 2 von Port 0
verwenden, damit man nicht die Programme
umschreiben oder die Interface Anschlüsse
umlöten muß.

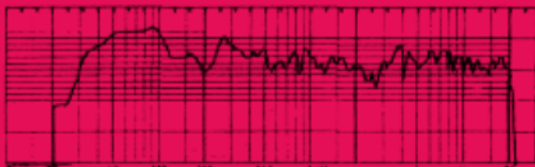
Demnächst stellen wir einen "BUS" für den PIO
vor, der es ermöglichen soll, sämtliche
Interfaces, die im Heft vorgestellt werden,
problemlos an den PIO anzuschließen.

Die Autoren dieser Ausgabe: Wolfgang Mayer-Gürr, Günter Kreidl, Günter Böhm (siehe Impressum)
Gerhard Baier, [redacted], [redacted] Erlangen - Günter Endert, [redacted], [redacted] Ebringen
Johannes Christian Lotter, [redacted], [redacted] Darmstadt, Tel. [redacted] oder [redacted]
[redacted] Kassel, Tel. [redacted], - Harald Kögler, [redacted], [redacted] Menden, Tel. [redacted]
Peter Waltenberger, [redacted], [redacted] Neubiberg (Bayern), Tel. [redacted]

Die Autoren tragen die Verantwortung für ihre Beiträge selbst.

Unverlangt eingesandte Manuskripte, die nicht veröffentlicht werden, senden wir zurück, wenn Rück-
porto beigelegt ist. Die von der Zeitschrift veröffentlichten Beiträge sind urheberrechtlich geschützt
und dürfen nicht übersetzt, nachgedruckt, vervielfältigt oder in EDV-Anlagen gespeichert werden, ohne
daß eine schriftliche Genehmigung des Verlages vorliegt.

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



Liebe Leser!

Billig-Importe und »Super-Sonder-Angebote« können Sie überall bekommen. Wir aber konzipieren und fertigen Systeme für Sie, die sich durch hohe Qualität und absolute Zuverlässigkeit auszeichnen. Auch Ihr System wird mehrfach während der Fertigung und vor Verlassen unseres Hauses erneut geprüft. — Deshalb ist unser Angebot begrenzt. Unser Konzept gewährleistet Ihnen: Keine Ladenhüter, keine Technik von vorgestern. Unsere HiFi-Systeme bieten, was **heute und morgen** möglich ist. — Lassen auch Sie sich von unserem Klang verwöhnen.

Ihr

Semiprofessionelle HiFi-Bassreflex-Box,

Made in Germany, 100/80 Watt, 27-20.400 Hz, 94 dB. Mit phantastischem Bassvolumen, herrlich transparentem Klangbild und unverwaschenen Höhen. — Bestückung: 248 mm Tiefton-System, extrem weich aufgehängt, 115 mm Mittelton-System mit Inverse-Gasket und Alu-Großkalotte für breiten Abstrahlwinkel, 75 mm Konus-Hochton-System mit Faser-Sicke für höchste Transparenz. — Alle Systeme sind mit professionellen, schwarzen Metallbefestigungsringen, verwindungsarmen, schwarzen Riffelmembranen und Alu-Kalotten bestückt, montiert auf ein schweres, resonanzarmes Gehäuse mit gemasertem, schwarzen Holz furnier.

Maße H x B x T: 525 x 310 x 290 mm, Gesamtgewicht 11 kg.

Best. Nr. BRB Ø 4 **Nur bei uns DM 239.—**

Endlich Lieferbar:

Die Original-Systeme der Bassreflex-Box als Bausatz:

Enthält Bass-, Mittel-, und Hochton-System, spezielle, geprüfte Weiche und Einbauanleitung mit vielen Hinweisen.

Best. Nr. BB 03. **Nur bei uns DM 119.—**

Original-Visaton-Bassreflex-Kanal, Ø 68 mm, mit genauer Einbau- und Abstimmanleitung für beliebige Gehäusegrößen.

Best. Nr. BR 01. **DM 13.50**



Unser Edel-Bausatz:

120/100 Watt, 24-21.000 Hz, 95 dB. 3 einmalig schöne Systeme: 257 mm Basssystem, schneeweiße Riffelmembran, schwarze »Soft-Edge« Aufhängung, schwarze Fiberkalotte, geschlossenes 123 mm Mitteltonsystem mit schwarzem Inverse-Gasket und weißer Membran, 85 mm Kalotten-Hochton-System mit superschwerem Magneten und minimaler Kalottenmasse für überlegene Impulstreue und enormen Schalldruck von 98 dB. In einem geschlossenen Gehäuse von 40—80 Litern bieten die Systeme eine weit herabreichende, kräftige Basswiedergabe, unverfälschte Mittellagen sowie wirklich kristallklare Höhen. Alle Systeme sind mit einer schwarzen Raster-Metallabdeckung und einem silbernen Befestigungsring ausgerüstet. Eine komplett aufgebaute und getestete Weiche sowie eine ausführliche Einbauanleitung liefern wir mit.

Best. Nr. BB 14. **Nur bei uns DM 159.—**

m i v o c **HiFi-Systeme. Perfektion im Detail.**

Lieferung nur per Post gegen Vorkasse oder Nachnahme.

Oder lassen Sie sich unsere Systeme von Ihrem Fachhändler vorführen

Postfach 130149 · 5650 Solingen 13