

# **nascocom**

## **JOURNAL 4/81**

### **5**

---

#### ZEITSCHRIFT FÜR BENUTZER DES NASCOM 1 ODER NASCOM 2

Herausgegeben von: M K - Systemtechnik  
Michael Klein  
Pater-Mayer-Straße 6  
Tel.: (07274) 2756  
Telex 0453500 mks d

Vertriebsstelle: M K - Systemtechnik  
Thomas Gräfenecker  
Kriegsstraße 164  
7500 Karlsruhe

Der Heftpreis beträgt DM 4.- Ein Abonnement erhalten Sie für 48.- im Jahr.  
Dafür bekommen Sie 12 Hefte pro Jahr bzw. 10 Hefte (zwei dicke Doppelhefte).

Die Autoren sind für den Inhalt ihrer Artikel selbst verantwortlich.  
Sämtliche Rückfragen bitte nur an die Autoren !! Jeder Abonnent kann kosten-  
lose Kleinanzeigen bis 40 Worte aufgeben.

---

#### Inhaltsverzeichnis

1	Editorial
2,3	Stephan Bürger Gedanken zum Thema NASCOM Erweiterung
4,5	Josef Zeller Anpassung des NASCOM-Bus-systemes an den ECB-Bus
6-8	Günter Böhm Erweiterungen des NASCOM auf KONTRON-Bus
9,10	Günter Böhm Anschluß einer KONTRON/ELZET I/O-Karte an NASCOM 1
11-16	Günter Böhm Nützliche Unterprogramme für NASCOM 1
17-21	Albert Schunck Integer PASCAL-Compiler für den NASCOM
22,23	Bernd Ploss Das Programm TTY-SYS
23,24	Bernd Ploss Strichcodeprogramme
25-28	Gerhard Baier UFO-Jagd
28	Gerhard Baier Lottozahlengenerator
29	Uwe Wurditsch Printplot mit Super Tiny Basic
30	
31	Kleinanzeigen

## Editorial

Liebe NASCOM-JOURNAL-Leser,

die letzten Wochen waren von Hektik gekennzeichnet. Die Firma ist umgezogen. Neue Anschrift siehe Deckblatt. Auch die Vertriebsstelle Karlsruhe ist umgezogen. Mehr Hektik geht schon fast nicht mehr.

Wir haben uns entschlossen, jetzt für April/Mai ein Sonderheft des NASCOM-JOURNAL herauszubringen. Verständlicherweise werden inzwischen einige Kunden ungeduldig.

Wir sind inzwischen nur noch mit einiger Mühe in der Lage, das NASCOM-JOURNAL auf gutem Niveau zu halten. Wochenenden und Sonntage sowie Feiertage gehen drauf, nur weil keine Zeit ist. Daher die Frage: Wer wäre bereit, die Redaktion des NASCOM-JOURNAL zu übernehmen? Und zwar gegen ein angemessenes Entgelt. Wir würden das JOURNAL nach wie vor drucken lassen und versenden. Sämtliche Beiträge unserer Leser würden dann direkt an den Chefredakteur gehen.

Eine Panne unseres Druckers beim letzten JOURNAL führte dazu, daß die meisten Ausgaben des JOURNAL 3/81 zwei unleserliche Seiten hatten. Wir werden diese beiden Seiten nachdrucken. Wir bitten die Autoren der schlecht gedruckten Listings außerdem, sie mögen uns nochmal je ein ganz einwandfreies Listing schicken, damit es beim Druck wirklich keine Probleme gibt.

Dieses NASCOM-JOURNAL enthält als Doppelausgabe 32 Druckseiten interessanter Information. Besonders möchte ich hinweisen auf die drei Beiträge, die sich mit dem Anschluß eines NASCOM an den ECB-Bus befassen. Ich halte diese Beiträge alle für recht interessant. Herr Böhm gibt sogar ein ganz konkretes Beispiel für einen solchen Anschluß an.

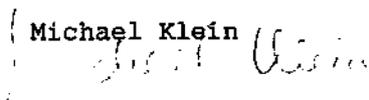
Wir haben alle drei Beiträge aufgenommen, weil die verschiedenen Überlegungen und die verschiedenen Problemlösungswege doch recht interessant erschienen.

Ausserdem befindet sich in diesem JOURNAL ein Beitrag von Albert Schunck über einen Integer-PASCAL-Compiler. Der sehr clevere Herr Schunck hat es tatsächlich fertiggebracht, einen PASCAL-Compiler selbst zu basteln. Alle Achtung! Sie können den Compiler übrigens bei MK-Systemtechnik erwerben.

Auch für dieses JOURNAL wünsche ich Ihnen eine angenehme Lektüre.

Ihr

Michael Klein



## Gedanken zum Thema Nascom Erweiterung

Die derzeitige Liefersituation für Nascom Teile ist bekanntlich denkbar schlecht. Darum sollen hier einige Möglichkeiten aufgezeigt werden wie ein N1 zu einem sehr leistungsfähigen Kleincomputer erweitert werden kann.

### 1. Auswahl eines geeigneten BUS-Systems:

Hier bietet sich der 64polige KONTRON-BUS, der auch vom ELZET-80 System verwendet wird, direkt an. Dieser BUS führt nahezu alle Signale des 77poligen NASCOM-BUS; das Europakartenformat ist, auch vom mechanischen Gesichtspunkt her, sehr universell; und das Wichtigste : Platinen für diesen BUS sind einfach und günstig erhältlich.

Für die im NASCOM-System zusätzlich notwendigen Leitungen gibt es noch genügend freie Stellen, sodaß es sich empfiehlt, die für Bank-Select vorgesehenen Adressleitungen zu belassen.

Neben 7 Steckplätzen für den KONTRON-BUS habe ich meinem System noch 5 Plätze hinzugefügt, welche für das von ELEKTOR verwendete System eingerichtet sind.

### 2. Übergang N1 zum K.-BUS:

Hier gibt es zwei Möglichkeiten:

- a) Verwendung eines bereits vorhandenen Buffer-Boards mit Handverdrahtung zum K.-BUS.
- b) Verwendung des Taktes vom N1 Board, plus Aufbau eines eigenen Buffer Boards zum Aufstecken auf den K.-BUS.

Dieses Board sollte aufnehmen:

Adress- und Datenbuspuffer mit SN 74244/45      4 IC's

Resetimpulserzeugung

NASCOM-MEM und -IO Decodierung

evt. WAIT-Zyklus Erzeugung

Schaltpläne hierzu s.h. Literaturtip.

Beide Möglichkeiten wurden erfolgreich erprobt, wobei das B.Board auf einer Veroboard Platine handverdrahtet wurde. Ich könnte mir jedoch vorstellen, daß sich bei genügendem Interesse jemand finden ließe, der eine entsprechende Platine entwickelt.

### 3. Mechanischer Aufbau:

z.B. Europakartenträger (ca. 50 DM)

N1-Platine oben längs befestigen

4. Will man ebenfalls die ELEKTOR Ram/Rom (8k ca. 250,--DM) Karten verwenden, so muß man den Karten die Signale RD und IORQ zuführen. IORQ dient hier als "Ram disable" sonst kann man böse Überraschungen beim Kassettenbetrieb erleben.

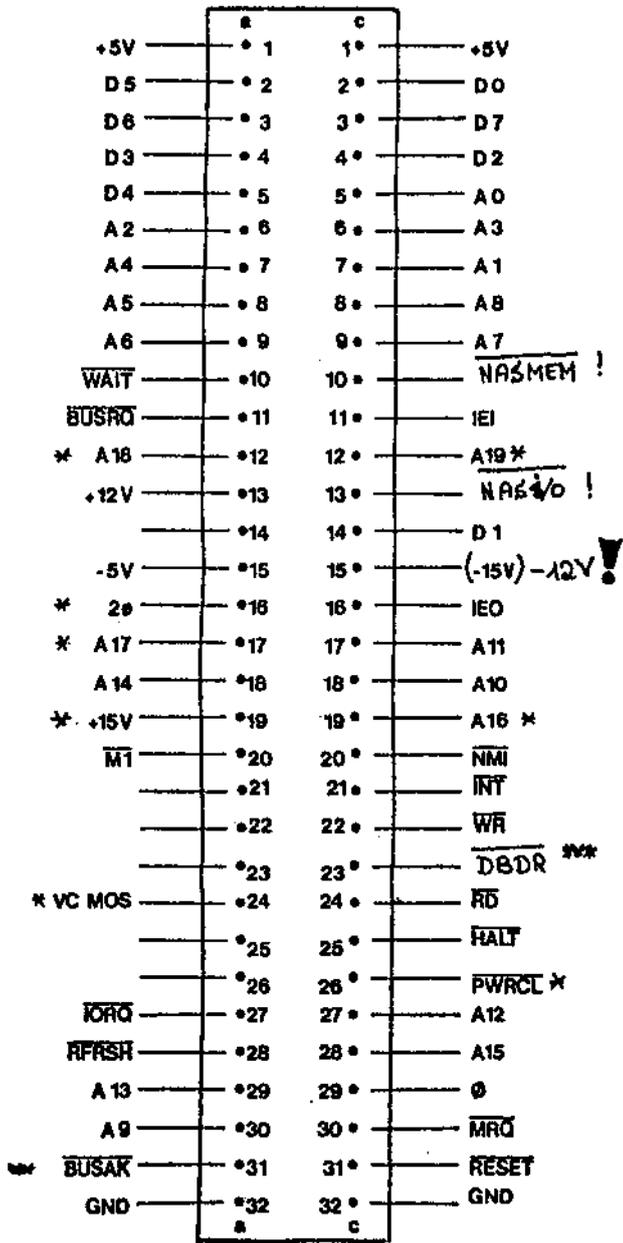
5. Da ich zahlreiche für das Betriebssystem T2 geschriebene Programme besitze (RTTY, DISASS) habe ich mir NASSYS in ein 2716 programmiert und eine Umschaltkarte NASSYS oder T2 + 1k Ram gebaut.

6. Positive Erfahrungen wurden in dieser Konfiguration gemacht mit:

- 8k Ram/Rom Karte nach ELEKTOR mit 2114 und Basic in 4x2716
- 32k dyn. Ram Karte aus ELZET-System
- 64k quasi stat. Ram Karte von Mikropoint (dyn. Ram mit RFH-Contr.)

7. Literaturtip:

Jedem der sich mit Z 80 Computern auch Hardware orientiert beschäftigen will, dem sei das Buch "Mikrocomputer Hard- und Software Praxis" von R.D. Klein empfohlen (Franzis ISBN 7723-6811-5).



BUS Belegung für ein angepasstes KONTRON/NASCOM System

- \* : beim NASCOM nicht verwendet.
- \*\* : keine DAYSY-CHAIN vorgesehen.
- \*\*\* : nur bei Verwendung des N.Bu.Boards nötig.
- ! : für NASCOM zusätzlich nötig.

Stefan Bürger  
 Frankfurt 71  
 Tel. [redacted]

## Anpassung des NASCOM Bussystemes an den ECB-Bus

Josef Zeller, [REDACTED], [REDACTED] Bayreuth, [REDACTED]

### Kurzbeschreibung:

Alle Z 80 Busleitungen (Daten- Adress-, Kontrollleitungen) sind entsprechend zu verbinden.

- 2) Das Steuersignal "DBDR", das die Bstreiber in die gewünschte Richtung schaltet (zur CPU hin, von der CPU weg) muss auf der Anpassungsplatine generiert werden.
- 3) Ueberlagern sich im NASCOM-System ein ROM-Bereich und ein RAM-Bereich am selben Adreßblock, so blendet der ROM-Bereich den RAM aus. Der ROM besitzt also Vorrang vor dem RAM.

Wenn sich ein ROM-Bereich auf einer ECB-Platine mit einem RAM auf einer NASCOM-Platine überlager, muss das Ausblendensignal "RAM Disable" auf der Anpassungsplatine erzeugt werden.

### 4) Realisierung

#### Zu 2) Umschalten der Treiber-IC

Ist im vorhandenen NASCOM-System schon eine Speicherkarte vorhanden, so sind bei der Lötbrücke J1 nur die Kontakte 1-2 zu verbinden. Der IC 1 ist dann ueberflüssig.

