

# 80-bus journal

Zeitschrift für NASCOM, GEMINI und andere  
Z80-Anwender

1. JAHRGANG • JANUAR 1983 • AUSGABE 1

Druckausgabe von

TV-Bildern



Der Heftpreis beträgt DM 5,-. Ein Abonnent erhalten  
Sie für DM 60,- im Jahr.

# 80-bus journal INTERN

## INHALT

2 80-BUS-Journal intern	RED
3 Leserbriefe	
4 TV-Bilder	Günter Böhm Klaus Mombaus Clemens Ballarin Michael Bach
15 GREY PLOT	Gerhard Klement
15 Druckeranschluß	Günter Böhm
16 SORT	Christian Peter
21 Konfigurationen	Günter Kreidl
24 Der neue Monitor	Günter Kreidl
26 RELOCATOR	Günter Kreidl

Liebe Leser!

Sie halten mit diesem Heft die erste Ausgabe einer neuen Zeitschrift in Händen, doch wird Inhalt und Gestaltung vielen von Ihnen nicht fremd sein. Die Herausgeber besorgten früher die Redaktion des NASCOM-Journals und wollten diese Zeitschrift übernehmen, als sie von Ihrem Herausgeber eingestellt werden sollte. Dies scheiterte an den finanziellen Forderungen dieses Herausgebers. Wir entschlossen uns daraufhin zur Herausgabe einer eigenen Zeitschrift und wurden darin von einer Vielzahl von Mitarbeitern und Lesern des NASCOM-Journals unterstützt. Da wir nunmehr unabhängig von einzelnen Firmeninteressen arbeiten können, sind Konzeption und Thematik des 80-BUS-Journals auf einen größeren Anwenderkreis hin ausgerichtet. Wir hoffen, daß Sie alle viele Anregungen für die Anwendung, Programmierung und Erweiterung von Z80-Mikrocomputersystemen im 80-BUS-Journal finden werden, und daß möglichst viele Leser sich die Zeit nehmen werden, auch einmal einen Artikel für das Journal zu schreiben.

Günter Böhm & Günter Kreidl

### THEMEN und MITARBEITER

Das 80-BUS-Journal ist keine professionelle Zeitschrift, die von einigen wenigen für einen großen Leserkreis gemacht wird. Sie soll ein Organ des Austauschs sein zwischen den Lesern. Wir gehen davon aus, daß praktisch jeder Leser Besitzer eines NASCOM-, GEMINI- oder eines anderen Z80-Mikrocomputersystems ist. Die Anwendung derartiger Systeme erfordert Kenntnisse in Programmierung und Interfacetechnik. Man kann zwar für fast jede Aufgabe die nötige Hard- und Software kaufen, aber Selbermachen spart Geld und macht Spaß, und außerdem lernt man viel dabei. Warum soll aber das Ergebnis unserer Arbeit nur uns selbst zugute kommen? Viele andere können vielleicht unsere Programme oder Schaltungen ebenfalls gebrauchen. Je mehr Leser die Ergebnisse ihrer Arbeit (oder ihres Spielens - es muß ja nicht alles gar so ernst sein!) dem 80-BUS-Journal zur Verfügung stellen, desto interessanter und vielseitiger wird unsere Zeitschrift. Die Redaktion ist dankbar für jeden Beitrag, seien es Artikel, Schaltungen, Programme oder Leserbriefe.

### HERAUSGEBER und KORRESPONDENTEN

Die ganze Redaktionsarbeit wird von den Herausgebern in ihrer Freizeit besorgt. Dies ist überhaupt nur möglich durch ein ganz rationelles System (dazu haben wir ja schließlich auch alle Computer!) und durch die Hilfe einiger Leser. Allein der Schriftverkehr ist kaum zu bewältigen. Sie schreiben uns, schicken uns vielleicht einen Beitrag und erwarten natür-

## IMPRESSUM

### HERAUSGEBER:

Günter Böhm Ludwigshafener Str. 21d  
75 Karlsruhe Tel. [REDACTED]

Günter Kreidl Redaktion, Layout (Grafik), Versand  
4172 Straelen Bertenweg 18  
Tel. [REDACTED]

Redaktion, Layout (Text), Buchhaltung

### MITARBEITER:

Wolfgang Mayer-Gürr [REDACTED]  
[REDACTED] Recklinghausen - Tel. [REDACTED]

Korrespondent: Ruhrgebiet  
Clemens u. Max Ballarin [REDACTED]

[REDACTED] Ueberlingen Tel. [REDACTED]  
Cassettenversand

### VERLAG:

Günter Kreidl 4172 Straelen

### VERTREBSWEISE und BEZUGSPREIS:

Einzelheft DM 5,-

Doppelheft DM 10,-

Jahresabonnement In- und Ausland DM 60,-

Es erscheinen 10 Hefte pro Jahr, davon zwei

Doppelhefte. Es können jeweils nur ganze

Jahrgänge abonniert werden. Bei Bestellungen

nach dem Erscheinungsdatum des ersten Heftes

eines Jahrgangs werden die bereits erschie-

nenen Hefte nachgeliefert. Die Lieferung von

Einzelheften durch den Verlag ist nicht mög-

lich. Bitte zahlen Sie direkt bei der Be-

stellung auf das Postcheckkonto:

Günter Kreidl [REDACTED] PSchA Essen

### HAFTUNG und RECHTE:

Für Fehler in Texten, Bildern, Programmen

und Schaltungen und daraus entstehende Schäd-

den kann keine Haftung übernommen werden.

Alle Rechte verbleiben grundsätzlich bei den

Autoren der Beiträge. Die Veröffentlichung von

Programmen und Schaltungen geschieht nur

für den persönlichen Gebrauch der Abonnenten

des 80-BUS-Journals; jede kommerzielle Aus-

wertung ist nur mit Genehmigung des Verfas-

sers erlaubt. Beiträge, die nicht mit einem

Copyright-Vermerk versehen sind, dürfen für

nichtkommerzielle Verwendung vervielfältigt

werden, wenn als Quelle das 80-BUS-Journal

und der Verfasser angegeben werden.

lich eine Antwort. Zusammen mit der anderen Arbeit ist das aber nicht zu schaffen. Deshalb sind wir auf den Gedanken gekommen, daß uns einige Leser dabei unterstützen könnten, indem sie in einer bestimmten Region den Kontakt mit den Lesern aufrechterhalten, Anfragen beantworten, Programme kopieren und was an Wünschen so alles an uns herangetragen wird. Wenn diese "Korrespondenten" jeweils nur für eine Region zuständig sind, könnten sie vielleicht auch einmal regionale Treffen der Leser organisieren, falls daran Interesse besteht. Wer dazu Lust hat, kann sich ja bei uns melden. Der Redaktion können Sie alle helfen, wenn Sie Ihre Beiträge nur noch auf Cassette einschicken (Software, Artikel, Leserbriefe). Wir werden uns bemühen, mit der Hilfe von Herrn Ballarin, jede Cassette mit den Programmen einer bestimmten Ausgabe bespielt zurückzusenden. Bitte seien Sie nicht böse, wenn das mal länger dauert!

Hier noch ein kurze Vorstellung der Herausgeber: Günter Böhm, Jahrgang 1947, ist von Beruf Realschullehrer und besitzt seit Ende 1978 einen NASCOM. Er kam über den Bau von Musiksynthesizern zur Computerei und ist bezüglich Programmierung und Computertechnik Autodidakt. Seit Anfang 1981 Mitarbeiter, ab Juni 1981 Redakteur des NASCOM-Journals. Günter Kneidl, Jahrgang 1949, ist von Beruf Geschäftsführer einer Großhandels- und Verbrauchergenossenschaft, seit 1980 Besitzer eines NASCOM und in Bezug auf die Computerei ebenfalls Autodidakt. Er war ebenfalls von Juni 1981 an Redakteur des NASCOM-Journals.

#### IN DIESEM HEFT

Dieses Heft hat mehrere Schwerpunkte, die z.T. auf Artikeln im NASCOM-Journal aufbauen. Günter Böhm hatte dort das Interface für eine TV-Kamera vorgestellt. Er selbst und eine Reihe von Lesern haben die damit gewonnenen Daten in verschiedener Weise verarbeitet. In diesem Heft sind eine Reihe von Programmen in Assembler, Basic und Pascal zum Ausdruck derartiger Bilder mit verschiedenen Drucken enthalten. Die Programme lassen sich sicherlich auch für andere grafische Anwendungen anpassen. Ein verbessertes Interface incl. Platinenlayout wird in einem der nächsten Hefte vorgestellt. Für ein ebenfalls im NASCOM-Journal erschienenes Dateiverwaltungsprogramm hat Christian Peter einen Bubbleplot in Maschinensprache geschrieben, der sich sicher auch für andere Anwendungen eignet. Mehr in die Zukunft weisen hingegen die beiden Artikel von Günter Kneidl, die sich mit Hard- und Software-Kompatibilität von NASCOM-, GEMINI- und ECB-BUS-Systemen befassen. Die Entwicklung eines CP/M- und RP/M-kompatiblen Betriebssystems für den NASCOM, das auch den Programmaustausch zwischen NASCOM- und GEMINI-Systemen ermöglicht, wird auch in den kommenden Ausgaben einen Schwerpunkt bilden.

#### VORSCHAU

Es sind einige größere Hardware-Erweiterungen in der Entwicklung, die wir in den nächsten Ausgaben vorstellen werden. Bereits im nächsten Heft wird die Hard- und Software für ein Spracherkennungssystem vorgestellt. Dazu wird auch eine Folie mit dem Platinenlayout erhältlich sein. Ebenfalls im kommenden Heft wird einEPROMmer für 2732/64-Chips inklusive Software vorgestellt. In Arbeit ist außerdem eine Karte mit hochauflösender Farbgrafik, eine Floppy-Controller-Karte, sowie eine Europakarten-Ausführung des 80x24-Zeichen Video-Interface. Auch wollen wir die im NASCOM-Journal begon-

nenen Reihen "Seite(n) für Einsteiger" und "Tips, Tricks und Käfer" fortsetzen. Wie man sieht (ein bißchen Eigenwerbung!) - es lohnt sich, das 80-BUS-Journal zu abonnieren.

#### FOLGENSERVICE

Von einer Reihe von Schaltungen, die im NASCOM-Journal erschienen sind bzw. im 80-BUS-Journal erscheinen werden, sind Abzfliegen lieferbar: Interface für Mini-Digital-Rekorder  
Grafikerweiterung (N2 Grafik für NASCOM 1)  
Kansas-City-Kassetteninterface  
A/D-Wandler  
Soundgenerator  
Monitor-Umschaltkarte (für 2 verschiedene Betriebssysteme)  
Video-Karte (80 x 24 Bildschirm)  
2716-Eprommer  
Spracherkennung

Der Unkostenbeitrag für eine normale Folie incl. Porto und Verpackung beträgt DM 7,50. Die großen Folien (doppelseitige Platine) für die Videokarte sind für DM 12,50 zu beziehen. Als Bestellung genügt eine Überweisung auf das Postcheckkonto

Günter Böhm [Redaktion] P.SchA Klak  
mit Angabe der gewünschten Folie.