Ist keine Speichererweiterung vorhanden, so wird durch IC 1 der Speicherbereich auf der NASCOM 1 - Hauptplatine eingestellt. Dazu werden bei der Lötbrücke J 4 die Kontakte 1-2-3-4 entsprechend des gewünschten 4k Memory Blocks eingestellt. Normalerweise 1-2-3-4 gegen Masse legen. Ausserdem wird J 9 überbrückt. Bei der Lötbrücke J4 ist 1-4 und 2-3 zu verbinden und auf der NASCOM-Grundplatine die "MEM EXT"-Lötbrücke auf "EXT" zu stellen.

Ist im NASCOM-System kein I/O-Board vorhanden, muss IC 2 eingesetzt werden (ansonsten bei Lötbrücke J2 Kontakte 1-2 verbinden). Beim Jumper J2 ist 1-3 und 2-3 zu verbinden. Auf der NASCOM-Grundplatine ist die I/O-EXT. Lötbrücke auf EXT. einzustellen.

#### Zu 3) RAM-Ausblendung auf NASCOM-Platinen

Der eingestellte ROM-Bereich auf einer ECB-Platine wird auf der Anpassungsplatine ebenfalls eingestellt. Es sind zwei ROM-Bereiche mit je 4k Speicher selektierbar. Die Dekodierung der zwei ROM-Bereiche geschieht durch die IC 7 und IC 9.

Soll ein 4k Block ausgeblendet werden, so ist IC 7 einzusetzen und gleichzeitig ist der Jumper J6 zu überbrücken. Beim J5 wird der Anfang des ROM-Blockes durch die obersten vier Adressbit decodiert. Dazu sind die Lötkontakte 1-2-3-4 entsprechend gegen Masse oder +5 Volt zu legen.

Für den zweiten ROM-Block gilt das Gleiche. IC 8 einsetzen und die Lötbrücken J7 und J8 müssen entsprechend verbunden werden.

Diese Schaltung wurde für NASCOM 1 entworfen. Ich habe noch nicht überlegt, wie sie für den NASCOM " aussehen würde.

J. Zeller



## ERWEITERUNG DES NASCOM AUF KONTRON BUS

Gerade jetzt, wo sich die Firma NASCOM im Umbruch befindet und niemand weiss, wie die zukünftige Liefersituation von NASCOM Ersatzteilen und Erweiterungen aussieht, empfiehlt es sich, die Möglichkeit eines Anschlusses des Nascom an ein anderes Bussystem zu überdenken.

Überall werden einem zur Zeit preisgünstige Platinen für den ELZET 80 oder KONTRON Bus angeboten, seien es RAM, EPROM oder I/O Platinen, so dass eine Erweiterung durch dieses System günstig erscheint.

Seit einiger Zeit arbeitet mein NASCOM 1 nun erfolgreich mit einer KONTRON 16k Eprom-Karte und einer RAM Erweiterung von 4k, die ich demnächst noch zu vergrössern gedenke. In die Epromkarte habe ich die 8k Basic des NASCOM eingesteckt, die bis jetzt auch einwandfrei arbeiten.

Die Erweiterung war nicht weiters schwierig: Eine Busplatine für ca. 10 Platinensteckplätze ist ohne weiteres zu haben, ein Netzgerät für die erforderlichen Spannungen ist auch schnell selbst gestrickt, bleibt nur noch ein Adapter, der den NASCOM Bus auf ein anderes Steckersystem umlegt. Diesen Adapter möchte ich nun hier vorstellen.

Bild 1 zeigt die Platinenanschlüsse des NASCOM, an die ich einfach ein mehrfarbiges 43poliges Flachbandkabel angelötet habe, und die Anschlüsse des 64poligen KONTRON Steckers.

Man sieht, dass nur die Adressleitungen und  $\overline{RD}$  bzw.  $\overline{MRQ}$  Signale gepuffert sind, die Datenleitungen und übrigen Steuersignale gehen direkt an den KONTRON Bus, d.h. die Litzen werden direkt an die Steckeranschlüsse angelötet.

Um eine doppelseitige Kaschierung zu vermeiden, wurden auch die Litzen für die Puffereingänge direkt an die IC Anschlüsse (74367) angelötet.

Die Adressen 12-15 werden invertiert und über einen Nor-Gatter an den Memory Select Eingang zurückgeleitet. (Die Brücke auf der NASCOM Platine muss natürlich auf Mem.Ext. umgelötet werden). So werden die RAMs auf der Platine nur angesprochen, wenn die oberen 4 Adressen  $\emptyset$  sind. ( $\emptyset \emptyset \psi \emptyset$  bis C F F )

Bild 2 zeigt die Bestückungsseite und die Leiterbahnseite der Adapterplatine. Vorsicht ist geboten bei den Leitungen, die genau durch den 64pol. Stecker gehen, hier gibt es beim Löten leicht Kurzschlüsse.

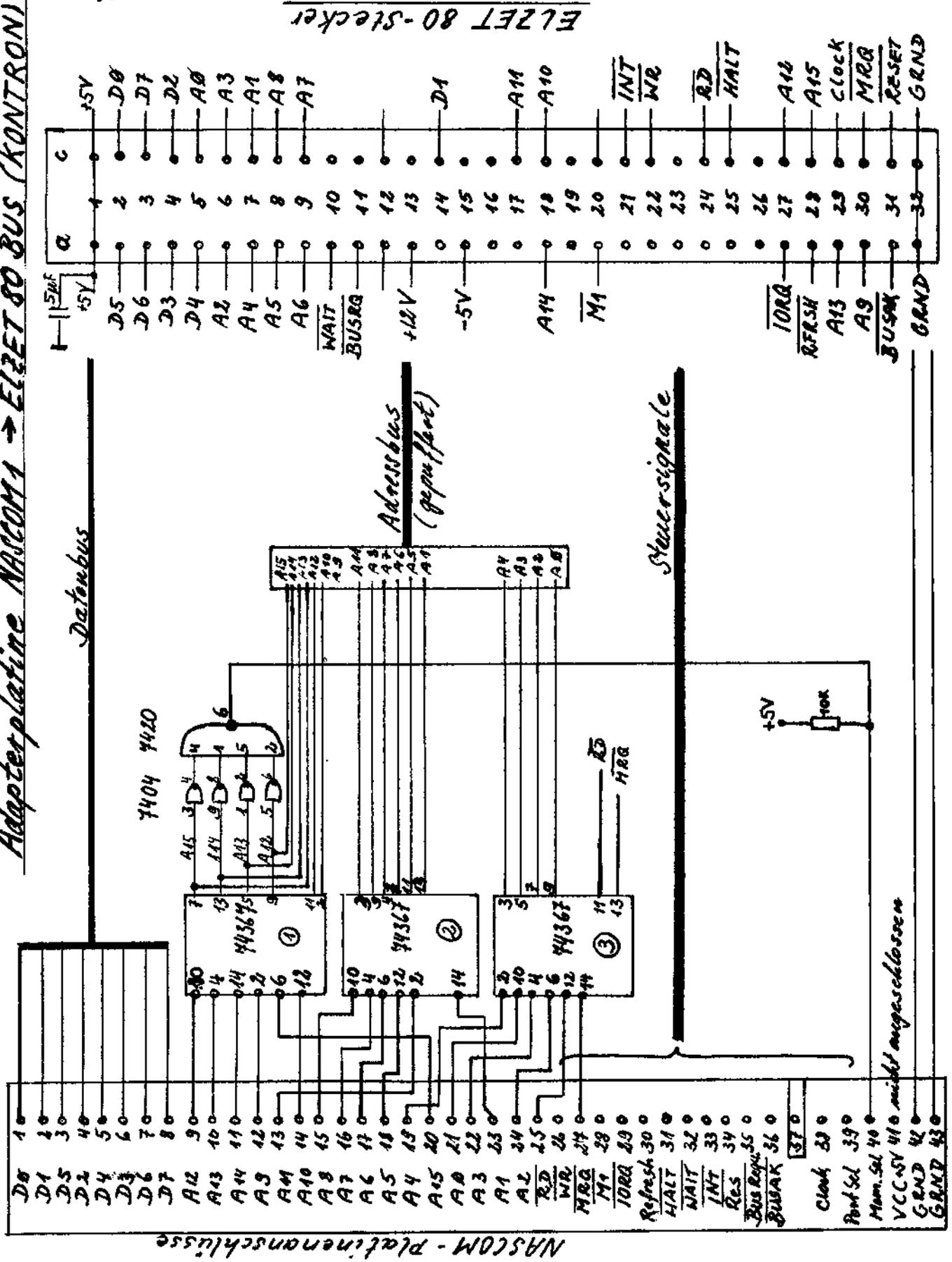
Über Ihre Erfahrung mit KONTRON/ELZET 80 Erweiterungen wäre ich sehr gerne informiert. So schreiben Sie mir doch bitte oder rufen Sie an, wenn Sie an einer Erweiterung selbst herumbasteln. Oft sind es nämlich Kleinigkeiten, von denen das Funktionieren abhängt, und die Erfahrung verschiedener Praktiker hilft einem oft weiter als jede Theorie. Bei meiner Epromkarte mussten z.B. die Busleitungstreiber einfach entfernt bzw. überbrückt werden, bevor die Karte funktionierte, nachdem ich tagelang mit Verzögerungen der verschiedenen Steuersignale experimentiert hatte. Warum einfach, wenn es auch kompliziert geht? (Oder umgekehrt?)

PS Die Bahnen der gepufferten  $\overline{RD}$ ,  $\overline{MRQ}$  Signale sind auf der Platine vergessen worden. Von den Ausgängen 11 und 13 des ICs 74367 3 sind also Leitungen zu den Steckeranschlüssen 24c und 30c zu führen.

# Adapterplatine NASCOM 1 → ELZET 80 BUS (KONTRON)

Spannungsversorgung  
 +5V  
 -5V  
 +12V  
 durch eigenes Netzteil

Bild 1



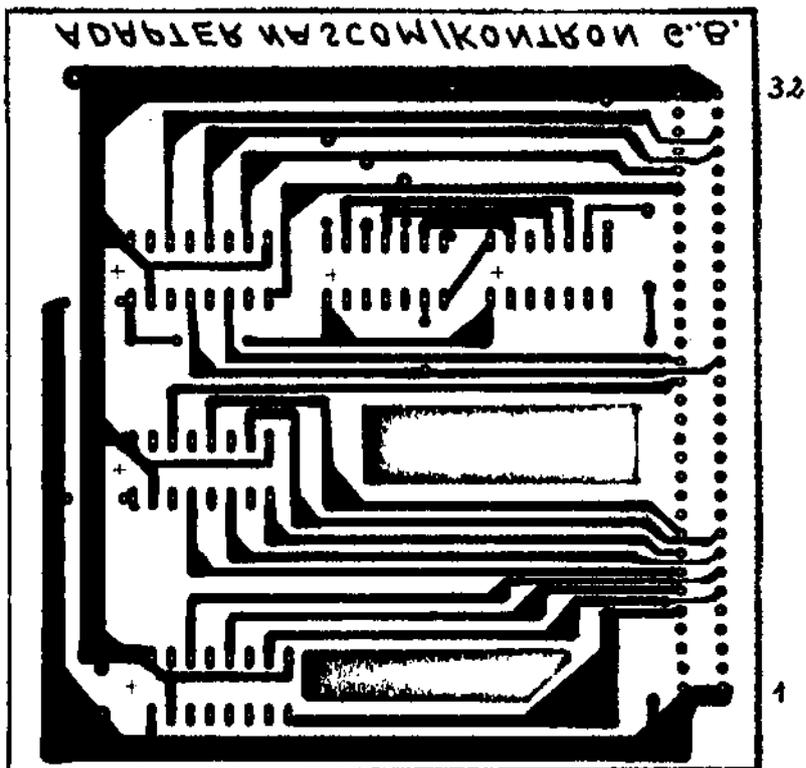
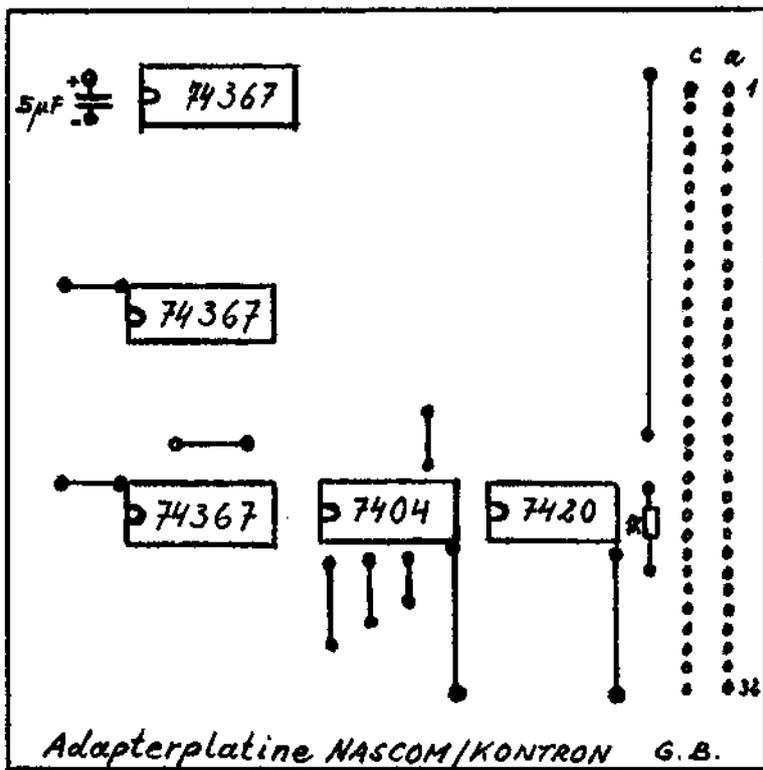


Bild 2

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

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

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

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

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

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

## 2. Laufwerk 880.- + MWSt.

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

## EPR0M-Programmiergerät für NASCOM 1 und NASCOM 2

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

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

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

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

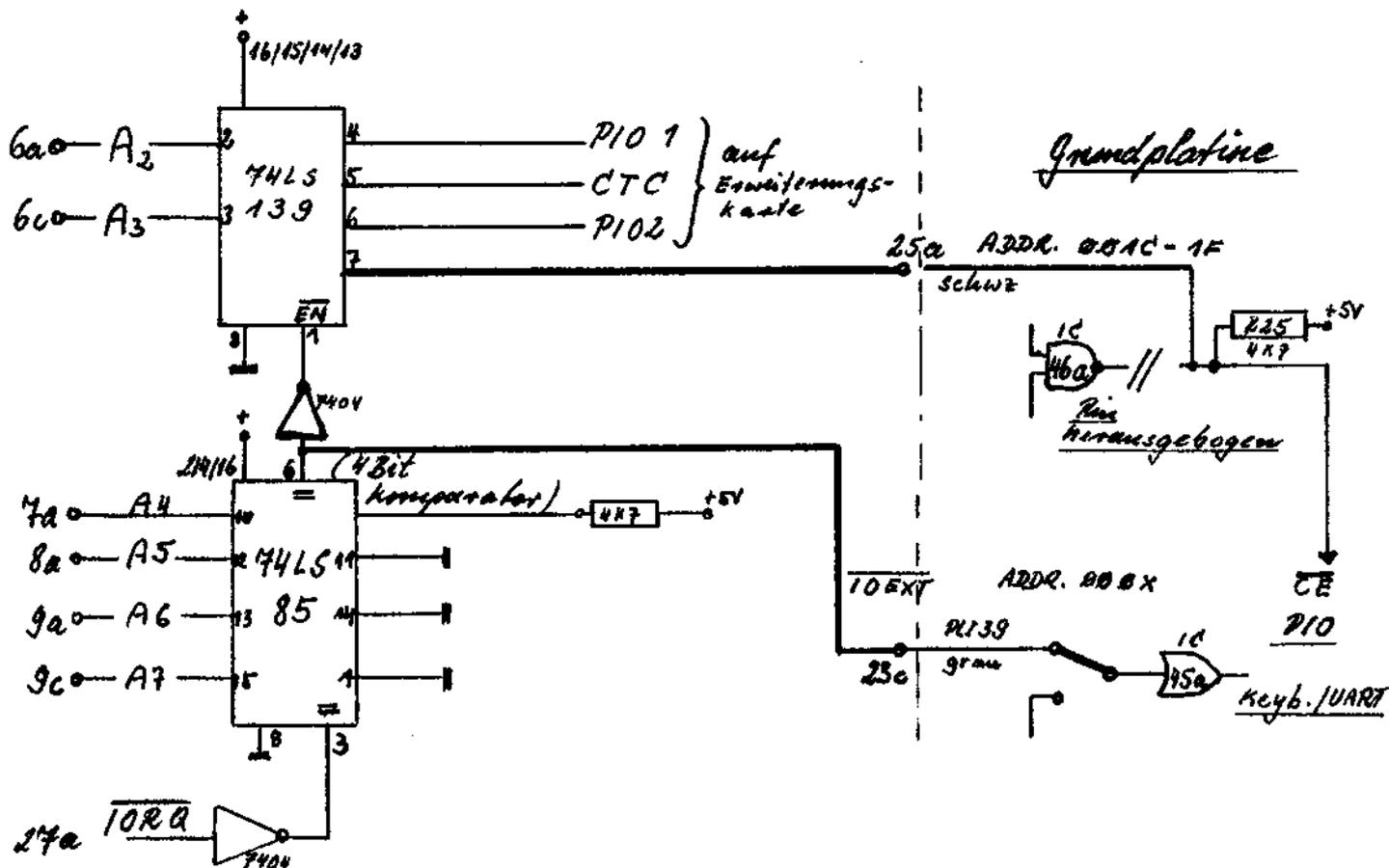
Bausatz: 168.- + MWSt; Typ des Rechnersystemes bzw. Betriebssystem angeben.

Wandler auf der Karte |

## Anschluss einer KONTRON/ELZET 80 I/O Karte an NASCOM 1

Bei Verwendung der beschriebenen Adaptorkarte fällt es nicht schwer, auch eine IN/OUT Platine (die fertig im Handel erhältlich ist) an das NASCOM System anzuschliessen. Die beschriebene Dekodierung, deren Hardware bereits auf der I/O Platine vorhanden ist und nur geringfügig geändert werden muss, kann auch als Anregung dienen, um eine selbstgestrickte Erweiterungskarte zu dekodieren oder eine bereits vorhandene NASCOM I/O Karte so zu verändern, dass die auf der Grundplatine liegende PIO weiter verwendet werden kann, was meines Wissens bei einer normalen Erweiterung nicht möglich ist. (Das wäre z.B. bei meinem System schade, da ich einen praktischen I/O Stecker im Pultgehäuse der Tastatur untergebracht habe, der dann nicht mehr verwendbar wäre) Die Funktion der Dekodierung ist ohne weiteres aus dem Schaltbild zu entnehmen. Hardwareänderungen auf der Basisplatine des NASCOM sind nur das Herausbiegen eines IC Beinchen und das Einlöten einer Leitung an einem Lötauge direkt links von IC 46, das von der Platinenoberseite gut zugänglich ist. (Ursprünglicher Anschluss von Pin 3 IC 46a)

Decodierung Kontron Plokarte / interne Ports



Adresse A4 0 Keyb./UART  
Adresse A4 H Erw. Platine / ind. Pio  
Adresse A5/A6/A7 uninteressant (0)

PORT ADRESSEN

00	Keyboard	18	Pio 2 A Data
01	UART Data	19	Pio 2 B Data
02	UART Status	1A	Pio 2 A Control
03		1B	Pio 2 B Control
-0F	frei	1C	Pio Basisplatine A Data
10	Pio 1 B Data	1D	" " B Data
11	Pio 1 A Data	1E	" " A Control
12	Pio 1 B Control	1F	" " B Control
13	Pio 1 A Control		
14			
-17	CTC		

In vielen Programmen ergibt sich die Notwendigkeit, Werte zu berechnen. Hierbei fällt auf, dass der Monitor des NASCOM als Rechner eine absolute Niete ist; ausser Addition und Subtraktion geht nichts mehr. So habe ich einmal eine Reihe von Unterprogrammen erstellt, die frei verschiebbar sind und in Maschinenprogrammen als einfache Rechenhilfe dienen.

Alle Rechengänge laufen im Hexadezimalsystem ab, können aber durch entsprechende Unterprogramme dezimal von der Tastatur eingegeben oder dezimal auf dem Bildschirm dargestellt werden. Ausgangswert und Ergebnis stehen jeweils in HL, unbeteiligte Register werden gerettet, so dass Push und Pop Befehle vor und nach Aufruf der Unterprogramme entfallen.

- OC80 Eine Zahl in HL wird mit einer Zahl in B multipliziert. (die Möglichkeit der Multiplikation mit einer mehr als zweistelligen Zahl ist durch die beschränkte Kapazität des 8 Bit Systems nicht gegeben)  
Bei Überschreiten der Rechenkapazität erfolgt ein relativer Sprung zur Fehleranzeige, der im Fall einer Verschiebung des Programms in der Adresse OC8E geändert werden müsste.
- OC95 Eine Zahl in HL wird durch eine Zahl in BC dividiert. Enthält BC  $\emptyset\emptyset$ , so rechnet das Programm bis in alle Ewigkeit. Man müsste hier eine Erkennung auf BC= $\emptyset$  einbauen (z.B. LD A,B OR C JRZ ENDE) oder einfach Divisionen durch  $\emptyset$  vermeiden.
- OCA5 Es wird die Quadratwurzel einer Zahl in HL errechnet. Diese Zahl darf nicht grösser als  $181^2$  dez sein, da sonst der Bereich der positiven Zahlen überschritten wäre. (32676 ist max.Radikant) Da Kommazahlen im Ergebnis nicht eingeplant sind, erübrigt es sich, die Schleifenzahl 10 zu erhöhen, um genauere Näherungswerte zu erhalten. Das Ergebnis des grössten Radikanten ist nach zehn Durchgängen auf 1 Zehntel genau. Ein Nachteil ist, dass die Wurzel aus 9 bis 15 jeweils 3 ergibt und erst ab 16 auf 4 steigt usw. Bei mir wurde das Programm zum Ansteuern von Schrittmotoren verwendet. Dabei kann ich mit Bruchteilen von Schritten nichts anfangen. Wer aber Wert auf genauere Ergebnisse legt, kann ja anstatt z.B. 5 dez 500 dez eingeben und erhält dann ein Ergebnis auf ein Hundertstel genau, wobei er sich nur ein Komma hinzudenken müsste. Dieser letzte Mangel kann sogar noch leicht im Anzeigeprogramm behoben werden, wenn man nach der Hunderter ein Komma einprogrammiert. Allerdings wird dann der Zahlenbereich nach oben auf 99,99 beschränkt. Das Wurzelprogramm enthält einen Aufruf des Divisionsprogramms bei OCB6. Dieser muss bei Verschiebung im Speicher geändert werden.
- OCC1 Fehlermeldung bei Bereichsüberschreitung und Sprung zu Monitor
- OCD5 Eine Dezimalzahl in HL wird in Hex umgewandelt. Das Ergebnis steht in HL. Register DE und B werden verändert und müssen vor Aufruf gerettet werden. Bei Verschieben beachten: Aufruf des Unterprogramms Multiplizieren bei OCE2, OCEF, ODO2 .
- OD10 Hilfsprogramm zu OD25
- OD25 Eine Hexzahl in HL wird fünfstellig dezimal auf Bildschirm ausgegeben. AF, BC und DE müssen vorher gerettet werden. Das Programm enthält Aufrufe von Multiplizieren und Dividieren OD11, OD1D, und Aufrufe des Hilfsprogramms in OD28, OD2E, OD34 und OD3A; bei Verschieben beachten.