## LESERBRIEFE

Das mit der Journalentwicklung ist ja ein dickes Stück! Hoffentlich klappt's mit dem 80-BUS-Journal, aber es wird schon eine starke Werbung nötig sein. Auf das Hochglanzpapier kann ich gerne verzichten. Wie wär's mit einer Offenlegung der Finanzen? (Ist geplant, RED.) Wie könnte man denn neue Leser werben? Außerdem erscheint mir fraglich, wieviel Zukunft der NASCOM überhaupt hat. (Nachteile: Bildschirmformat, Abmessungen der Platine, in Deutschland ungebräuchlicher Bus...) Mit Software hingegen ist m. E. der NASCOM ganz gut versorgt. (RED: Deshalb auch unser neues Konzept im Journal!) Anbei der Ausdruck Ihres Bildes und die Pascal-Programme. Viel Erfolg bei Ihren Projekten; das 80-BUS-Journal werde ich weiterhin unterstützen. Michael Bach, Stegen

Du stehst nun im NASCOM-Journal seit drei Jahren da man das Recht auf eine Kleinanzeige von 40 Wörtern hat, und nun wird mir die Anzeige böswillig unterschlagen! Dies und die sogenannte "Doppelausgabe" machen das Maß voll: Ich kündige das Abonnement und glaube kaum, daß mir das jemand unter den gegebenen Umständen streitig machen kann. Außerdem werde ich darauf drängen, daß mir die noch ausstehende Dezemberausgabe vor dem nächsten Dezember ausgeliefert wird. Das 80-BUS-Journal habt Ihr doch hoffentlich noch diesen Monat fertig?

Dieter Oberle, Vollmersweiden  
Du hältst es in Händen, Lieber Dieter  
Noch eine Bemerkung, um Mißverständnisse zu vermeiden: Kündigen kann man das NASCOM-Journal nicht bei uns! Wir sind nicht der Rechtsnachfolger und haben auch sonst nichts mehr damit zu schaffen. RED

# TV-Bilder

von GÜNTER BÖHM

Bedauerlicherweise ist das Oktoberheft des NASCOM Journals immer noch nicht erschienen, und es liegt nicht in unserer Macht, ob es dieses Jahr noch oder überhaupt jemals erscheint. So wurden meine Fragen zur Hardware des TV- Interfaces noch nicht beantwortet, und somit bin ich auch noch nicht daran gegangen, das Platinenlayout zu entwerfen. Vielleicht ist dies garnicht von Nachteil, denn inzwischen kam ich an einen Artikel von Rolf-Dieter Klein aus ELEKTRONIK (Heft 3/82) über einen A/D-Wandler zur Digitalisierung von TV-Bildern, der mir neue Impulse gab. Hier werden keine schwer beschaffbaren TTL-Speicher benötigt, und der Rechner bestimmt selbst, welchen Punkt er abtastet, und er kann sich zum Abspeichern so viel Zeit lassen, wie er will. Die Darstellung von Graustufen hat mich auch dazu angeregt, mit dieser Möglichkeit zu experimentieren. Zudem hat mir Günter Kreidl einen Artikel aus der selben Zeitschrift in Aussicht gestellt, der mir vielleicht noch weitere Ideen bringt, sodaß dadurch eine Schaltung entsteht, die es dem Anwender überläßt, wie hoch seine Auflösung sein soll und wieviele Graustufen er verwenden möchte.

So wurde in der Zwischenzeit das Hauptaugenmerk nicht auf die Abspeicherung der TV-Bilder gelenkt, sondern auf die Auswertung; Ausgabe auf Bildschirm und Drucker. Anschließend an meinen Artikel finden Sie die Lösungen einiger Leser/Mitarbeiter, wie sie das gespeicherte Bild auf verschiedenste Weise zu Papier (bzw. Bildschirm) gebracht haben. Damit auch Sie mit Druckausgaben experimentieren können, drucken wir das Listing ab. Es nimmt zwar Platz weg, ist aber notwendig für alle, die das TV-Interface nicht nachgebaut haben. Das Eintippen ist einfacher als gewöhnlich, da kleinere Fehler keine Rolle spielen. Deshalb wurde auch auf Prüfsummen verzichtet. Gerne drucken wir auch Ihre Ausgaberroutine ab, sofern Sie von den bereits veröffentlichten abweicht und anderen Lesern hilft, die Bildausgabe an ihren Drucker anzupassen.

Doch hier nun zunächst die Programme, die ich im Laufe der Beschäftigung mit meinen Videoaufnahmen erstellt habe.

## TESTPROGRAMM

Dieses Programm entspricht dem BASIC Programm aus dem Oktoberheft des NASCOM Journals. Natürlich ist es weit schneller, zusätzlich kann man den Bildausschnitt wählen, der auf dem Schirm dargestellt werden soll. Die Parameter werden als Hexzahl zusammenhängend eingegeben, z.B. bedeutet 1010, daß der Bildausschnitt mit der sechzehnten Zeile bei Punkt sechzehn (von 256) anfängt.

Dieses Programm eignet sich gut zum Test der Hardware, weil es ohne Umweg über den Speicher direkt Zeile für Zeile auf dem Bildschirm ausgibt (ein Gesamtbild dauert etwa 6 Sekunden). Bei Hardwarefehlern, Wackelkontakten oder auch zum Einstellen der Bildhelligkeit ist das von Vorteil.

Mit NL wird die nächste Abspeicherung aufgerufen; mit ESC kann man einen neuen Bildausschnitt wählen. Wenn ich mich recht erinnere, kehrt man mit N zu NASSYS zurück.

```
1000 F5 C5 D5 E5 D7 10 47 7E 30
1008 E6 C0 FE C0 78 20 01 B6 CB
1010 77 E1 D1 C1 F1 C9 7C 0E 4E
1018 03 16 FF 14 91 30 FC 81 92
1020 47 7D 0E 00 CB 3F 30 01 3D
1028 0C 5F D7 13 3E 01 04 87 57
1030 10 FD CB 41 28 04 87 87 93
1038 18 02 CB 3F F6 C0 C9 26 11
1040 00 7A E6 0F 6F C5 06 06 FF
1048 29 10 FD C1 16 00 7B FE DE
1050 30 38 02 1E 2F 19 11 0A 4B
1058 08 19 C9 EF 0C 41 55 53 36
1060 4C 45 53 45 4E 20 44 45 90
1068 53 20 42 49 4C 44 45 53 9E
1070 20 41 42 0D 5A 45 49 4C 64
1078 45 2F 50 55 4E 4B 54 20 AE
1080 28 32 20 48 45 58 5A 41 8A
1088 48 4C 45 4E 29 3F 0D 00 34
1090 2A 0C 0C DF 66 2A 0E 0C 6B
1098 DF 66 DF 6A DF 63 DF 79 D0
10A0 3E CF D3 07 D3 06 3E C0 6E
10A8 D3 07 D3 06 EF 0C 00 3E A4
10B0 20 D3 04 3A 0C 0C 47 3E 8E
10B8 32 D3 04 3C D3 04 10 F7 EB
10C0 06 00 0E 00 3E 32 D3 04 2B
10C8 3E 33 D3 04 3E 32 D3 04 67
10D0 3E 36 D3 04 DB 04 CB 77 4C
10D8 28 FA 3E 22 D3 04 C5 3A 40
10E0 0E 0C 47 3E 2A D3 04 3E CE
10E8 22 D3 04 10 F6 C1 3E 2A 20
10F0 D3 04 3E 22 D3 04 DB 04 ED
```

```

10F8 CB 7F 28 0E 78 69 FE 03 6A
1100 30 02 C6 30 D6 03 67 CD 46
1108 00 10 0C 3E 60 B9 20 DE 8A
1110 3E 12 D3 04 3E 22 D3 04 7F
1118 04 3E 30 B8 20 A4 CF FE E4
1120 0D 28 89 FE 1B CA 5B 10 3D
1128 FE 4E 20 F2 DF 5B EC F9 B6

```

Start 105B

#### TV-MENUE

Dieses etwas umfangreichere Programm besteht aus drei Teilen, die von einem Menue aufgerufen werden können.

Durch Drücken der A-Taste wird die Aufnahme durch die TV-Kamera gestartet. Nach etwa 6 Sekunden blinkt der Cursor wieder und meldet, daß das Bild im Speicher ab 2000H abgelegt ist.

Nun kann man das Bild durch B auf dem Bildschirm betrachten. Es erscheint zunächst mit der linken oberen Ecke, das heißt es wird der Bildausschnitt sichtbar, der mit Zeile 1 Punkt 1 beginnt. Durch die Cursor-Tasten kann man das Bild verschieben. So kann das Bild von links nach rechts und von oben nach unten (danach oder dazwischen auch umgekehrt) durchfahren werden. Die Abstände zwischen den einzelnen Bildausschnitten bilden einen Kompromiß zwischen Geschwindigkeit (es würde ja ewig dauern, bis das Bild Punkt für Punkt von links nach rechts verschoben wäre, da es sich ja nach jedem Tastendruck neu aufbauen muß) und der Position von Details auf dem Schirm, um Einzelheiten erkennen zu können. Hier ist für die Zukunft eine Routine vorgesehen, die es ermöglicht, einzelne Punkte auf dem Schirm und gleichzeitig im Speicher zu manipulieren (zu setzen oder zu löschen).

ESC läßt das Programm zum Menue zurückspringen.

Nun kann man durch D das gespeicherte Bild ausdrucken lassen. Die Druckroutine bezieht sich im vorliegenden Listing auf einen TANDY Lineprinter VIII, kann aber sicher leicht für einen anderen Drucker geändert werden. Vielleicht helfen dabei die Beispiele der nachfolgenden Autoren.

Die abgedruckte Modifikation erlaubt es, das Bild im Gesamten auf dem Schirm zu betrachten. Die Auflösung läßt aber sehr zu wünschen übrig. Um aber schnell die Qualität eines Bildes zu erfassen (z.B. was die

Helligkeitseinstellung angeht) tut sie gute Dienste. Hierbei wurde wie im Originalprogramm die Klötzchengrafik von Michael Bach verwendet, für deren Erstellung ich mich nochmals im Namen der "Maschinen-Programmierer" herzlich bedanken möchte.

Eine weitere Programmvariante ist der Einsatz der Blockgrafik des TANDY VIII zum Ausdruck des Bildes. Das Ergebnis sehen Sie auf dem Titelblatt. Falls Sie gerne nähere Informationen zu diesem Unterprogramm haben möchten, melden Sie sich bitte.