subroutine 'multiplizieren'	("HL" mal "B")
0c80 push af	Register retten
0c81 push de	wenn B=∅ Rücksprung mit
0c82 push bc	Ergebnis ∅∅ ∅∅
0c83 ld a,00	Zahl nach DE
0c85 cp a,b	Zahl so lange zu ∅ addieren...
0c86 ex de,hl	bei Überlauf: Sprung "Out of Range"
0c87 ld hl,0000	... bis B=∅
0c8a jrz +07 hex	Register restaurieren
0c8c add hl,de	Return (Ergebnis in HL; andere
0c8d jrc +34 hex	Register unbeeinflusst)
0c8f djnz -03 hex	
0c91 pop bc	
0c92 pop de	
0c93 pop af	
0c94 ret	

subroutine 'dividieren'	("HL" durch "BC") durch ∅ nicht
0c95 push af	möglich
0c96 push de	Register retten
0c97 ld de,0000	Ergebnisreg. auf ∅
0c9a sub hl,bc	BC von HL subtrahieren bis
0c9c jrc +05 hex	Überlauf: Rücksprung
0c9e inc de	Ergebnis +1
0c9f jr -05 hex	Schleife bis Überlauf
0ca1 ex de,hl	Ergebnis nach HL
0ca2 pop de	Reg. restaurieren
0ca3 pop af	(Ergebnis in HL; andere Register
0ca4 ret	unbeeinflusst)

subroutine 'quadratwurzel'	(max. 181 <sup>2</sup> dez.)
0ca5 push af	Register retten
0ca6 push bc	Schleifenzähler (10 Näherungs-
0ca7 push de	werte)
0ca8 ld e,0b	HL div. durch 2 = x <sub>0</sub>
0caa push hl	Schleifenzähler herunterzählen
0cab srl, n	nach 10 Durchgängen Sprung Ende
0cad r r, l	x <sub>0</sub> nach BC
0caf dec e	Ausgangszahl a
0cb0 jrz +0c hex	Call Dividieren (a:x <sub>0</sub> )
0cb2 push hl	Ergebnis + x <sub>0</sub>
0cb3 pop bc	Sprung Anfang Schleife
0cb4 pop hl	Ausgangszahl wegwerfen
0cb5 push hl	Reg. restaurieren
0cb6 call 0c95	Return (Ergebnis in HL; andere
0cb9 add hl,bc	Register unbeeinflusst)
0cba jr -0f hex	
0cbc pop bc	
0cbd pop de	
0cbe pop bc	
0cbf pop af	
0cc0 ret	

subroutine fehlermeldung 'out of range'	
0cc1 print:	Clear Screen
out of range .	Print Text
0cd1 nop	
0cd2 jp 0359	Jump Monitor

subroutine 'dezimal in hex umwandeln'

```

0cd5    push hl
0cd6    □srl,h          Tausender auf 1.Stelle in H
0cd8    srl,h
0cda    srl,h
0cdc    srl,h
0cde    ldb,h         Tausender inB
0cd2    ld hl, 03e8   LD HL , 1000(hex)

0ce2    call 0c80     Call Multipliz.
0ce5    ex  de,hl    Tausender in DE
0ce6    pop  hl
0ce7    push hl      Ausgangszahl wieder retten
0ce8    ld   a,h
0ce9    and  a,0f     Tausender ausklammern
0ceb    ld   b,a      Hunderter nach B
0cec    ld   hl,0064  LD HL, 100(hex)
0cef    call 0c80     Call Multipliz.
0cf2    add  hl,de    Hunderter zu Tausendern addieren
0cf3    ex   de,hl    beide nach DE
0cf4    pop  hl
0cf5    push hl      Ausgangszahl
0cf6    □srl,l       Zehner auf 1. Stelle in L
0cf8    srl,l
0cfa    srl,l
0cfc    srl,l
0cfe    ld   b,l       Zehner in B
0cff    ld   hl,000a   LD HL, 10 (hex)
0d02    call 0c80     Call Multipliz.
0d05    add  hl,de    Zehner addieren
0d06    ex   de,hl    1000 100 u. 10 in DE
0d07    pop  hl      Ausgangszahl
0d08    ld   a,l       Zehner in L ausklammern
0d09    and  a,0f
0d0b    ld   l,a
0d0c    ld   h,00     LD H, 00
0d0e    add  hl,de    Vorherige Ergebnisse zu Einern addieren
0d0f    ret          RETURN (Ergebnis in HL)

```

subroutine 'dezimale anzeige einer hex zahl'

```

0d10    push hl      Subroutine
0d11    call 0c95     Call Dividieren
0d14    ld   a,l
0d15    push bc
0d16    pop  hl
0d17    ld   b,a
0d18    add  a,30
0d1a    call 013b     Display A ASCII
0d1d    call 0c80     Call Multipliz.
0d20    ex   de,hl
0d21    pop  hl
0d22    □subc hl,de
0d24    ret
0d25    ld   bc,2710  START LD BC, 10 000
0e28    call 0d10     Call Subroutine
0d2b    ld   bc,03e8  LD BC, 1000
0d2e    call 0d10     Call Subroutine
0d31    ld   bc,0064  LD BC, 100
0d34    call 0d10     Call Subroutine
0d37    le   bc,000a  LD BC, 10
0d3a    call 0e10     Call Subroutine
0d3d    ld   a,l       LD A, Einer
0d3e    add  a,30     add 30 (ASCII Format)
0d40    call 013b     Display A ASCII
0d43    ret          RETURN

```

Wie oft sind Zehner, Hunderter, Tausender und Zehntausender in der Hexzahl enthalten? Die jeweiligen Ergebnisse werden addiert und in DE abgelegt.

Einer werden zum Ergebnis in HL addiert.

hauptprogramm 'rechner'

Od44	ld hl,(0c0c)	Subroutine: Arg.1 hex in HL und
Od47	call 0cd5	Arg.2 hex in B
Od4a	push hl	(Arg.2 nur zweistellig!)
Od4b	ld hl,(0c0e)	Call Dez./Hex.
Od4e	call 0cd5	
Od51	ld b,l	
Od52	pop hl	
Od53	ret	
<hr/>		
Od54	print:	<u>Programmstart:</u>
	6eingabe h v d b	"Eingabe Hex Dez ?"
	?	
Od67	nop	
Od68	call 0069	keyboard
Od6b	jnc -03 hex	Warteschleife
Od6d	cp a,48	"H" gedrückt? (Soll Eingabe in Dezimal
Od6f	jrnz +0c hex	erfolgen?)
Od71	ld (0bca),a	ja: Anzeige "H"
Od74	ld a,c9	Subroutine Dez./Hex Beginn mit Return laden
Od76	ld (0cd5),a	(Subr. wird nicht ausgeführt)
Od79	jr +09 hex	"D" gedrückt?
Od7b	cp a,44	beliebige Taste: zurück zu keyboard
Od7e	jrnz -15 hex	"D" gedrückt! Call "Dez/Hex aktivieren"
Od7f	call 0e1f	Anzeige
Od82	ld a,09	
Od84	ld (0bcb),a	
Od87	print:	(Soll Ausgabe in Dezimal erfolgen?)
	ausgabe ?	
Od92	nop	
Od93	call 0069	keyboard
Od96	jrnc -03 hex	ist"H"?
Od98	cp a,48	
Od9a	jrnz +13 hex	ja: "H" anzeigen
Od9c	ld (0bcc),a	Anfang von Subr.Hex/Dez mit CD 32 02 C9
Od9f	ld hl,32cd	laden(erwirkt Call der Monitorrout.
Oda2	ld (0d25),hl	"HL als hex darstellen"anstelle von
Oda5	ld hl,c902	"dez.Darstellung...")
Oda8	ld (0d27),hl	
Odab	jr +09 hex	ist"D"?
Odad	cp a,44	beliebige Taste: jump keyboard
Odaf	jrnz -1c hex	ja: Call:
Odb1	call 0e28	"Subrout. "Hex/Dez wieder herstellen "
Odb4	ld hl,0dbd start2	Neue Control Tabelle bei ODBD
Odb7	ld (0c45),hl	Call INLINE (Read a line and execute)
Odba	call 0286	Neue Kontrolltabelle:
Odbd	m CF0d	Multiplizieren = M jp 0DCf
Odc0	d db0d	Dividieren = D jp 0DDB
Odc3	q f40d	Quadrieren = Q jp 0DF4
Odc6	w 040e	Wurzel = W jp 0E04
Odc9	u 130e	Umwandeln = U jp 0E13
Odcc	0 0000	

```

Odcf  call 0d44  MULT.  Call Sub. "Arg. in HL und B"

Odd2  call 0c80  Call "Multipl."
Odd5  call 0d25  Call "Anzeige"
Odd8  jp 0db4  Jump Start 2

Oddb  ld hl,(0c0c) DIV.  Arg.1 in HL
Odde  call 0cd5  Call Dez/Hex
Ode1  push hl
Ode2  ld hl,(0c0e)  Arg.2 in HL
Ode5  call 0cd5  Call Dez/Hex
Ode8  push hl  Arg.2 in BC
Ode9  pop bc
Odea  pop hl
Odeb  call 0c95  Call Dividieren
Odee  call 0d25  Call Anzeige
Odf1  jp 0db4  Jump Start 2
Odf4  ld hl,(0c0c) QUAD. Arg.1 in HL
Odf7  call 0cd5  Call Dez/Hex
Odfa  ld b,l  HL darf nur zweistellig sein
Odfb  call 0c80  Call Multiplizieren
Odfe  call 0d25  Call Anzeige
Oe01  jp 0db4  Jump Start 2
Oe04  ld hl,(0c0c) 2√ Arg.1 in HL
Oe07  call 0cd5  Call Dez/Hex
Oe0a  call 0ca5  Call wurzel
Oe0e  call 0d25  Call Anzeige
Oe10  jp 0db4  Jump Start 2
Oe13  ld hl,(0c0c) UMWA! Arg.1 in HL
Oe16  call 0cd5  Call Dez/Hex
Oe19  call 0d25  Call Anzeige
Oe1c  jp 0db4  Jump Start 2
Oe1f  ld (0bca),a  Subrout. Dez/Hex RESTAURIEREN
Oe22  ld a,e5  (Beginn der Subr. wird mit E5 geladen)
Oe24  ld (0cd5),a
Oe27  ret
Oe28  ld (0bcc),a  Subrout. Sub.Hex/Dez RESTAURIEREN
Oe2b  ld hl,1001  (Beginn der Subr. wird mit 01 10 27 CD
Oe2e  ld (0e25),hl  geladen)
Oe31  le hl,cd27
Oe34  ld (0d27),hl
Oe37  ret

```

Günter Böhm

Karlsruhe

## Integer Pascal Compiler für den Nascom

Mittlerweile gibt es zahlreiche Basic Interpreter für die Nascom Computer, hier wird zum ersten Mal ein Pascal Compiler vorgestellt.

Implementiert ist praktisch das gesamte Standard Pascal, mit einer einzigen Ausnahme, wie der Name beinhaltet kann der Compiler nur ganze Zahlen im Bereich -32767 bis +32767 verarbeiten (HEX-Integers zwischen 0000 und FFFF'H).

Trotz dieser Einschränkung eignet sich dieses Pascal aber besonders gut zum Lernen (bis es mit CP/M auf dem Nascom die großen Compiler gibt). Der niedrige Preis ermöglicht es sparsamen Computerfans auch ohne Anschaffung eines Floppy-Systems diese schnelle Alternative zu BASIC zu benutzen. Bei 4MHz CPU Takt besiegte dieses Pascal sogar das APPLE-PASCAL !

Das gesamte Programmpaket ist genau 8,5K lang, benötigt mindestens 16K und läuft auf NASSYS1. Es besteht aus einem Command Processor, einem Editor, einem 2-pass Compiler und einigen Runtime Routinen. Insgesamt gibt es 27 (!) Fehlermeldungen die aus Speicherplatzgründen wie bei ZEAP nummerncodiert ausgegeben werden. Die Arbeit mit dem Editor wird durch Auto Repeat sehr komfortabel.

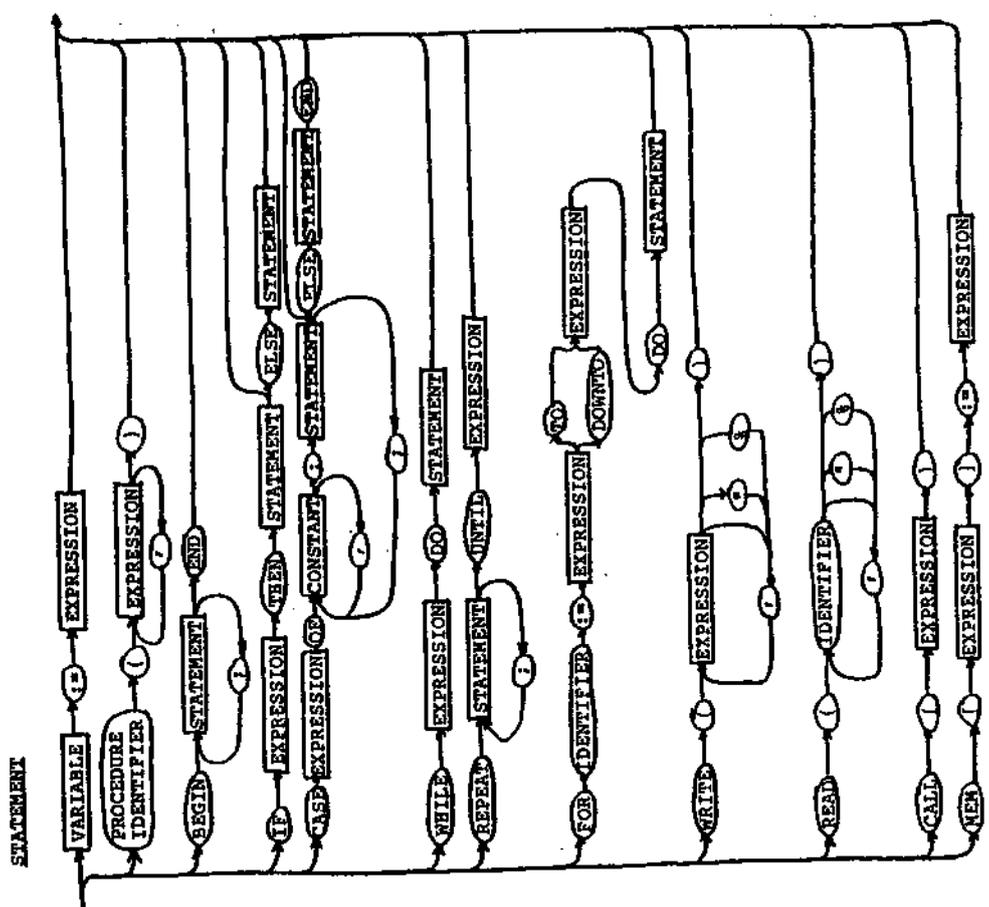
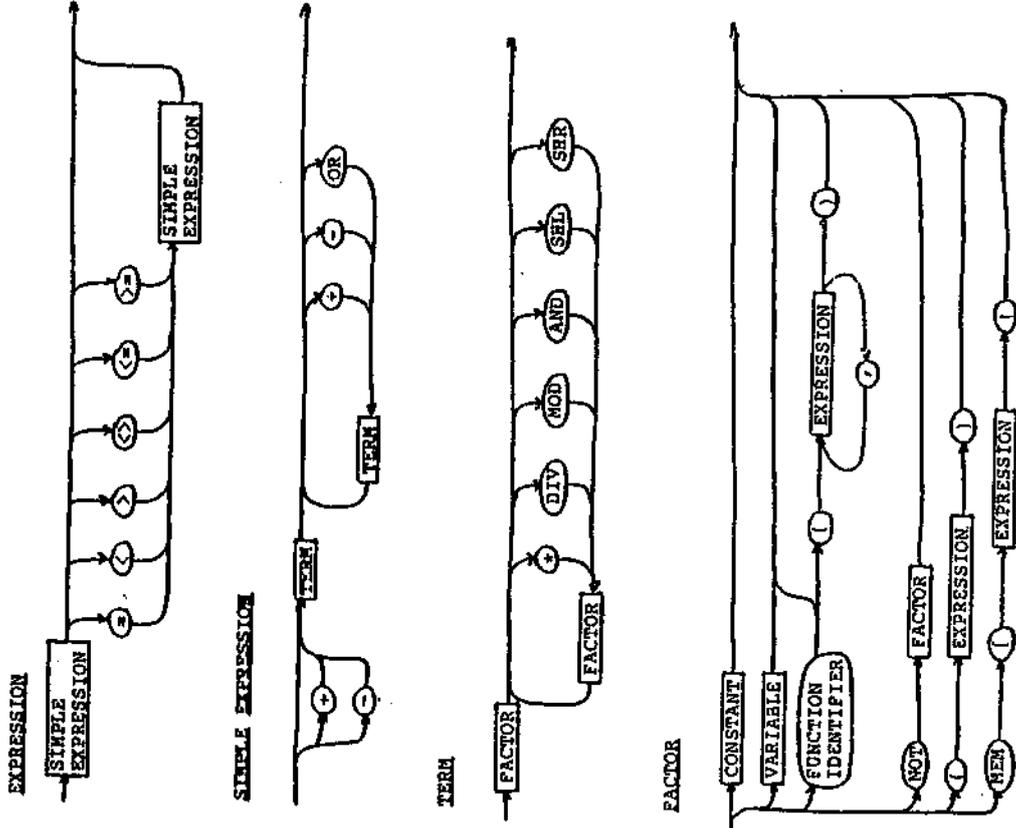
Im Vergleich zum Standard Pascal gibt es mehrere Erweiterungen, z.B. "PEEK" und "POKE", SHL und SHR (shiftbefehle), ELSE und andere. Bei Ein- und Ausgabe kann zwischen hexadezimal, dezimal und ASCII gewählt werden.

Der Editor bietet folgende Kommandos :

```
AMMMM DATA : Füge Source Zeile hinzu
B           : Gehe zum Source Anfang
C           : überprüfe Syntax
CMMMM      : Compiliere und lege Objekt ab Adresse MMMM ab
DMMMM      : Lösche Zeile Nummer MMMM
P           : Gebe Source Text auf Printer aus
RNNNN DATA : Ersetze oder verbessere Zeile MMMM
S           : Schreibe Source auf Tanband
T           : Springe zu NASSYS
Z           : Lösche Source Buffer
LMMMM      : gebe Zeile MMMM auf Bildschirm aus
NMMMM      : nummeriere mit Abstand MMMM durch
```

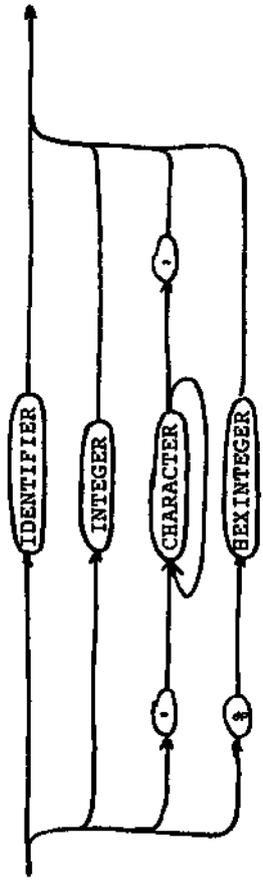
Die nachfolgenden Diagramme erläutern die verfügbare Syntax und den Source Aufbau. Das abgedruckte Beispielprogramm liest Zahlen in ein ARRAY ein, sortiert sie nach der "Quicksort"-Methode und gibt sie in Achter-Gruppen aus. man beachte besonders die rekursiven Aufrufe !!.

In den Diagrammen brauchen rechteckig eingerahmte Punkte eine weitere Definition, runde keine mehr.