Anschließend noch das oben erwähnte Listing der Dame auf unserem Titelblatt. Ich wünsche Ihnen viel Erfolg beim Experimentieren mit der Ausgabe (Sie wissen ja, wohin mit den Ergebnissen). Meine weiteren Versuche auf diesem Gebiet werden wohl die Erstellung von Schriften sein (z.B. zum Satz unserer Überschriften im Journal) und letztendlich einmal die Schrifterkennung. Aber bis dahin ist sicher noch ein weiter Weg; denn die Kamera erzeugt meiner Ansicht nach weit mehr "Schmutz" als ein Mikrofon bei der Spracherkennung. Aber bisher haben wir schon einiges durchgekämpft, und so kommt sicher die Zeit, wo wir uns vom Rechner das 80-Bus Journal vorlesen lassen.



Selbstbau-Plotter Schriftweite 5



TANDY Line Printer 7 Nadeln gleichz.

ZEAP Z80 Assembler - Source Listing

```

00110 ; TV KAMERA INTERFACE
00200 ; NASSYS 3
00300 ; START #10000
00400 ; GÜNTER BOHM K'HE 9.11.82
00500 ;
-----
1000 00600 ORG #1000
1000 EF 00700 MENUE RST 40
1001 0C 00800 DEFB #C ;CLRS
1002 4155464E 00900 DEFB "AUFNEHMEN A
      45484D45
      4E202020
      20202020
      202041
1015 0D 01000 DEFB #D ;NL
1016 42494C44 01100 DEFB "BILDSCHIRM B
      53434B49
      524D2020
      20202020
      202042
1029 0D 01200 DEFB #D
102A 44525543 01300 DEFB "DRUCKEN D
      4B454E20
      20202020
      20202020
      202044
103D 0D000 DEFB #D
103F DF7E 01400 DEFB #7BDF ;BLINK
1041 FE41 01500 CP "A
1043 GADF10 01600 JP Z START1
1046 FE42 01700 CP "B
104B 2B0B 01800 JR Z BILD
104A FE44 01900 CP "D
104C CA1F12 02000 JP Z DRUCK
104F FE4E 02100 CP "N
1051 20AD 02200 JR NZ MENUE
1053 DF5B 02300 DEFB #5BDF ;MRET
-----
1055 EF 02400 ;
1056 0C00 02500 ;
1058 210020 02600 ;
105B 220E0C 02700 ;
105E CD3911 02800 ;
1061 210000 02900 ;
1064 220C0C 03000 ;
1067 DF7E 03100 BILD RST 40
      03200 DEFB #C ;CLRS
      03300 LD HL,#2000;SPEICHERBEG.
      03400 LD (#C0E),HL
      03500 CALL SCREEN
      03600 LD HL,00
      03700 LD (#C0C),HL
      03800 BLINK DEFW #7BDF
-----
;VERSCHIEBEN EINES TV BILDES AUF
;96X48 BILDSCHIRM 9.11.82 G.BHM
0390 0390 2A0C0C LD HL, (#C0C)
0400 0410 106C FE11 CP #11 ;CURSOR LEFT
0420 0430 1070 7D LD A,L ;PUNKT
0440 0450 1071 B7 OR A
0460 0470 1072 2BF3 JR Z BLINK
0480 0490 1074 2D LD L
0490 0500 1075 220C0C LD (#C0C),HL
0510 0520 107B 2A0E0C LD HL, (#C0E)
0530 0540 107C 2B DEC HL
0540 0550 107D 220E0C LD (#C0E),HL
0560 0570 1080 CD3911 CALL SCREEN
0580 0590 1083 1BE2 LD BLINK
0600 0610 1085 FE12 CP #12 ;CURSOR RIGHT
0620 0630 1087 2016 JR NZ UP
0640 0650 1089 7D LD A,L ;PUNKT
0660 0670 108A FE0A CP I0 ;RECHTER RAND
0680 0690 108C 2B09 JR Z BLINK
0700 0710 108E 2C INC L
0720 0730 108F 220C0C LD (#C0C),HL
0740 0750 1092 2A0E0C LD HL, (#C0E)
0760 0770 1095 23 INC HL
0780 0790 1096 23 INC HL
0800 0810 1097 220E0C LD (#C0E),HL
0820 0830 109A CD3911 CALL SCREEN
0840 0850 109D 18C8 JR BLINK
0860 0870 109F FE13 CP #13
0880 0890 10A1 2019 JR NZ DOWN
0900 0910 10A3 7C LD A,H ;ZEILE
0920 0930 10A4 R7 OR A
0940 0950 10A5 28C0 JR Z BLINK
0960 0970 10A7 25 DEC H
0980 0990 10AB 220C0C LD (#C0C),HL
1000 1010 10AB 2A0E0C LD HL, (#C0E)
1020 1030 10AE 110002 LD DE,512 ;B ZEILEN MAL 2
1040 1050 10B1 B7 OR A
1060 1070 10B2 ED52 SBC HL,DE
1080 1090 10B4 220E0C LD (#C0E),HL
1100 1110 10B7 CD3911 CALL SCREEN
1120 1130 10BA 18AB JR #790
1140 1150 10BC FE14 LD #14 ;CURSOR DOWN
1160 1170 10BE 201B JR NZ ESCAPE
1180 1190 10C0 7C LD A,H
1200 1210 10C1 FE07 CP 7 ;UNTERER RAND
1220 1230 10C3 2BA2 JR Z BLINK
1240 1250 10C5 24 INC H
1260 1270 10C6 220C0C LD (#C0C),HL
1280 1290 10C9 2A0E0C LD HL, (#C0E)
1300 1310 10CC 110002 LD DE,512 ;B ZEILEN DIFFERENZ MAL 2
1320 1330 10CF 19 ADD HL,DE
1340 1350 10D0 220E0C LD (#C0E),HL
1360 1370 10D3 CD3911 CALL SCREEN
1380 1390 10D6 18BF JR #1B
1400 1410 10D8 FE1B ESCAPE CP #1B
1420 1430 10DA 208B JR NZ BLINK
1440 1450 10DC C30010 JP MENUE

```

0960 ;			
0970 ;			
0980 ;	BILD SPEICHERN	A:207 ;PORTMODE	1480 OUT (60),A ;NAECHSTE ZAEHLUNG VORBER.
0990 ;	VER 1.0	Günter Böhm 17.10.82	1490 DJNZ ZEILE; SCHLEIFE FUER BILD
1000 ;			1500 JP MENUE ;END OF ROUT.
1010 ;			
1020 ;	START1	LD A,156 ;ZEILEN	
1030	LD A,172		
1040	OUT (62),A		
1050	START2	LD A,32 ;CLEAR RAM ZAEHLER	
1060	OUT (60),A		
1070	LD R,156 ;ZEILEN		
1080	LD HL,#2000 ;SPEICHERBEGINN		
1090	LD C,32 ; BMAL32 PUNKTE		
1100	LD A,50		
1110	OUT (60),A ; LOW TAKT ZEILE		
1120	LD A,51		
1130	OUT (60),A ;HIGH TAKT ZEILE		
1140	LD A,50		
1150	OUT (60),A ; LOW TAKT ZEILE		
1160 ;			
1170	LD A,54		
1180	OUT (60),A ;TAKTFREIGABE TV		
1190	IN A,(60)		
1200	BIT 6,A		
1210	JR Z, WAIT ; WARTEN BIS EINGELESEN (CLE		
1220	LD A,34		
1230	OUT (60),A ; CLEAR ZAEHLER		
1240 ;			
1250	PUNKTE	E,8 ;8 BITS	
1260	LD D,0 ;URZUSTAND 1 BYTE		
1270	LD A,42		
1280	LD A,34		
1290	OUT (60),A ; AUSLESETAKT		
1300	IN A,(60) ; AUSLESEN		
1310	BIT 7,A		
1320	AND #80 ;MASKE BIT 7		
1330	OR D ;MIT BYTE VERKNUEPFEN		
1340	LD Z, LAST		
1350	DEC E		
1360	JR A ;BIT NACH RECHTS		
1370	LD D,A		
1380	JR D		
1390	JR D		
1400	LD (HL),A		
1410	INC HL		
1420	DEC C		
1430	JR NZ PUNKTE		
1440 ;			
1450	LD A,18		
1460	OUT (60),A ; CLEAR RAM ZAEHLER		
1470	LD A,34		
10DF 3EFC			
10E1 D33E			
10E3 3E00			
10E5 D33E			
10E7 3E20			
10E9 D33C			
10EB 069C			
10ED 210020			
10F0 0E20			
10F2 3E32			
10F4 D33C			
10F6 3E33			
10F8 D33C			
10FA 3E32			
10FC D33C			
10FE 3E36			
1100 D33C			
1102 DR3C			
1104 CB77			
1106 2BFA			
AR)			
1108 3E22			
110A D33C			
110C 1E0B			
110E 1600			
1110 3E2A			
1114 3E22			
1116 D33C			
1118 DR3C			
111A CB7F			
111E E6B0			
111F 1D			
1120 2B05			
1122 CB3F			
1124 57			
1125 18E9			
1127 77			
1128 23			
1129 00			
112A 20E0			
112C 3E12			
112E D33C			
1130 3E22			
1132 D33C			
1134 10BA			
1136 C30010			
1139 210000			
113C 22100C			
113F 2A0E0C			
2			
1142 162F			
1144 0E0C			
1146 060B			
1148 7E			
1149 CD6111			
114C CB3F			
114E 10F9			
1150 23			
1151 0D			
1152 20F2			
1154 D5			
1155 11400			
1158 19			
1159 CD1312			
115C D1			
115D 15			
115E 20E4			
1160 C9			
1161 E5			
1162 F5			
1163 CB47			
1165 F5			
1166 C0FC11			
1169 F1			
116A 04E511			
116D F1			
116E E1			
116F C9			
1170 F5			
1171 C5			
1172 D5			
1173 E5			
1174 D710			
1176 47			
1620 LD D,47 ;ZEILEN			
1630 LD G,12 ;BYTES PRO ZEILE			
1640 LD B,8 ;7 MAL SHIFT			
1650 LD A,(HL)			
1660 CALL OUT ;B BITS AUSGEBEN			
1670 SRL A			
DJNZ BITS			
INC HL			
DEC C			
JR NZ BYTES			
1720 ;			
1730 LINES			
LD DE,20 ;DIFFERENZ NEXT LINE			
ADD HL,DE			
CALL CRLF			
POP DE			
DEC D			
JR NZ HORIZ			
RET ;UNTERPROGRAMM ENDE			
1810 ;			
1820 OUT			
PUSH HL ;SUBROUT. TESTEN U.AUSGEBEN			
PUSH AF			
BIT 0,A			
PUSH AF			
CALL Z SCHWAR			
POP AF			
CALL NZ WEISS ;HELLEN PUNKT SETZEN			
POP AF			
POP HL			
RET			
1920 ;			
1930 SET			
PUSH BC			
PUSH DE			
PUSH HL			
RCL AF			
LD B,A			
1940			
1950			
1960			
1970			
1980			

GRAFIK von M.Bach NJ2/82





```

120D E1 3050 POP HL
120E 2C 3060 INC L
120F 22100C 3070 LD (#C10),HL
1212 C9 3080 RET
1213 ED5B100C 3090 i
1217 14 3100 CRLF
1218 1E00 3110 LD DE, (#C10)
121A ED53100C 3120 INC D ;NEXT LINE
121E C9 3130 LD E,0
3140 LD (#C10),DE
3150 RET
3160 i
3170 ;PROGRAMM ZUR DRUCKAUSGABE TV BILD
3180 ;30.11.82 TANDY LINE PRINTER
3190 ;G. Böhm Karlsruhe
3200 DRUCK LD A,18 ;GRAFIK CODE
3210 CALL PRINT
3220 LD HL,#2000-32
3230 LD B,22 ;FAKTOR FUER 156 ZEILEN
3240 NXTLIN LD C,32 ;BYTES PRO Z.
3250 LD A,#D ;CRLF
3260 CALL PRINT
3270 PUSH BC
3280 NXTBYT CALL GETBYT
3290 POP BC
3300 INC HL
3310 INC C
3320 DEC JR
3330 LD DE,192 ;DIFFERENZ NEXT ZEILENBLOCK
3340 ADD HL,DE
3350 DJNZ NXTLIN
3360 JP MENUE ;ENDE DRUCKAUSGABE
3370 i
3380 GETBYT LD DE,0
3390 NXTBIT INC D ;BITNUMMER
3400 LD A,B
3410 CP D
3420 RET C
3430 PUSH HL
3440 GETBIT LD C,7 ;DRUCKNADELN
3450 BITNUM LD B,D
3460 PUSH DE
3470 LD DE,32 ;DIFF. NEXT ZEILE
3480 ADD HL,DE
3490 POP DE
3500 LD A,(HL)
3510 SHIFT SRL A
3520 DJNZ C WEI
3530 JR C WEI
3540 SET 7,E
3550 SRL E ;BIT 0 IN E GANZ RECHTS
3560 DEC C ;ZEILENBLOCK FERTIG?
3570 JR NZ BITNUM
1262 7B 1263 CBFF
1265 CD6B12
1268 E1
1269 18DA
126B F5
126C DB00
126E CB7F
1270 28FA
1272 F1
1273 DF6F
1275 FF
1276 C9
3580 LD A,E
3590 SET 7,A ;FUER GRAFIK BIT 7 HI
3600 CALL PRINT
3610 POP HL ;"ERSTE" ZEILE
3620 JR NXTBIT
3630 i
3640 PUSH AF ;A AN TANDY AUSGEBEN
3650 IN A,(0) ;LEITUNG AN PORT 0
3660 BIT 7,A ;BIT 7
3670 JR Z BUSY
3680 POP AF
3690 DEFM #6DFD ; SRLX
3700 RST #38 ; DELAY
3710 RET
3720 i
3730 i
1520 ;MODIFIKATION:GESAMTBILD Z.11.82
1530 ;AUSGABE EINES TV BILDES
1540 ;VON SPEICHER ZU BILDSCHIRM
1550 ;(KLOTZSCHEN GRAFIK M.BACH)
1560 ;VER 1.0 8.11.82 G.Böhm
1570 ;UNTERPROGR. FUR "BILD"
1580 i
1590 SCREEN LD HL,U ;AUSGANGSPUNKT ZEILE UND
1600 LD (#C10),HL ; PUNKT IN ARG3
1610 LD HL,(#0E) ;BEG. D. BILDSPEICHERS ARGZ
1620 LD D,47 ;ZEILEN
1630 HORIZ LD C,32 ;BYTES PRO ZEILE
1635 LD E,5
1640 BYTES LD B,8 ;7 MAL SHIFT
1650 LD A,(HL)
1660 BITS DEC E ;JEDES DRITTE BIT AUSGEBEN
1661 CALL Z 001
1670 SRL A
1680 DJNZ BITS
1690 INC HL
1700 DEC C
1710 JR NZ BYTES
1720 i
1730 LINES PUSH DE ;NEUE ZEILE
1740 LD DE,64 ;DIFFERENZ 3 ZEILEN WEITER
1750 ADD HL,DE
1760 CALL CRLF
1770 POP DE
1780 DEC D
1790 JR NZ HORIZ
1800 RET ;UNTERPROGRAMM ENDE
1900 POP HL
1905 LD E,3
1910 RET
1920 i

```

ZEAP Z80 Assembler - Source Listing

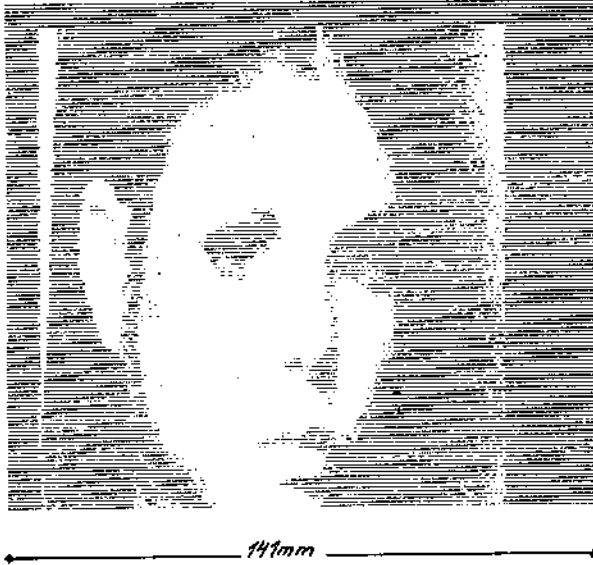
```

0010 ;DRUCKAUSGABE TV BILD MIT
0020 ;BLOCKGRAFIK TANDY VIII
0030 ;GUNTER BOHM, KARLSRUHE
0040 ;19.12.82
0050 ;
0060 PUFFER EQU #C80
0070 START LD HL,#2000 ;BILDSPEICHER
0080 LD C,78 ;DOPPELZEILEN/BILD
0090 ZEILE LD B,32 ;BYTES/ZEILE
0100 AUSG1 PUSH BC
0110 PUSH HL
0120 CALL HOLBYT
0130 LD B,4
0140 LD HL,PUFFER
0150 LD A,(HL)
0160 AUSG2 CALL CODE
0170 INC HL
0180 DJNZ AUSG2
0190 POP HL
0200 INC HL
0210 POP BC
0220 DJNZ 10EA
0230 LD A,#D ;CRLF
0240 CALL PRINT
0250 LD DE,32
0260 ADD HL,DE ;NEXT DOPPELZEILE
0270 DEC C
0280 JR NZ ZEILE
0290 DEFW #5BDF ;MRET
0300 ;
0310 ;
0320 ;HOLBYT LEGT DEN DRUCKCODE FÜR 1 BYTE
0330 ;IN 4 PUFFERADRESSEN AB
0340 HOLBYT LD B,4
0350 PUSH HL
0360 LD A,(HL)
0370 LD HL,PUFFER
0380 PUSH AF
0390 AND 3
0400 LD (HL),A
0410 POP AF
0420 SRL A
0430 SRL A
0440 INC HL ;NEXT PUFFER
0450 DJNZ BITS1
0460 POP HL
0470 LD DE,32 ;ZEILENDIFFERENZ
0480 ADD HL,DE
0490 LD B,4
0500 PUSH HL
2644 0C80
1000 210020
1003 0E4E
1005 0620
1007 C5
1008 E5
1009 CD2B10
100C 0604
100E 21800C
1011 7E
1012 CD6110
1015 23
1016 10F9
1018 E1
1019 23
101A C1
101B 10EA
101D 3E0D
101F CD8710
1022 112000
1025 19
1026 0D
1027 20DC
1029 DF5B
102B 0604
102D E5
102E 7E
102F 21800C
1032 F5
1033 E603
1035 77
1036 F1
1037 CB3F
1039 CB3F
103B 23
103C 10F4
103E E1
103F 112000
1042 19
1043 0604
1045 E5
1046 7E
1047 21800C
1048 E603
1049 CB27
104F 86
1051 2F
1052 E60F
1055 C6E0
1057 77
1058 F1
1059 CB3F
105B CB3F
105D 23
105E 10EA
1060 E1
1061 C9
1062 FEE4
1064 2003
1066 3D
1067 181F
1069 FEEB
106B 2003
106D 3C
106E 1818
1070 FEE3
1072 2820
1074 FEE5
1076 281C
1078 FEE7
107A 2818
107C FEE8
107E 2818
1080 FEE9
1082 2814
1084 FEED
1086 2810
1088 F5
1089 DR00
108B CB7F
108D 28FA
108F F1
1090 DF6F
1092 FF
1093 C9
1094 C604
1096 18F0
1098 D604
109A 18EC
0510 LD A,(HL)
0520 LD HL,PUFFER
0530 PUSH AF
0540 AND 3
0550 SLA A
0560 SLA A
0570 ADD A,(HL)
0580 CPL ;DRUCKER INVERSION D.BILDSCH.
0590 AND #F
0600 ADD A,#E0 ;CODE FÜR BLOCKGR.
0610 LD (HL),A
0620 POP AF
0630 SRL A
0640 SRL A
0650 INC HL
0660 DJNZ BITS2
0670 POP HL
0680 RET
0690 ;
0700 ;CODE WANDELT DIE CODES, DIE VON
0710 ;DER LOG. VIERERMATRIX ARBEITEN,
0720 ;IN DEN TANDY CODE UM
0730 CP #E4
0740 JR NZ NXTB
0750 DEC A
0760 JR PRINT
0770 CP #EB
0780 JR NZ NXTADD
0790 INC A
0800 JR PRINT
0810 CP #E3
0820 JR Z RADD
0830 CP #E5
0840 JR Z RADD
0850 CP #E7
0860 JR Z RADD
0870 CP #EB
0880 JR Z RSUB
0890 CP #E9
0900 JR Z RSUB
0910 CP #EC
0920 JR Z RSUB
0930 PUSH AF
0940 IN A,(0)
0950 BIT 7,A
0960 JR Z RUSY
0970 POP AF
0980 DEFW #6FDF ;SRLX
0990 RST #3B ;RDEL
1000 RET
1010 ADD A,4
1020 JR PRINT
1030 SUB 4
1040 JR PRINT
1050 ;

```



Im folgenden zwei Programme zur Druckausgabe des TV-Bildes mit verschiedenen Druckern. Klaus Mombaur programmierte in Maschine für einen ITOH, Clemens Ballarin in PASCAL für einen MX Drucker. Die Beispiele wurden mit einer Maßangabe versehen, um einen Größenvergleich zu haben. Vergleichen Sie dazu noch die Ausgabe meines Selbstbau-Plotters (steht immer noch zum Verkauf), die in etwa die richtigen Seitenverhältnisse zeigt.