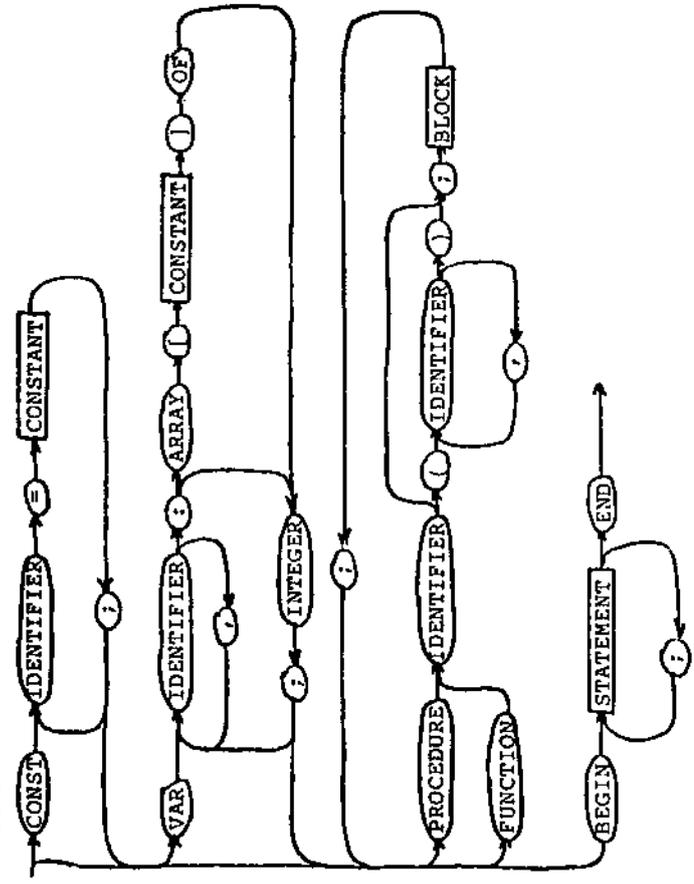
CONSTANT



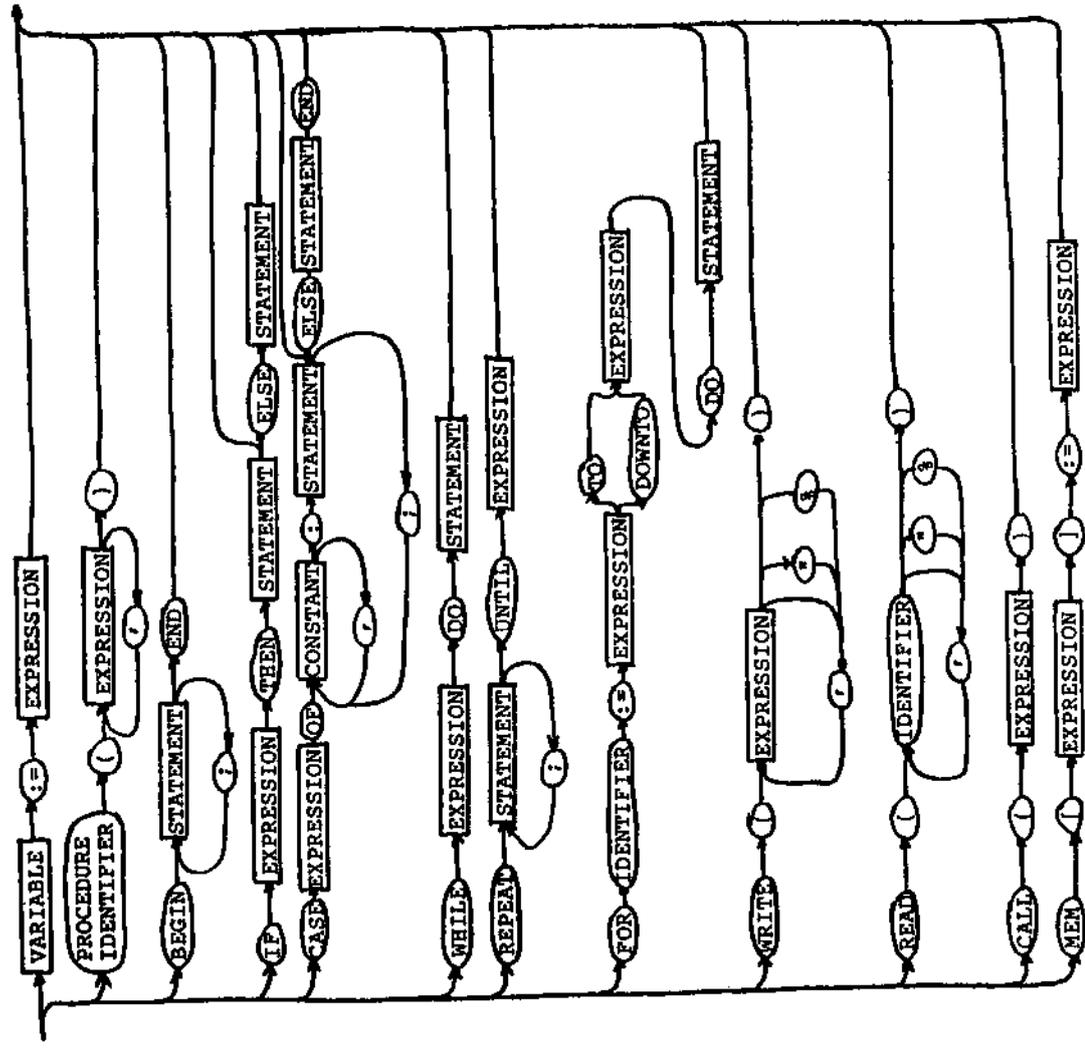
VARIABLE



BLOCK



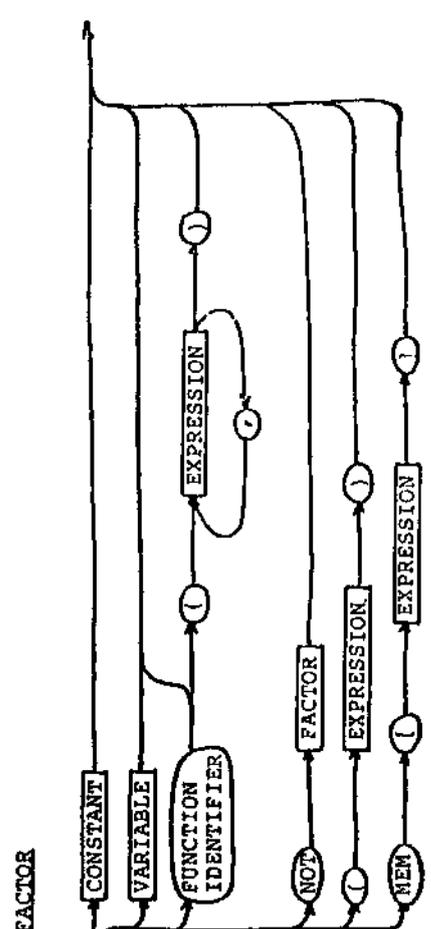
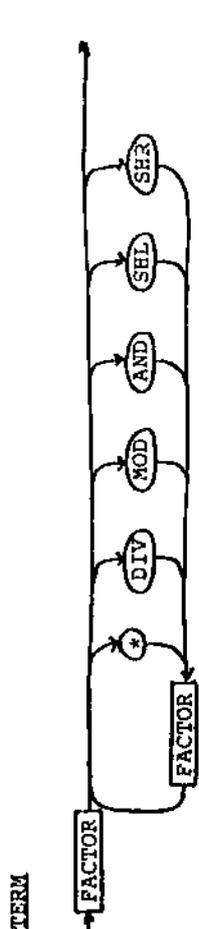
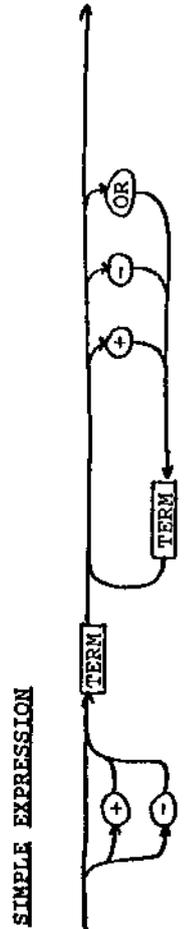
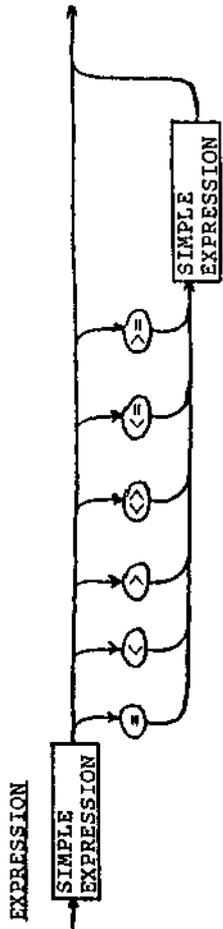
STATEMENT



```

CONST NL =%lf; {%0D for NAS-SYS}
VAR X : ARRAY [100] OF INTEGER;
Y,Z : INTEGER;
PROCEDURE SORT (N);
VAR A, B, C, D : INTEGER;
BEGIN
  A:=LEFT; B:=RIGHT;
  C:=X[(LEFT + RIGHT) DIV 2];
  REPEAT
    WHILE X[A]<C DO A:=A+1;
    WHILE C<X[B] DO B:=B-1;
    IF A<=B THEN
      BEGIN
        D:=X[A]; X[A]:=X[B]; X[B]:=D;
        A:=A+1; B:=B-1;
      END
    UNTIL A>B;
  IF LEFT<B THEN SRT(LEFT,B);
  IF A<RIGHT THEN SRT(A,RIGHT)
  END;
END;
SRT(1,N)
END;
PROCEDURE NEWLINE(X);
BEGIN
  IF X MOD 8 = 0
  THEN WRITE (NL"");
  ELSE WRITE (' ');
  END;
BEGIN
  WRITE('Enter values: ',NL"");
  Y:=0;
  REPEAT
    Y:=Y+1;
    READ(X{Y}); NEWLINE(Y)
  UNTIL X{Y}=0;
  SORT(Y-1);
  WRITE(NL"sorted values are: ',NL"");
  FOR Z:=1 TO Y-1 DO
    BEGIN
      WRITE (X{Z}); NEWLINE (Z)
    END
  END.

```



Albert Schunck  
München

## das programm tty-sys

eine beschreibung dieses programms hatte ich ja schon lange angekuendigt - im heft 4/80, aber dann kam immer etwas dazwischen und die veroeffentlichung hat sich bis jetzt verzoeigert. jetzt soll das programm, das schon einige zeit lieferbar ist ( fuer 98.- dm + mwst), aber auch im nascom-journal vorgestellt werden.

das programmsystem tty-sys ist die software-schnittstelle zum anschluss eines 5-kanal-fernsehreibers an einen nascom mit dem betriebssystem nas-sys. es enthaelt die routinen zur zeichenausgabe an den fernschreiber, zur zeicheneingabe vom fernschreiber, zum stanzen und lesen von lochstreifen, zum drucken von hex-listings und zum ausdrucken des bildschirminhalts. die baudrate des fernsehreibers kann softwaremaessig zwischen 45, 50, 75 und 100 umgeschaltet werden. die druckausgabe kann beliebig formatiert werden (zahl der zeichen pro zeile und zahl der zeilen pro seite). die ascii-zeichen, die im baudot-code nicht enthalten sind, werden als doppelzeichen ausgegeben. beim einlesen werden die doppelzeichen wieder erkannt, so koennen alle ascii-zeichen (von 20h bis 5fh) eindeutig ausgegeben und eingelesen werden.

tty-sys wird in einem eprom 2708 mit der anfangsadresse b000 geladert (andere anfangsadressen auf anfrage). es ist zusammen mit dem unveraenderten nas-sys verwendbar. die bedienung wird jedoch komfortabler, wenn in der befehltabelle von nas-sys bei dem befehl 'p' die adresse der interpretieroutine von tty-sys (b027) eingetragen wird. durch diese modifikation im zweiten eprom bleibt nas-sys auch ohne tty-sys weiter verwendbar.

tty-sys verwendet die speicherzellen Obfb-Obff (im unrichtbaren bildschirmbereich) als variabelnspeicher. weiterer arbeitsspeicher wird nicht benoetigt.

tty-sys ist standardmaessig fuer eine taktfrequenz des rechners von 2 mhz ausgelegt. soll es mit 4 mhz laufen, muss dies bei der bestellung angegeben werden.

tty-sys erweitert nas-sys um folgende befehle:

p0 a b c (e b000 a b c)

dieser befehl initialisiert ein- und ausgabe ueber den fernschreiber. a ist die zahl der zeilen pro seite (a=0: standardwert 66, a=ff: keine seiteneinteilung), b ist die zahl der zeichen pro zeile (b=0: standardwert 69) und c gibt die baudrate an (c=0: 45 baud, c=1: standardwert 50 baud, c=2: 75 baud und c=3: 100 baud). fuer fehlende argumente werden die standardwerte eingesetzt.

zeilen- und zeichenzahl sowie baudrate, die mit diesem befehl gesetzt wurden, gelten dann auch fuer die befehle p1 bis p5 (bis zum druecken von reset oder dem ausfuehren von rst 0). wurde p0 nicht ausgefuehrt, verwenden diese befehle die standardwerte.

nach der initialisierung kann die ein- und ausgabe durch die ausgabe der folgenden steuerzeichen gesteuert werden:

ctrl a (ascii 01): die druckausgabe wird initialisiert, d.h. zeichen, die ab jetzt ausgegeben werden, werden auch auf dem fernschreiber gedruckt. der fernschreibermotor wird eingeschaltet, falls er noch nicht laeuft.

ctrl b (ascii 02): die eingabe fernschreiber wird initialisiert. der fernschreibermotor wird eingeschaltet, falls er noch nicht laeuft. die nascom-tastatur ist jetzt abgeschaltet. wird am fernschreiber 'a' eingegeben oder an der nascom-tastatur die shift-taste gedruickt, wird die eingabe vom fernschreiber wieder abgeschaltet.

ctrl c (ascii 03): ab jetzt keine weitere druckausgabe, der motor bleibt an.

ctrl d (ascii 04): schaltet motor aus, macht blattvorschub, falls druckausgabe initialisiert.

ctrl e (ascii 05): an den fernschreiber werden 50 nullzeichen ausgegeben, z.b. fuer anfang und ende eines lochstreifens.

die steuerzeichen kann man an der nascom-tastatur eingeben, indem man die taste 'a' und dann gleichzeitig z.b. die taste 'a' (ctrl a) drueckt. gibt man nacheinander ctrl a und ctrl b ein werden aus- und eingabe auf den fernschreiber umgeschaltet. jetzt kann der nascom zusammen mit allen programmen, die sich dafuer eignen, vom fernschreiber aus gefahren werden (z.b. auch ueber funk).

die adressen der eingaberoutine ttyin (b20f) wird bei rsuin (0c7b), die der ausgaberroutine ttyout (b07d) bei rsuout (0c78) eingetragen und die ein- und ausgabetafellen auf intu bzw. outtu umgesetzt. von maschinenprogrammen aus spricht man ttyin und ttyout ueber die normalen ein- und ausgaberroutinen an oder ueber uin und uout. die auszugebenen zeichen werden an ttyout im a-register uebergeben. nach einem aufruf von ttyin erfolgt ein ruecksprung, sobald ein zeichen vom fernschreiber vorliegt. beim ruecksprung ist dann das carry-flag gesetzt, das zeichen steht im a-register.

der nas-sys befehl 'n' setzt die ein- und ausgabetafellen wieder auf die standardwerte.

p1 a b (e b001 a b)

auf dem fernschreiber werden die speicherinhalte ab der adresse a bis zur adresse b (ausschliesslich) als hex-listing gedruckt.

p2 a b (e b002 a b)

der bildschirminhalt wird auf dem fernschreiber gedruckt. a ist die nummer der ersten zeile (a=1: ab 080a) und b die anzahl der zeilen.

p3 a b (e b003 a b)

dient zum ausstanzen eines speicherbereichs auf lochstreifen, verwendet wird das gleiche format wie bei tabulate (mit pruefbytes). nach der eingabe des befehls fragt der rechner: 'titel?'. danach kann man den namen des programms eingeben, der beim wiedereinlesen des lochstreifens angezeigt wird.

p4 (e b004)

einlesen eines lochstreifens, der mit p3 gestanzt wurde. fehlerhafte zeilen werden angezeigt (wie bei 'load').

p5 (a 6005)

-----

druckausgabe von naspen aus wird vorbereitet (bei 101e wird die druckroutine einetragen). will man mit naspen arbeiten, macht man zunächst einen kaltstart von naspen, kehrt zu nas-sys zurück und gibt dort den befehl p5 (evtl. vorher p0) ein. dann kehrt man mit einem warmstart zu naspen zurück. in naspen läuft dann die druckausgabe mit dem naspen-befehl p ueber den fernschreiber. diese beschreibung wurde uebrigens mit naspen editiert und mit tty-sys gedruckt.

betrieb mit anderen systemprogrammen:

tty-sy kann zusammen mit dem 8k-basic, mit asm/eprom v2.0, mit naspen, mit dis-sys und mit zeap verwendet werden. dabei gibt es folgende besonderheiten:

**8k-basic:**  
vor dem start des basic wird p0 ausgefuehrt. steuerzeichen werden mit chr\$( direkt oder vom programm aus abgeschickt. print chr\$(1) bewirkt z.b. die ausgabe von ctrl a.