FS - Bildplatten (auf ITOH 0510)

```

Drucker init.
0CDA 3E LD A,0F Port A1
0CDC D3 OUT (06),A Byteausgabe
0CDE 3E LD A,CF Port B1
0CE0 D3 OUT (07),A Bitmanipulation
0CE2 3E LD A,P5 Port B1 DB 1,3
0CE4 D3 OUT (07),A auf Ausgabe
0CE6 3E LD A,FF Port B1 DB 1,3
0CE8 D3 OUT (05),A auf H
0CEA C89F RES 3,A INPUT PRIME(neg)
0CEC D3 OUT (05),A kurz L
0CEE C8DF SET 3,A
0CF0 D3 OUT (05),A
0CF2 DF50 RST TDEL

Druckart progr.1
0CF4 Z1 LD HL,0E2F Tabellen-ADR
0CF7 CD CALL 0E00 -> U-Druckvorb
0CFA Z1 LD HL,0E3B Tabellen-ADR
0CFD CD CALL 0E00 -> U-Druckvorb

Hauptprogramm
0D00 Z1 LD HL,2000 Beg BS-Speicher
0D03 16 LD D,9C 156 Zeilen
0D05 0E LD C,20 32 Bytes/Zeile
0D07 06 LD B,07 7 Pkte/Byte
0D09 7E LD A,(HL)
0D0A CD CALL 0D30 -> U-Ausgabe

```

```

0D00 C83F SRL A naechster Pkt
0D0F 10 DJNZ F9(0D0A)
0D11 23 INC HL
0D12 00 DEC C naechstes Byte
0D13 20 JR NZ,F2(0D07)

0D15 3E LD A,00 Neue Zeile
0D17 CD CALL 0E10 -> U-Druck
0D1A E5 PUSH HL
0D1B 00 NOP
0D1C Z1 LD HL,0E3B Tabellen-ADR
0D1F CD CALL 0E00 -> U-Druckvorb
0D22 E1 POP HL
0D23 15 DEC D naechste Zeile
0D24 20 JR NZ,DF(0D05)
0D26 00 NOP
0D27 76 HALT Ende

```

U - Ausgabe

```

0D30 F5 PUSH AF
0D31 C84F BIT 1,A Testent:
0D33 20 JR Z,07(0D3C)
0D35 3E LD A,00 = ws
0D37 CD CALL 0E10 -> U-Druck
0D3A F1 POP AF
0D3B C9 RET
0D3C 3E LD A,01 = sw
0D3E 10 JR F7(0D37)

```

U - Druckvorb

```

0E00 C5 PUSH BC
0E01 0E LD C,04 Port A
0E03 06 LD B,0C Zeichenzahl
0E05 EDA3 OUTI
0E07 F5 PUSH AF
0E08 CD CALL 0E12 -> U-Druck +2
0E0B F1 POP AF
0E0C 20 JR NZ,F7(0E05) wdh
0E0E C1 POP BC
0E0F C9 RET

```

U - Druck

```

0E10 D3 OUT (04),A
0E12 D8 IN A,(05) /STROBE(neg)
0E14 C80F RES 1,A /
0E16 D3 OUT (05),A / kurz L
0E18 C8CF SET 1,A /
0E1A D3 OUT (05),A /
0E1C D8 IN A,(05) INPUT BUSY
0E1E C867 BIT 4,A pruefen
0E20 20 JR NZ,FA(0E1C)
0E22 C9 RET

```

Druck-Tabelle

```

0E2F 00
0E30 00 BS
0E31 0E S0 doppelte Breite
0E32 10 ESC
0E33 54 30 35 T 05
0E34 10 ESC
0E37 4C 30 31 30 L 010
0E3B 00 00 00 00 Dummy
0E40 10 ESC
0E41 53 30 32 32 34 S 0224

```

Start: 0CDA

Bildspeicher: 2000 - 3400

NASCOM 2 mit MAS-SYS 1



```

PROGRAM TV-DRUCK; (*15.11.82*)
(*Ausgabe eines G.Böhm-Bildes auf der Schreibmaschine.
von Michael Bach*)
CONST ZEILEN=156; BYTES=32; BILD-ANF=#8000;
VAR A-PUNKT, ZEILE, BYTE-NR, BIT-NR, MUSTER, MASKE, PUNKT: INTEGER;
JA: BOOLEAN;

```

```

BEGIN
(*'USER-OUT' auf Schreibmaschine schalten*)
MEM(.$C7B.):=0; MEM(.$C79.):=B3; MEM(.$C73.):=B78; JA:=FALSE;
FOR ZEILE:=0 TO ZEILEN DO BEGIN
A-PUNKT:=0;
FOR BYTE-NR:=4 TO BYTES-2 DO BEGIN
MUSTER:=MEM(.(ZEILE*BYTES+BYTE-NR)+BILD-ANF.);
MASKE:=1;
FOR BIT-NR:=0 TO 7 DO BEGIN
PUNKT:=MUSTER AND MASKE; MASKE:=MASKE*2;
IF PUNKT<>0 THEN PUNKT:=1; JA:=NOT JA;
(*nur Jeder 2. Punkt aber verschieden schwarz*)
IF JA THEN BEGIN
CASE (PUNKT+A-PUNKT) OF
0: WRITE('X');
1: WRITE('.');
2: WRITE(' ');
END; (*CASE*)
END;
A-PUNKT:=PUNKT;
END;
WRITELN; ZEILE:=ZEILE+1; (*nur Jede 2.*)
END;
MEM(.$C73.):=B79; (*'U' wieder aus*)
END.

```

```

PROGRAM TV-PLOT; (*15.11.82*)
(*Ausgabe eines G.Böhm-Bildes auf dem Bildschirm
von Michael Bach*)
LABEL 1;
CONST ZEILEN=156; BYTES=32; BILD-ANF=#8000;
VAR IX, IY, X, Y, X-TV, Y-TV, HELL, SCHWELLE, : INTEGER;
MASKENFELD: ARRAY(.0..7.) OF INTEGER;

```

```

FUNCTION PUNKT(PX, PY: INTEGER): INTEGER;
VAR BIT-NR, BYTE-NR, MASKE, MUSTER: INTEGER;
BEGIN
BYTE-NR:=PX SHIFT -3; BIT-NR:=PX-(BYTE-NR SHIFT 3);
MASKE:=MASKENFELD(.BIT-NR.);
MUSTER:=MEM(.BILD-ANF+(PY*BYTES+(BYTE-NR)).) AND MASKE;
IF MUSTER<>0 THEN PUNKT:=1 ELSE PUNKT:=0;
END; (*PUNKT*)

```

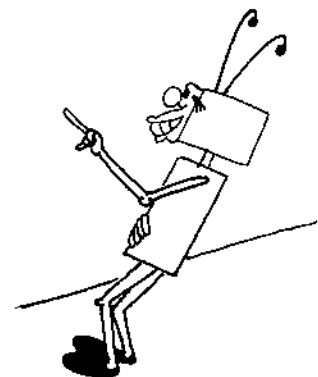
```

BEGIN
(*'USER-OUT' fuer Listing auf Schreibmaschine schalten*)
MEM(.$C7B.):=0; MEM(.$C79.):=B3;
WRITE(CHR(12));
INIT MASKENFELD TO 1,2,4,8,16,32,64,128;
SCHWELLE:=(2*3) DIV 2;
FOR Y:=0 TO 47 DO BEGIN
FOR X:=0 TO 95 DO BEGIN
(*Abbildung TV-Bild auf Klötzchengrafik*)
HELL:=0; IY:=(47-Y)*3-8;
FOR Y-TV:=IY TO IY+2 DO BEGIN
IX:=X*2+4B;
FOR X-TV:=IX TO IX+1 DO HELL:=HELL+PUNKT(X-TV, Y-TV);
END; (*FOR Y-TV*)
IF (HELL>=SCHWELLE) THEN PLOT(X, Y, 1);
END; (*FOR X*)
IF KEYBOARD THEN GOTO 1;
END; (*FOR Y*)
REPEAT (*Negativdarstellung*)
FOR X:=0 TO 3000 DO;
FOR Y:=0 TO 47 DO FOR X:=0 TO 95 DO PLOT(X, Y, 2);
UNTIL KEYBOARD;
1: ;
END.

```

Lieber Herr Bach,  
nun habe ich Ihnen schon wieder das Listing versaut!  
Der tiefgesetzte Strich in Ihren Variablen wurde durch  
einen Bindestrich ertsetzt, weil er für den Drucker  
den Einsatz zum Unterstreichen gibt. Einmal habe ich  
ihn vergessen. Das Ergebnis habe ich stehenlassen,  
damit Sie Stoff für einen Leserbrief haben. Ihr  
Nascomp!

**Nascom/ECB-Bus Baugruppen**  
Video 40 x 16 m. Blockgraphik  
/ inv., Bausatz kompl. 258DM.  
ECB-Bus Platine 10 Plätze,  
leer 56,--. Programmierkarte  
2508 - 64 u. 2716 - 128, leer  
90,--. Baus. 348,--. REP 16  
RAM/EPROM 61/2716, leer 79 DM  
Baus. o. Sp. 215,--. Monitor  
12", grün 280,--. ITOH-Drucker.  
Joachim List, Klaus Niemann  
Wiesbaden.



Das Programm GREY PLOT drucken wir jetzt schon ab, da es gut zu unserem Thema "Druckausgabe" paßt. Manche Leser können vielleicht schon so etwas mit dem Unterprogramm anfangen. Es druckt Intensitätswerte als Graustufen ab. Gerhard Klement hat damit den Ablauf einer Sonnenfinsternis in Indien dokumentiert. Sicher schickt uns G. Klement aber noch ein aufrufendes Programm zur Erläuterung dazu, möglicherweise mit Bildmaterial zur Demonstration. Falls das TV-Interface bald auch auf Graustufen anwendbar wäre, könnte dies das ideale Programm zur Druckausgabe werden.

```

10 REM == GREY PLOT ==
20 REM == IMAGE DISPLAY SUBROUTINE ==
30 :
40 REM =====
50 REM IA = IMAGE (64*64)
60 REM NX = NUMBER OF ROWS OF IA TO BE PRTD
70 REM NY = NUMBER OF COLS OF IA TO BE PRTD
80 REM ' NX,NY MAX 64
90 REM LA 1:LINEAR SCALE
100 REM LA 2:SQUARE ROOT SCALE
110 REM LA 3:LOGARTHMIC SCALE
120 REM LA 4:'ABSORPTION' SCALE
130 REM MI :MINIMUM GRAY LEVEL IN IA,
140 REM CALCULATED IN THE CALLING PGM
150 REM MA :MAXIMUM GRAY LEVEL IN IA,
160 REM CALCULATED IN THE CALLING PGM
170 REM NE :A VALUE EQUAL 1 GIVES THE NORMAL IMAGE,0 THE NEGATIVE IMAGE
180 REM
190 REM =====
200 :
210 REM -- SETUP --
220 :
230 REM -- GRAY CHARACTERS --
240 GR$(1)="MMMMMM#HHHHXHXOZWMNOS=I*+++=-.-"
250 GR$(2)="WWWWW#####+----"
260 GR$(3)="####00+-"
270 GR$(4)="+"
280 :
290 GN=32
300 REM -- Calc. factors --
310 K1=(MA-MI)/GN
320 K2=(SQR(MA)-SQR(MI))/GN
330 K3=(MA-MI)/LOG(GN+1)
340 D=1:IFMI>1THEND=MI
350 K4=-LOG(MA/D)/GN
360 REM -- Calc. gray scale --
370 FORI=1TO32
380 ONLAGOTO390,400,410,420
390 LL(I)=MI+(I-1)*K1:GOTO430
400 LL(I)=(SQR(MI)+(I-1)*K2)^2:GOTO430
410 LL(I)=MI+K3*LOG(I):GOTO430
420 LL(I)=MA*EXP(K4*(GN-I))
430 NEXT
440 REM -- Check max. --
450 IFNX>64THENNXX=64
460 IFNY>64THENNY=64
470 REM -- Setup IB gray level array --
480 FORI=1TONX:FORJ=1TONY:D=1
490 FORK=1TO32
500 IFIA(I,J)>=LL(K)THEND=K
510 NEXTK:IB(I,J)=D:NEXTJ,I
520 REM -- Output line by line --
530 FORI=1TONX
540 FORJ=1TO5:BL(J)=0:NEXTJ
550 FORL=1TO5:LI$(L)="":NEXT
560 FORK=2TO2*NYSTEP2
570 J=K/2:NG=IB(I,J)
580 REM -- Make neg. if wanted --
590 IFNE=0THENNG=33-NG

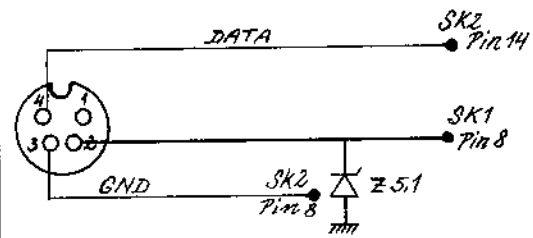
```

```

600 FORL=1TO5:IFNG<>32THENBL(L)=1
610 D#=MID$(GR$(L),NG,1)
620 LI$(L)=LI$(L)+D#+D#
630 NEXTL,K:PRINTCHR$(10)
640 FORL=1TO5:IFBL(L)=0THEN670
650 FORQ=1TO200:NEXT
660 PRINTLI$(L)CHR$(13);
670 NEXTL:NEXTI
680 PRINTCHR$(10)CHR$(10)CHR$(10)
690 PRINTCHR$(10)CHR$(10)CHR$(10)
700 RETURN
ok

```

Nun müssen wir uns aber langsam bei allen Lesern entschuldigen, die sich nicht für Grafik interessieren. Vielleicht zieht das Argument, daß wir, um Überschriften zu sparen (Fotosatz ist teuer), alles unter den Titel TV-Bild packen mußten, was uns dazu vorlag. Nun noch schnell zum Abschluß eine Information über den Anschluß eines TANDY VIII Printers, den ich kürzlich günstig erwerben konnte.



Wie P. Urban (N.Journal Okt,82) stellte sich mir die Frage: Wohin mit der Busy-Leitung? Hardwaremäßig habe ich das Problem einfacher gelöst: Die Busy-Leitung (die die Werte +-12V annimmt) wurde einfach über eine Zenerdiode (5,1V) an Bit 7 von Port0 angehängt. Nun muß bei der Ausgabe natürlich der Zustand des Ports abgefragt werden, aber das schadet nichts, da eine direkte Ausgabe über den UART wegen der Notwendigkeit einer Code-Umsetzung bei manchen Zeichen nicht möglich ist. Den Takt für die serielle Druckausgabe beziehe ich übrigens von dem FSK-Modem (N.Journal 11/12-81). Wer an einem Menue-Programm für den TANDY VIII interessiert ist (zur Einstellung der Schriftarten, Walzenvorlauf etc.), erhält von mir gegen Erstattung der Kopierkosten und Porto gerne ein vollständiges Assemblerlisting.

Günter Böhm

# SORT

von CHRISTIAN PETER

Im Journal 3-82 wurde ein Datenverwaltungs-Programm vorgestellt. Das Programm "SORT" kann zur Erweiterung dienen. Dies ist zunächst die Version, die ich schon fertig habe und die auch schon einige Zeit problemlos läuft. Ich möchte (wenn ich die Zeit finde) noch folgende Zusätze programmieren:

- Sortierfolge laut Tabelle (damit auch Umlaute richtig einsortiert werden)
- Sortierung mehrerer Felder (wenn Gleichheit im ersten Feld: Sort nach zweitem Feld; z.B.: Beier Franz vor Beier Otto, ...)

Es ist meiner Meinung nach auf jeden Fall notwendig das Source-Listing abzudrucken, damit jeder die nötigen Änderungen für seine Bedürfnisse machen kann.

## PROGRAMMBESCHREIBUNG

Das Programm "SORT" ist ein positions-unabhängiger Bubble-Sort (d.h. es ist relocierbar oder wie man das auch immer ausdrücken will). Es benötigt ca. 512 Byte Hauptspeicher und einen daran anschließenden Platz für Variablen in der Größe von 24 Byte.

Folgende Konventionen wurden für die Programmbeschreibung eingehalten:

Die Datei wird als "File" bezeichnet, ein einzelner Eintrag heißt "Record", die einzelnen Felder heißen "Fields". Das Programm kann Files beliebiger Größe sortieren, solange sie in den Hauptspeicher passen und jeder Record maximal 256 Byte lang ist.

Bevor das Programm aufgerufen wird, müssen zuerst die Feldbestimmungen definiert werden. Das Programm ist so ausgelegt, daß es möglichst große Freiheiten erlaubt, wo ein Field im Record steht.

Folgende Variablen müssen eingegeben werden:

- \* START Startadresse des Files im Hauptspeicher
- \* END Endadresse + 1(!)
- \* RECSEP ASCII-Zeichen zur Trennung der einzelnen Records
- \* FLDSEP ASCII-Zeichen zur Trennung der einzelnen Fields
- \* POSMRK ASCII-Zeichen, das die Startposition des zu sortierenden Feldes markiert (z.B. Blank) entweder +1, 0 oder -1
- \* DISPLC gibt an, in welcher Richtung vom gefundenen FLDSEP das Feld liegt
- \* FLDCNT gibt an, vom wievielten FLDSEP aus in Richtung DISPLC nach POSMRK gesucht wird

Das Programm findet das Feld, nach dem der File sortiert werden soll, indem es zuerst den nächsten RECSEP sucht, von dort aus der Anzahl von FLDCNT entsprechend nach FLDSEP's sucht. Wenn DISPLC=0, dann ist das Feld gefunden. Wenn DISPLC=-1, dann sucht das Programm rückwärts bis zum nächsten Zeichen POSMRK, das erste Zeichen nach POSMRK ist dann das erste Zeichen des Feldes. Wenn DISPLC=+1, dann sucht das Programm POSMRK in aufsteigender Richtung.

Wenn zwei Records ausgetauscht werden (sortieren), wird der erste Record anschließend an das Ende des Files zwischengespeichert, während der zweite Record an die Stelle des ersten kopiert wird. Es ist daher nötig, daß hinter dem File noch Platz (ca. 256 Bytes) gelassen wird.

Während des Sortiervorganges läuft ein Zähler mit, damit man weiß, daß das Programm auch wirklich läuft. Am Ende des Sortiervorganges erscheint die Meldung "END OF SORT".

## BEISPIEL FÜR DIE DEFINITION DER FELDER

Es soll eine Adressdatei sortiert werden, die folgenden Aufbau aufweist:

+ /NAME/STRASSE/ORT/ + /NAME/STRASSE/ORT/ + ...

- \* "+" ist in dem Fall der Recordseparator RECSEP und wird mit 2B definiert
- \* "/" steht für Carriage-return und ist der Fieldseparator; FLDSEP ist daher 0D und wird mit 0D definiert

Die Datei soll nun alphabetisch sortiert werden. Das Feld NAME kann folgenden Inhalt haben: event. TITEL, VORNAME, NACHNAME. Um den Nachnamen zu finden, können wir nicht von vorne anfangen zu suchen, weil wir nicht wissen, ob ein Titel vorhanden ist oder nicht. Aber es ist bekannt, daß der Nachname direkt vor dem zweiten FLDSEP steht. Daher definieren wir FLDCNT mit 2 und DISPLC mit -1 und POSMRK mit 20H (wenn VORNAME und NACHNAME durch ein Blank getrennt sind).

## BEMERKUNGEN

Es ist für diesen Sort unerheblich, ob alle Records gleiche Länge haben. Wichtig ist nur, daß alle Records, soweit sie die Feldbestimmungen betreffen, gleich aufgebaut sind.

Es werden keinerlei Fehlermeldungen ausgegeben, wenn die Feldbestimmung nicht zutreffen sollten. Wichtig ist auch, daß die Variable START direkt auf ein RECSEP-Zeichen zeigt und die Variable END auf das nächste Zeichen nach dem letzten RECSEP.

## EINGABE DER VARIABLEN

Das Programm wurde (wie gesagt) positions-unabhängig geschrieben. Alle Variablen werden daher relativ zum Programmbeginn abgespeichert. Um nicht immer ausrechnen zu müssen, wo jetzt was zu stehen hat, werden die Variablen am Programmbeginn automatisch in den Datenbereich übertragen. An dieser Stelle, kann man daher seine Werte eingeben (Zeilen 330 bis 420 im Source-Listing).

## ANMERKUNG ZUM PROGRAMMAUFBAU

Das Source-Listing mutet vielleicht ein bißchen seltsam an, wenn man es genauer betrachtet:

Es sind einige Sprünge im Programm, die wieder auf Sprünge führen. Diese Technik wurde deshalb angewandt, um alle Unterprogramm-Aufrufe relativ machen zu können. Nachdem die relative Distanz aber beschränkt ist, konnte das nur mit Hilfe von solchen Brücken realisiert werden.

## ÄNDERUNGEN

Soll das Programm als Unterprogramm laufen, dann müssen vor dem Aufruf alle Register gerettet werden (einschließlich IX) und die Zeile 2390 "SCAL 5BH" muß durch mehrere POPs und durch ein RET ersetzt werden. Wenn der Zähler nicht mitlaufen soll, müssen die Zeilen 2210 bis 2280 weggelassen werden.