**asm/eprom:**  
vor dem start von asm/eprom wird p0 ausgefuehrt. in asm/eprom werden dann mit dem befehl u die user i/o-routinen aktiviert. steuerzeichen kann man dann direkt an der tastatur eingeben.

**naspen:**  
die arbeit mit naspen ist bei dem befehl p5 beschreiben.

**dis-sys:**  
vor dem aufruf von dis-sys wird p0 ausgefuehrt und die druckausgabe durch eingabe von ctrl a aktiviert.

**zeap:**  
nach einem kaltstart von zeap kehrt man zu nas-sys zurück. dann traegt man bei 0f05 die adresse von ttyout (b02e) ein, fuehrt p0 aus und gibt ctrl a ein (von zeap aus kann man keine steuerzeichen abschicken). mit einem warmstart kehrt man zu zeap zurück und kann dort mit dem befehl u source-listings und mit der option 4 beim assemblieren assembler-listings drucken.

doppelzeichen:

die folgende tabelle zeigt, fuer welche anzahlzeichen doppelzeichen verwendet werden.

!	q.	^	q'	z	qh	#	qs
%	qp	&	qt	x	qx	,	qi
<	q(	>	q)	@	qa	[	qe
\	q/	]	qf	^	q?	_	q-

hardware:

die steuersignale fuer den fernschreiber stehen an der pio port b zur verfuegung. bit 0 dient zum schalten des linienstroms. bit 0=0 bedeutet: linienstrom unterbrochen. bit 1 ist der eingang von den sendekontakten und bit 6 steuert den motor. bit 6=0 bedeutet: motor ein. nach dem abschalten des motors werden alle ausgaenge hochgeholt. zur anschluss des fernschreibers muss man ein geeignetes interface benutzen, das galvanisch trennen sollte (optokopler).

empfohlen wird das interface 'fernschreiberinterface v 1.1.'

bernd ploss  
datenelektronik - systementwicklung

██████████ rastatt  
tel. ██████████

Falls Sie gerne ein TTY-SYS haben möchten, senden Sie bitte Ihre Bestellung an MK-Systemtechnik. Wir liefern Ihnen das TTY-SYS z.Zt. ab Lager.

## STRICHCODEPROGRAMME

Bestimmt ist Ihnen schon einmal der Strichcode aufgefallen, mit dem viele Waren gekennzeichnet werden. In vielen Kaufhäusern und Supermärkten sind die Kassen schon mit Strichcodelesern ausgerüstet, die das umständliche Eintippen von Artikelnummer und Preis überflüssig machen.

Aber nicht nur zur Kennzeichnung von Waren ist der Strichcode geeignet. Eine Anwendung, für die er geradezu ideal ist, sind Computerprogramme. Bisher kann man ein Programm entweder in Klartext in einer Zeitschrift abdrucken. Dann muß der Leser das Programm mühsam von Hand in den Rechner eintippen. Oder man vertreibt das Programm auf einem speziellen maschinenlesbaren Datenträger (z. B. Kassette oder EPROM). Dann muß aber in der Regel jedes Exemplar einzeln vervielfältigt werden, was das Programm natürlich erheblich verteuert.

Programme in Strichcode haben die Vorteile beider Verfahren. Das Programm kann billig durch Druck oder Fotokopia vervielfältigt werden, ist aber direkt maschinenlesbar. Um ein Programm im Strichcode in den Rechner zu laden benötigt man einen Strichcodeleser und Software zur Dekodierung des Strichcodes.

Ein Problem war bislang der Strichcodeleser. Hewlett-Packard liefert einen Lesestift für knapp 300 DM. Ich war allerdings skeptisch, ob ein Lesestift zu diesem Preis für viele Nascom-Besitzer interessant wäre oder ob die Mehrheit nicht weiterhin die Programme von Hand eintippen würde. Jetzt hat die Zeitschrift mc aus dem Franzis-Verlag einen Strichcodeleser für ca. 100 DM angekündigt und damit wird der Strichcode auch für das Nascom-Journal interessant.

In diesem Heft ist deshalb erstmals ein Programm im Strichcode abgedruckt. Dabei handelt es sich um das Programm Mastermind, das in einem der letzten Hefte bereits in Hex-Darstellung veröffentlicht wurde. Um das Programm in den Nascom einzulesen benötigt man lediglich noch einen Strichcodeleser, der direkt an die PIO angeschlossen wird,

und ein Dekodierprogramm. Das Dekodierprogramm für den Nascom ist ab sofort lieferbar und kostet 88,- DM incl. MwSt.  
Wenn der Strichcode bei vielen Lesern des Nascom-Journal auf Interesse stößt, können denn in Zukunft die veröffentlichten Programme auch in Strichcode erscheinen.

Bernc Ploss

```

00 0C80
01 0C99
02 0CB1
03 0CCA
04 0CF3
05 0CFC
06 0D14
07 0D2D
08 0D45
09 0D5D
0A 0D75
0B 0D80
0C 0DA5
0D 0DBE
0E 0DD7
0F 0DF0
10 0E09
11 0E29
12 0E3C
13 0E54
14 0E6B
15 0E83
16 0E9C
17 0EB4
18 0ECD
19 0EE5
1A 0EFD
1B 0F17
1C 0F20

```

Bedienungsanleitung: UFO-JAGD

Monitor: NAS-SYS  
Speicher: OC80 - ODAD  
Start: EOC80

Nach dem Druecken der Taste "G" beginnt das Spiel. Die Ufos bewegen sich mit unterschiedlicher Geschwindigkeit ueber den Bildschirm. Mit der "SPACE"-Taste koennen die Raketen auf die UFOs abgeschossen werden. Die Treffer werden gezaehlt. Fuer jedes Spiel stehen 20 UFOs zur Verfuegung. Ein neues Spiel kann durch Druecken der Taste "G" begonnen werden.

\* GERHARD BAIER

\* ERLANGEN

\*

\*

\*

```

-----
FFFF          DELAY      EQU      OFFFFH
00EF          PRS        EQU      0EFH
000F          GESCHW     EQU      0FH
62DF          IN         EQU      62DFH
00CF          RIN        EQU      0CFH
00F7          ROUT       EQU      0F7H
000C          CS         EQU      0CH
OC80          UFOJGD     ORG      OC80H
OC80          EF 0C      DB       PRS,CS
OC82          53 54 41 52 54 20 28 DB       'START (G):'
              47 29 3A
OC8C          00          DB       00H
OC8D          CF          UFOJGD1 DB       RIN
OC8E          FE 47      CMP      47H
OC90          20 FB      JR       NZ,UFOJGD1
OC92          EF 0C 00   DB       PRS,CS,00H
OC95          21 86 0D   LD       HL,UFOJGD1
OC98          11 CA 0B   LD       DE,OBCAH
OC9B          01 27 00   LD       BC,0027H
OC9E          ED B0      LDIR
OCA0          01 00 01   UFOJGD2 LD       BC,0100H
OCA3          D9          EXX
OCA4          01 3D 01   LD       BC,013DH
OCA7          DD 21 8A 0B LD       IX,08BAH
OCAB          FD 21 AA 0B UFOJGD3 LD       IY,0BAAH
OCAE          FD 36 FF 03 LD       (IY-01H),03H
OCB3          FD 36 00 5E LD       (IY+00),5EH
OCB7          FD 36 01 4C LD       (IY+01H),4CH
OCBB          FF FF      UFOJGD4 DW       DELAY
OCBD          10 35      DJNZ    UFOJGD5
OCBF          06 0F      GESCHA LD       B,GESCHW
OCC1          DD 36 FB 20 LD       (IX-05H),20H
OCC5          DD 36 FC 17 LD       (IX-04H),17H
OCC9          DD 36 FD 05 LD       (IX-03H),05H
OCCD          DD 36 FE 05 LD       (IX-02H),05H
OCD1          DD 36 FF 05 LD       (IX-01H),05H
OCD5          DD 36 00 3E LD       (IX+00H),3EH
OCD9          DD 23      INC     IX
OCDB          0D          DEC     C
OCDC          3E 00      LD       A,00H
OCDE          B9          CMP     C
OCDF          20 13      JR       NZ,UFOJGD5
OCE1          0E 3D      LD       C,3DH
OCE3          3E 32      LD       A,32H
OCE5          21 D7 0B   LD       HL,0BD7H
OCE8          BE          CMP     (HL)
OCE9          CA 8D 0C   JMP     Z,UFOJGD1
OCEC          23          INC     HL
OCED          CD 56 0D   CALL    ZAEHL

```

## UFD-JAGD

2

OCF0	DD 21 BA 08		LD	IX, 08BAH
OCF4	D9		UFOJGD5	EXX
OCF5	3E 00		LD	A, 00H
OCF7	B9		CMP	C
OCF8	20 06		JR	NZ, UFOJGD6
OCFA	DF 62		DW	IN
OCFC	30 29		JR	NC, UFOJGD9
OCFE	0E 0F		LD	C, 0FH
OD00	10 25		UFOJGD6	DJNZ
OD02	FD 36 00 20		LD	(IY+00H), 20H
OD06	0D		DEC	C
OD07	3E 00		LD	A, 00H
OD09	B9		CMP	C
OD0A	20 06		JR	NZ, UFOJGD7
OD0C	04		INC	B
OD0D	0E 00		LD	C, 00H
OD0F	D9		EXX	
OD10	18 99		JR	UFOJGD3
OD12	06 06		UFOJGD7	LD
OD14	C5		PUSH	BC
OD15	06 40		LD	B, 40H
OD17	FD 2B		UFOJGD8	DEC
OD19	10 FC		DJNZ	UFOJGD8
OD1B	C1		POP	BC
OD1C	FD 7E 00		LD	A, (IY+00H)
OD1F	FE 20		CMP	20H
OD21	20 08		JR	NZ, *+0AH
OD23	FD 36 00 5E		LD	(IY+00H), 5EH
OD27	D9		UFOJGD9	EXX
OD28	C3 BB 0C		JMP	UFOJGD4
OD2B	21 EF 0B		LD	HL, 0BEFH
OD2E	CD 56 0D		CALL	ZAEHL
OD31	CD 6A 0D		CALL	BLITZ
OD34	3E 32		LD	A, 32H
OD36	21 D7 0B		LD	HL, 0BD7H
OD39	BE		CMP	(HL)
OD3A	CA 8D 0C		JMP	Z, UFOJGD1
OD3D	23		INC	HL
OD3E	CD 56 0D		CALL	ZAEHL
OD41	C3 A0 0C		JMP	UFOJGD2
			*	
			*	
			* UNTERPROGRAMME	
			*	
OD44	16 20		CLSCR	LD D, 20H
OD46	21 BA 0B		CLSCR0	LD HL, 0BBAH
OD49	7D		CLSCR1	LD A, L
OD4A	FE 0A		CMP	0AH
OD4C	20 04		JR	NZ, CLSCR2
OD4E	7C		LD	A, H
OD4F	FE 0B		CMP	0BH
OD51	C8		RET	Z
OD52	2B		CLSCR2	DEC HL
OD53	72		LD	(HL), D
OD54	18 F3		JR	CLSCR1
			*	
OD56	ED 5F		ZAEHL	LD A, R