ZEAP Z80 Assembler - Source Listing

```

1010 ;PROGRAM "SORT" VERS. 3.4
1020 ;FOR NAS-SYS
1030 ;@ BY CH. PETER

1000 ORG 1000H
1000 ENT
1000 START EQU 00H
1000 END EQU START+2
1000 RECSEP EQU END+2
1000 FLDSEP EQU RECSEP+1
1000 POSMRK EQU FLDSEP+1
1000 DISPLC EQU POSMRK+1
1000 FLDONT EQU DISPLC+1
1000 FLAG EQU FLDONT+1
1000 FIELD1 EQU FLAG+1
1000 FIELD2 EQU FIELD1+2
1000 FLD1LN EQU FIELD2+2
1000 FLD2LN EQU FLD1LN+1
1000 CNT EQU FLD2LN+1
1000 RCREG1 EQU CNT+2
1000 RCREG2 EQU RCREG1+2
1000 RLEN1 EQU RCREG2+2
1000 RLEN2 EQU RLEN1+1
1000 CURSOR EQU 0C29H

1270 ;WHERE AM I
1280 RCAL FNDLOC
1290 FNDLOC POP IX ;POP RETURN ADR. INTO IX
1300 LD BC,01FEH
1310 ADD IX,BC

1330 ;INIT VARIABLES
1009 DD3600FF 1340 LD (IX+START),0FFH
100D DD36012F 1350 LD (IX+START+1),2FH
1011 DD36042A 1360 LD (IX+RCSEP),*
1015 DD360802 1370 LD (IX+FLDONT),2
1019 DD36050D 1380 LD (IX+FLDSEP),0DH
101D DD360620 1390 LD (IX+POSMRK),20H
1021 DD3607FF 1400 LD (IX+DISPLC),-1
1025 DD36025C 1410 LD (IX+END),5CH
1029 DD360359 1420 LD (IX+END+1),59H

103D 1849 JR MAIN1
102F F5 1470 RECORD PUSH AF
1030 C5 1480 PUSH BC
1031 D5 1490 PUSH DE
1032 E5 1500 PUSH HL
1033 DD4E0A 1510 LD L,(IX+FIELD1)
1036 DD660B 1520 LD H,(IX+FIELD1+1)
1039 010001 1530 LD BC,01000H
103C DD7E04 1540 LD A,(IX+RCSEP)

1550 CPDR ;GO BACK UNTIL REG.O.P
INC HL ;POINT HL TO REG.O.REC
INC HL
LD (IX+RCREG1),L ;STORE HL
LD (IX+RCREG1+1),H
LD BC,00FFH ;MAX. CHAR. TO SEARCH
CPDR ;SEARCH END OF REC
LD A,C ;GET BYTE COUNT
CPL ;CONVERT TO POSITIV
LD (IX+RLEN1),A
LD L,(IX+FIELD2)
LD H,(IX+FIELD2+1)
LD BC,0100H
LD A,(IX+RCSEP)
CPDR
INC HL
INC HL
LD (IX+RCREG2),L
LD (IX+RCREG2+1),H
LD BC,00FFH
CPDR
LD A,C
CPL
LD (IX+RLEN2),A
POP HL
POP DE
POP BC
POP AF
RET

1850 MAIN1 JR MAIN2
1860 RECORD1 JR RECORD
1880 SWAP
1890 PUSH AF
1900 PUSH BC
1910 PUSH DE
1920 LD L,(IX+RCREG1) ;LD HL,(RCREG1)
1930 LD H,(IX+RCREG1+1)
1940 LD E,(IX+END) ;LD DE,(END)
1950 LD D,(IX+END+1)
1960 LD B,00
1970 LD C,(IX+RLEN1)
1980 LDIR ;COPY REC1 TO END O.FIL
1990 LD L,(IX+RCREG2)
2000 LD H,(IX+RCREG2+1)
2010 LD E,(IX+RCREG1)
2020 LD D,(IX+RCREG1+1)
2030 LD B,0
2040 LD C,(IX+RLEN2)
2050 LDIR ;MOVE RECORD 2
2060 LD L,(IX+END) ;POINT TO REC. 1
2070 LD H,(IX+END+1)
2080 LD B,0
2090 LD C,(IX+RLEN1) ;GET LENGTH
2100 LDIR ;MOVE REC. 1

```

```

10B3 E1 2110 POP H.
10B4 D1 2120 POP DE
10E5 C1 2130 POP BC
10E6 F1 2140 POP AF
10B7 C9 2150 RET

10B8 1874 2170 MAIN2 JR MAIN
10E8 1880 2180 SWAP1 JR SWAP
10E8 188C 2190 RECD2 JR RECD1

2210 CMPARE PUSH AF
2220 PUSH BC
2230 PUSH DE
2240 PUSH HL
2250 LD L,(IX+FIELD2)
2260 LD H,(IX+FIELD2+1)
2270 RCAL LENGTH
2280 LD (IX+FLD2LN),C
2290 LD L,(IX+FIELD1)
2300 LD H,(IX+FIELD1+1)
2310 RCAL LENGTH
2320 LD (IX+FLD1LN),C
2330 LD A,(IX+FLD2LN)
2340 LD B
2350 JR NC,TAB3
2360 LD C,A
2370 TAB3 LD B,0
2380 LD E,(IX+FIELD2)
2390 LD D,(IX+FIELD2+1)
2400 LOOP2 LD A,(DE)
2410 LD (HL)
2420 CP C,CARRY
2430 JR NZ,NOCARY
2440 INC HL
2450 INC DE
2460 DEC C
2470 JR NZ,LOOP2
2480 LD A,(IX+FLD2LN)
2490 CP (IX+FLD1LN)
2500 JR C,CARRY
2510 NOCARY POP HL
2520 POP DE
2530 POP BC
2540 POP AF
2550 AND A
2560 RET

2580 SWAP2 JR SWAP1
2590 CMPR1 JR CMPARE
2600 RECD3 JR RECD2

2620 CARRY POP HL
2630 POP DE
2640 POP BC
2650 POP AF
2660 SCF
;SET CARRY

1100 1888
1102 188A
1104 1886

1106 E1
1107 D1
1108 C1
1109 F1
110A 37

10B3 E1 2110 POP H.
10B4 D1 2120 POP DE
10E5 C1 2130 POP BC
10E6 F1 2140 POP AF
10B7 C9 2150 RET

10B8 1874 2170 MAIN2 JR MAIN
10E8 1880 2180 SWAP1 JR SWAP
10E8 188C 2190 RECD2 JR RECD1

2210 CMPARE PUSH AF
2220 PUSH BC
2230 PUSH DE
2240 PUSH HL
2250 LD L,(IX+FIELD2)
2260 LD H,(IX+FIELD2+1)
2270 RCAL LENGTH
2280 LD (IX+FLD2LN),C
2290 LD L,(IX+FIELD1)
2300 LD H,(IX+FIELD1+1)
2310 RCAL LENGTH
2320 LD (IX+FLD1LN),C
2330 LD A,(IX+FLD2LN)
2340 LD B
2350 JR NC,TAB3
2360 LD C,A
2370 TAB3 LD B,0
2380 LD E,(IX+FIELD2)
2390 LD D,(IX+FIELD2+1)
2400 LOOP2 LD A,(DE)
2410 LD (HL)
2420 CP C,CARRY
2430 JR NZ,NOCARY
2440 INC HL
2450 INC DE
2460 DEC C
2470 JR NZ,LOOP2
2480 LD A,(IX+FLD2LN)
2490 CP (IX+FLD1LN)
2500 JR C,CARRY
2510 NOCARY POP HL
2520 POP DE
2530 POP BC
2540 POP AF
2550 AND A
2560 RET

2580 SWAP2 JR SWAP1
2590 CMPR1 JR CMPARE
2600 RECD3 JR RECD2

2620 CARRY POP HL
2630 POP DE
2640 POP BC
2650 POP AF
2660 SCF
;SET CARRY

1100 1888
1102 188A
1104 1886

1106 E1
1107 D1
1108 C1
1109 F1
110A 37

```

```

;INIT BYTE-COUNT

```

```

2670 RET
2680 PUSH AF
2690 PUSH DE
2700 PUSH HL
2710 LD E,(IX+FLDSEP)
2720 LD D,(IX+POSMRK)
2730 LD C,0
2740 LD A,(HL)
2750 CP D
2760 JR Z,END2
2770 CP E
2780 JR Z,END2
2790 CP Z,END2
2800 CP Z,END2
2810 CP 0DH
2820 JR Z,END2
2830 INC BC
2840 INC HL
2850 JR TAB2
2860 POP HL
2870 POP DE
2880 POP AF
2890 RET

2910 MAIN LD (IX+CNT),0
2920 LD (IX+CNT+1),0
2930 LD (IX+FLAG),0
2940 LD L,(IX+START)
2950 LD H,(IX+START+1)
2960 RCAL LOCATE
2970 LD (IX+FIELD1),L
2980 LD (IX+FIELD1+1),H
2990 RCAL LOCATE
3000 LD (IX+FIELD2),L
3010 LD (IX+FIELD2+1),H
3020 RCAL RECD3
3030 RCAL CMPR1
3040 JR NC,TAB4
3050 RCAL SWAP2
3060 LD (IX+FLAG),1
3070 LD L,(IX+RCBEG2)
3080 LD H,(IX+RCBEG2+1)
3090 LD C,(IX+RCLENZ)
3100 LD B,0
3110 SCF
3120 ADC HL,BC
3130 LD C,(IX+END)
3140 LD B,(IX+END+1)
3150 AND A
3160 SBC HL,BC
3170 JR C,NEXT
3180 LD A,(IX+FLAG)
3190 AND A
3200 JR Z,END3
3210 LD HL,080AH
3220 LD (CURSOR),HL

```

```

;INIT CNT
;RESET FLAG
;INIT HL
;FIND FIELD
;STORE HL
;FIND NEXT FIELD
;CHECK FIELDS
;JUMP IF OK
;EXCHANGE RECORDS
;SET SORT FLAG
;CALCULATE E.O. REC2+J
;GET EOF ADDRESS
;RESET CARRY
;EOF REACHED?
;IF NOT EOF --> NEXT
;EOF REACHED
;TEST A
;IF FLAG=0 --> END

```

```

112E DD361000 2910 MAIN
1132 DD361100 2920
1136 DD360900 2930 LOOP3
113A DD06E000 2940
113D DD066001 2950
1140 DD77500A 2960
1142 DD7500A 2970
1145 DD7400B 2980
1148 DD76E 2990
114A DD7500C 3000
114D DD7400D 3010
1150 DD7E2 3020
1152 DD7AE 3030
1154 DD3005 3040
1156 DD7AB 3050
1158 DD360901 3060
115C DD6E14 3070 TAB4
115F DD6615 3080
1162 DD4E17 3090
1165 DD600 3100
1167 37 3110
1168 ED4A 3120
116A DD4E02 3130
116D DD4603 3140
1170 A7 3150
1171 ED42 3160
1173 381D 3170
1175 DD7E09 3180
1178 A7 3190
1179 281F 3200
117E 210A0B 3210
117E 22290C 3220

```

```

;IF A>=C JUMP
;GET MIN. LENGTH
;INIT B
;POINT DE TO FLD2
;GET BYTE OF FLD 2
;COMP. WITH FLD 1
;NO DIFF. FOUND
;CHECK LENGTH
;IF FLD2LN < FLD1LN