## UFO-JAGD

3

OD58	E6 0A		AND	0AH
OD5A	F6 04		OR	04H
OD5C	32 C0 0C		LD	(GESCHA+1), A
OD5F	11 39 30		LD	DE, 3039H
OD62	7E		LD	A, (HL)
OD63	34		INC	(HL)
OD64	BB		CMP	E
OD65	C0		RET	NZ
OD66	72		LD	(HL), D
OD67	2B		DEC	HL
OD68	34		INC	(HL)
OD69	C9		RET	
		*		
OD6A	06 03	BLITZ	LD	B, 03H
OD6C	16 2A	BLITZ1	LD	D, 2AH
OD6E	CD 46 0D		CALL	CLSCRO
OD71	CD 7D 0D		CALL	WAIT
OD74	CD 44 0D		CALL	CLSCR
OD77	CD 7D 0D		CALL	WAIT
OD7A	10 F0		DJNZ	BLITZ1
OD7C	C9		RET	
		*		
OD7D	C5	WAIT	PUSH	BC
OD7E	06 06		LD	B, 06H
OD80	FF FF	WAIT1	DW	DELAY
OD82	10 FC		DJNZ	WAIT1
OD84	C1		POP	BC
OD85	C9		RET	
		*		
OD86	55 46 4F 2D 4A 41 47	UFOTXT	DB	'UFO-JAGD'
	44 20 20 20 20 20			
OD93	30 31 2E 20 55 46 4F		DB	'01. UFO'
	20 20 20 20			
OD9E	41 62 73 63 68 75 65		DB	'Abschuesse: 00'
	73 73 65 3A 20 30 30			
	20			
ODAD	00		END	UFOJGD

00147 Statements Assembled  
 39161 Bytes frei  
 Keine Fehler erkannt

BLITZ	OD6A	BLITZ1	OD6C	CLSCR	OD44
CLSCRO	OD46	CLSCR1	OD49	CLSCR2	OD52
CS	000C	DELAY	FFFF	GESCHA	0CBF
GESCHW	000F	IN	62DF	PRS	00EF
RIN	00CF	ROUT	00F7	UFOJGD	0CB0
UFOJGD1	0CB0	UFOJGD2	OCA0	UFOJGD3	OCAB
UFOJGD4	OCBB	UFOJGD5	OCF4	UFOJGD6	OD00
UFOJGD7	OD12	UFOJGD8	OD17	UFOJGD9	OD27
UFOTXT	OD86	WAIT	OD7D	WAIT1	ODB0
ZAEHL	OD56				

0C80	EF	0C	53	54	41	52	54	20	35	0D1B	2B	10	FC	C1	FD	7E	00	FE	96
0C88	28	47	29	3A	00	CF	FE	47	7A	0D20	20	20	08	FD	36	00	5E	D9	DF
0C90	20	FB	EF	0C	00	21	86	0D	66	0D28	C3	BB	0C	21	EF	0B	CD	56	FD
0C98	11	CA	0B	01	27	00	ED	B0	4F	0D30	0D	CD	6A	0D	3E	32	21	D7	F6
0CA0	01	00	01	D9	01	3D	01	DD	A3	0D38	0B	BE	CA	8D	0C	23	CD	56	B7
0CAB	21	8A	0B	FD	21	AA	0B	FD	37	0D40	0D	C3	A0	0C	16	20	21	BA	DA
0CB0	36	FF	03	FD	36	00	5E	FD	82	0D48	0B	7D	FE	0A	20	04	7C	FE	83
0CB8	36	01	4C	FF	FF	10	35	06	90	0D50	0B	CB	2B	72	1B	F3	ED	5F	21
0CC0	0F	DD	36	FB	20	DD	36	FC	18	0D58	E6	0A	F6	04	32	C0	0C	11	5E
0CC8	17	DD	36	FD	05	DD	36	FE	11	0D60	39	30	7E	34	BB	C0	72	2B	A0
0CD0	05	DD	36	FF	05	DD	36	00	0B	0D68	34	C9	06	03	16	2A	CD	46	CE
0CD8	3E	DD	23	0D	3E	00	B9	20	46	0D70	0D	CD	7D	0D	CD	44	0D	CD	CC
0CE0	13	0E	3D	3E	32	21	D7	0B	BD	0D78	7D	0D	10	F0	C9	C5	06	06	A9
0CE8	BE	CA	8D	0C	23	CD	56	0D	68	0D80	FF	FF	10	FC	C1	C9	55	46	BC
0CF0	DD	21	8A	0B	D9	3E	00	B9	5C	0D88	4F	2D	4A	41	47	44	20	20	67
0CF8	20	06	DF	62	30	29	0E	0F	E1	0D90	20	20	20	30	31	2E	20	55	01
0D00	10	25	FD	36	00	20	0D	3E	E0	0D98	46	4F	20	20	20	20	41	62	5D
0D08	00	B9	20	06	04	0E	00	D9	DF	0DA0	73	63	6B	75	65	73	73	65	10
0D10	18	99	06	06	C5	06	40	FD	E2	0DAB	3A	20	30	30	20	00	00	00	8F

```

100 REM LOTTOZAHLEN-GENERATOR
110 REM
120 REM (C) 1981 GERHARD BAIER
130 REM
140 REM ERLANGEN
150 REM
200 PRINT
210 PRINT "LOTTOZAHLEN-GENERATOR"
220 DIM L(5)
250 FOR I=0 TO 5
260 L(I)=INT(RND(.5)*48+1.5)
270 IF I=0 THEN 300
280 FOR J=0 TO I-1
290 IF L(I)=L(J) THEN 260
295 NEXT J
300 NEXT I
310 FOR I=0 TO 4
320 IF L(I)<L(I+1) THEN 370
330 B=L(I)
340 L(I)=L(I+1)
350 L(I+1)=B
360 GOTO 310
370 NEXT I
400 PRINT
410 PRINT "IHR LOTTO TIP:";PRINT
420 FOR I=0 TO 5
430 PRINT TAB(6);L(I)
440 NEXT I
450 PRINT;PRINT
460 PRINT "VIEL GLUECK"
470 END

```

## Printplot mit Super Tiny BASIC

=====

Der Printplot ist eine gute Möglichkeit, den Zeitverlauf von Funktionen auf einem Zeilendrucker oder Fernschreiber darzustellen.

Die Auflösung reicht natürlich bei weitem nicht an einen x-t-Schreiber oder den von Peter Bentz beschriebenen Trommelplotter heran.

Die verwendeten Variablen haben folgende Bedeutung:

- E ... Endwert der Ordinate
- N ... Maximalwert auf der Zeitachse
- T ... läuft von 0 bis N
- X,Y,Z ... Darzustellende Zeitfunktionen.

In den Zeilen 42 - 44 steht nur ein Demonstrationsbeispiel, hier können umfangreiche Rechnungen oder Aufrufe an Meßgrößen stehen. (Sie müssen dann allerdings die etwas schlampigen Zeilennummern ändern, um mehr Platz zu bekommen).

Wenn Sie mehr oder weniger Größen darstel-

len wollen, müssen Sie die Zeilen 55 - 65 entsprechend ändern bzw. erweitern. In diesem Programmteil wird sichergestellt, daß bei Gleichheit mehrerer Werte nur ein Symbol ausgedruckt wird.

Eine Anmerkung zu Zeile 10: die Befehle V=3018; MCC bewirken, daß die Zeile mit dem Masstab in die Titelzeile des Bildschirms geschrieben wird. Diese Zeile wird dann bei großen N nicht mitgerollt.

&gt;EF600

B-BASIC V1.1

OK

&gt;L.

```

10 L.E=4,N=12,M=3018;MCC;P." ",
20 FOR B=0 TO E-1;P.#1;B,".....",
30 NEXT B
40 P.#1;B
41 FOR T=0 TO N;P.#2;T,
42 L. X=3*T
43 L. Y=10+E-X
44 L. Z=40/(T+1) *X
45 L. C=0
50 FOR A=0 TO 10+E
55 IF (A=C)+(A=X)+(A=Y)+(A=Z) C=C+5
60 IF A=X P."x";GOTO 80
61 IF A=Y P."y";GOTO 80
62 IF A=Z P."z";GOTO 80
65 IF A=C P.".";C=C+5;GOTO 80
70 P." ",
80 NEXT A
90 P.
100 NEXT T

```

OK

&gt;KON

	0	1	2	3	4
0x	.	.	.	.	.
1.	x	.	.	.	.
2.	.	x	.	.	.
3.	.	.	x	.	.
4.	.	.	.	x	.
5.	.	.	.	.	x
6.	.	.	.	.	.
7.	.	.	.	.	.
8.	.	.	.	.	.
9.	.	.	.	.	.
10.	.	.	.	.	.
11.	.	.	.	.	.
12.	.	.	.	.	.

OK

&gt;

OK

VERKAUFE

Number Cruncher Kit für NASCOM 1 (MM 57109) incl. ausführlicher  
Dokumentation Preis: DM 200.-

ZEAP 2.= Tape-Version für NAS-SYS Preis: DM 120.-

12k extended CLD-BASIC für NASCOM 2 (Tape) Preis: 80.-

Wolfgang Burgbacher, [REDACTED], [REDACTED], Tel.: [REDACTED]/

VERKAUFE \*\*\*\*\*SONDERANGEBOT\*\*\*\*\*Drucker mit Parallelschnittstelle für NASCOM

Einzelstück; Microline 80 Drucker, Neupreis 2048.- von KONTRON,  
zu haben für 1546.- + MWSt. Gerät war ca. 4 Monate in Betrieb.  
Läuft problemlos auch mit BASIC Toolkit.

\*\*\*\*Das ist der letzte Drucker, der noch verfügbar ist. Die anderen  
sind aufgrund unserer Annonce im NASCOM-JOURNAL 3/81 schon ver-  
kauft. \*\*\*\*

## NOCH EINIGE STÜCKE:

3k Assembler auf EPROM mit deutscher Beschreibung statt 269.- nur 199.-

NASPEN wahlweise für NASBUG oder NAS-SYS. (Bitte angeben). Statt  
169.- nur 99.-

Graphik-ROM-Kit und Schach-Erweiterungs-Kit erweitert NASCOM 1  
auf die NASCOM 2 Graphik. Erlaubt zusätzlich Schach-Graphik (aus-  
tauschbar). Statt 189.- nur 155.-

SCHACH, alte Version ohne Graphik für 49.- ist ausverkauft !

SCHACH neue Version mit Graphik, statt 128.- jetzt nur noch  
88.-

Wahlweise für NASBUG oder NAS-SYS, wahlweise für NASCOM 1  
oder für NASCOM 2. Im Preis nicht inbegriffen das Graphik-  
ROM.

Diese Sonderangebots-Aktion läuft bis zum 31.5.1981

Alle Bestellungen an MK-Systemtechnik  
Pater-Mayer-Straße 6  
6728 Germersheim/Rhein

# kleinanzeigen

## VERKAUFE

Eprom-Programmer für NAS-SYS 140.- DM      8k EPROM-Karte für KIM 1 50.-  
Fernschreiber LO 15 c mit Lochstreifensender und Empfänger sowie Inter-  
face für NASCOM 250.- DM.

Achim Kaufmann      [REDACTED]      [REDACTED]      Tel. [REDACTED]

---

## VERKAUFE

Verk. wegen Hobbyaufgabe NASCOM II, 32k, NAS-SYS "i" (Interrupt)  
mit Ass., Disass., Toolkit, Debug, NASPEN und "Edel"-Prommer  
(2708 und 5V-2716), kompl. Literatur      DM 2000.-

Dazu Drucker TX 80,      DM 1300.-

Monitor 9", grün      DM 350.-

Kassettenrecorder, Eigenbau      DM 50.-

Tel.: [REDACTED]

---

## VERKAUFE

ZEAP 1, 3k Editor/Assembler für T2 und T4 , Original-Handbuch und  
Kassette für 50.- DM.

Max Ballarin , [REDACTED] , [REDACTED]

---

## VERKAUFE

NASCOM 1 mit Buffer-Platine, 32k RAM voll bestückt, Controllerkarte  
1. Floppy-Disk, 2. Netzteil getrennt. Metall-Gehäuse, Tastatur in  
Plastikgehäuse, Preis VB, Andriessen, [REDACTED] , [REDACTED]

---

## VERKAUFE

NASCOM CLD-Floppy mit NASBUS-Controller-Karte und BASF-Laufwerk samt  
Unterlagen und CLD-DOS neuwertig für 1000.-. 48k RAM-Karte für NASCOM 1  
oder NASCOM 2 aufgebaut und getestet mit 48k bestückt 500.-

Helmut Riedmann OE9ERI [REDACTED] [REDACTED] [REDACTED]

---