```

```

;RESET CARRY

```

```

;SET CARRY

```

```

1181 DD6E10      3230      LD      L,(IX+CNT)
1184 DD6611      3240      LD      H,(IX+CNT+1)
1187 23         3250      INC     HL
1188 DD7510      3260      LD      (IX+CNT),L
118B DD7411      3270      LD      (IX+CNT+1),H
118E DF66        3280      SCAL   66H
1190 18A4        3290      JR      LOOP3
1192 DD6E12      3300      LD      L,(IX+RCREG1)
1195 DD6613      3310      LD      H,(IX+RCREG1+1)
1198 18A6        3320      JR      LOOP4
119A 210F08      3330      LD      HL,0B0FH
119D 22290C      3340      LD      (CURSOR),HL
11A0 EF         3350      RST    28H
11A1 3E3E3E20    3360      DEFM   />>>: END OF SORT <<</

11E4 00         3370      DEFB   0DH
11E5 00         3380      DEFB   0
11E6 DF5E        3390      SCAL   5EH
;RETURN TO MONITOR

11B8 F5         3410      LOCATE PUSH AF
11E9 C5         3420      PUSH  BC
11EA D5         3430      FUSH  DE
11EB 010001     3440      LD      BC,0100H
11BE DD7E04     3450      LD      A,(IX+RCSEP)
11C1 EDB1       3460      CP     IR ;FIND NEXT REGSEP
11C3 DD5608     3470      LD      D,(IX+FLDCNT)
11C5 7A         3480      LD      A,D
11C7 A7         3490      AND    A
11C8 280E       3500      JR      Z,TAB1 ;IF ZERO --> JUMP
11CA 010001     3510      LOOP1
11CD DD7E05     3520      LD      LD BC,0100H
11D0 EDB1       3530      CP     IR
11D2 15         3540      DEC    D
11D3 18F3       3550      JR      LOOP1
11D5 DD7E07     3560      TAB1  LD      A,(IX+DISPLOC) ;SET DISPLACEMENT
11D8 A7         3570      AND    A ;A=0?
11D9 2818       3580      JR      Z,END1
11DB FE80       3590      CP     80H
11DD 380C       3600      JR      C,POSITV
11DF DD7E05     3610      LD      A,(IX+POSMRK)
11E2 010001     3620      LD      BC,0100H
11E5 EDB9       3630      CP     DR ;FIND POSMRK BACKW.
11E7 23         3640      INC    HL
11E8 23         3650      INC    HL
11E9 1808       3660      JR      END1
11EB DD7E06     3670      LD      A,(IX+POSMRK)
11EE 010001     3680      LD      BC,0100H
11F1 EDB1       3690      CP     IR ;FIND POSMRK FWD
11F3 D1         3700      POP    DE
11F4 C1         3710      POP    BC
11F5 F1         3720      POP    AF
11F6 C9         3730      RET

```

Anmerkung der Redaktion

Beim Test des Sort traten Schwierigkeiten auf, die von Christian Peter schon vorhergesehen wurden: in einigen Records meiner Datei waren die Feldbestimmungen nicht genau eingehalten worden. So war einmal vor einen Nachnamen nur ein Anfangsbuchstabe angehängt, im Namensfeld stand nur ein Name (z.B. eine Firma) oder das A0 (Graph+Space) war versehentlich als einfacher Space eingegeben. Besonders letzteres trat einermals auf, worauf ich wohl die Vereinbarungen zur Feldtrennung (NJ 3/4-82 Dateiverwaltung) nochmals überdenken und vereinfachen werde. Grundsätzlich scheint es aber sowieso sinnvoll, die Records auf richtigen Aufbau zu untersuchen, bevor die Datei durch den SORT zerstört wird. Das folgende Miniprogramm (auf die Konventionen von DATEI eingeteilt) könnte auch direkt in den Sort als Sicherung eingebaut werden. Dann müßte man aber noch weitere Fehlermöglichkeiten einplanen. In der abgedruckten Form stellt es nur fest, wenn ein Feld kein Posmark enthält oder wenn nicht genügend Felder vorhanden sind.

Das zweite Listing zeigt eine Möglichkeit, den Sort durch ein Menue aufzurufen, so daß man nicht per Modify die gewünschten Suchkriterien mühsam eingeben muß. Der Sort kann so direkt für DATEI verwendet werden.

ZEAP Z80 Assembler - Source Listing

```

0010 ;TEST auf korr. Strings für SORT
0020 ;27.12.82 G.B.
0030 ORG #C80
0040 LD HL,#2800 ;START
0050 LD DE,#5748 ;ENDE
0060 LD BC,0 ;COUNTER A0/SPACE
0070 LD A,(HL)
0080 CP #A0
0090 JR NZ SPACE
0100 INC B ;A0 ZAEHLEN
0110 SPACE CP #20
0120 JR NZ BELL
0130 INC C ;SPACE ZAEHLEN
0140 LD A,B
0150 OR A
0160 JR Z ERROR1 ;WENN KEIN A0 IN FELT

```

```

0098 0600 LD B,0 ;INIT NEXT FIELD
009A FE07 CP 7
009C 200B JR NZ NEXT
009E 79 LD A,C
009F FE03 CP 3
00A1 201D JR NZ ERROR2 ;WENN NICHT 3 SPACES IM R
EC.
00A3 010000 LD RC,0 ;INIT NEXT RECORD
00A6 23 INC HL
00A7 E5 PUSH HL
00AB B7 OR A
00AC ED52 SBC HL,DE
00AD E1 POP HL
00AF 38DB JR C LOOP
00B0 DF5B DEFW #5BDF ;MRET
00B1 0D RST #28
00B2 4B45494E DEFB #D
20413020 DEFM *KEIN A0 *
00BB 00 DEFB 0
00BC DF66 DEFW #66DF ;TBCD3
00BE 1811 JR NXTRCD
00C0 EF ERROR2 RST #28
00C1 0D DEFB #D
00C2 4B45494E DEFM *KEIN SPACE *
20535041
43452020
00CE 00 DEFB 0
00CF DF66 DEFW #66DF
00D1 23 NXTRCD INC HL
00D2 7E LD A,(HL)
00D3 FE07 CP 7
00D5 20FA JR NZ NXTRCD
00D7 23 INC HL
00D8 18AC JR INIT
0170 0170 BELL
0180 0190 NEXT
0200 0210 ERROR1
0220 0230 ERROR2
0240 0250 NEXT
0260 0270 SBC HL,DE
0280 0290 POP HL
0300 0310 JR C LOOP
0320 0330 DEFW #5BDF ;MRET
0340 0350 DEFB #D
0360 0370 DEFM *KEIN A0 *
0380 0390 DEFB 0
0400 0410 DEFB 0
0420 0430 NXTRCD INC HL
0440 0450 LD A,(HL)
0460 0470 CP 7
0480 0490 JR NZ NXTRCD
0500 0510 INC HL
0520 0530 JR INIT
0540 0550
0560 0570
0580 0590
0600 0610
0620 0630
0640 0650
0660 0670
0680 0690
0700 0710
0720 0730
0740 0750
0760 0770
0780 0790
0800 0810
0820 0830
0840 0850
0860 0870
0880 0890
0900 0910
0920 0930
0940 0950
0960 0970
0980 0990
1000 1010
1020 1030
1040 1050
1060 1070
1080 1090
1100 1110
1120 1130
1140 1150
1160 1170
1180 1190
1200 1210
1220 1230
1240 1250
1260 1270
1280 1290
1300 1310
1320 1330
1340 1350
1360 1370
1380 1390
1400 1410
1420 1430
1440 1450
1460 1470
1480 1490
1500 1510
1520 1530
1540 1550
1560 1570
1580 1590
1600 1610
1620 1630
1640 1650
1660 1670
1680 1690
1700 1710
1720 1730
1740 1750
1760 1770
1780 1790
1800 1810
1820 1830
1840 1850
1860 1870
1880 1890
1900 1910
1920 1930
1940 1950
1960 1970
1980 1990
2000 2010
2020 2030
2040 2050
2060 2070
2080 2090
2100 2110
2120 2130
2140 2150
2160 2170
2180 2190
2200 2210
2220 2230
2240 2250
2260 2270
2280 2290
2300 2310
2320 2330
2340 2350
2360 2370
2380 2390
2400 2410
2420 2430
2440 2450
2460 2470
2480 2490
2500 2510
2520 2530
2540 2550
2560 2570
2580 2590
2600 2610
2620 2630
2640 2650
2660 2670
2680 2690
2700 2710
2720 2730
2740 2750
2760 2770
2780 2790
2800 2810
2820 2830
2840 2850
2860 2870
2880 2890
2900 2910
2920 2930
2940 2950
2960 2970
2980 2990
3000 3010
3020 3030
3040 3050
3060 3070
3080 3090
3100 3110
3120 3130
3140 3150
3160 3170
3180 3190
3200 3210
3220 3230
3240 3250
3260 3270
3280 3290
3300 3310
3320 3330
3340 3350
3360 3370
3380 3390
3400 3410
3420 3430
3440 3450
3460 3470
3480 3490
3500 3510
3520 3530
3540 3550
3560 3570
3580 3590
3600 3610
3620 3630
3640 3650
3660 3670
3680 3690
3700 3710
3720 3730
3740 3750
3760 3770
3780 3790
3800 3810
3820 3830
3840 3850
3860 3870
3880 3890
3900 3910
3920 3930
3940 3950
3960 3970
3980 3990
4000 4010
4020 4030
4040 4050
4060 4070
4080 4090
4100 4110
4120 4130
4140 4150
4160 4170
4180 4190
4200 4210
4220 4230
4240 4250
4260 4270
4280 4290
4300 4310
4320 4330
4340 4350
4360 4370
4380 4390
4400 4410
4420 4430
4440 4450
4460 4470
4480 4490
4500 4510
4520 4530
4540 4550
4560 4570
4580 4590
4600 4610
4620 4630
4640 4650
4660 4670
4680 4690
4700 4710
4720 4730
4740 4750
4760 4770
4780 4790
4800 4810
4820 4830
4840 4850
4860 4870
4880 4890
4900 4910
4920 4930
4940 4950
4960 4970
4980 4990
5000 5010
5020 5030
5040 5050
5060 5070
5080 5090
5100 5110
5120 5130
5140 5150
5160 5170
5180 5190
5200 5210
5220 5230
5240 5250
5260 5270
5280 5290
5300 5310
5320 5330
5340 5350
5360 5370
5380 5390
5400 5410
5420 5430
5440 5450
5460 5470
5480 5490
5500 5510
5520 5530
5540 5550
5560 5570
5580 5590
5600 5610
5620 5630
5640 5650
5660 5670
5680 5690
5700 5710
5720 5730
5740 5750
5760 5770
5780 5790
5800 5810
5820 5830
5840 5850
5860 5870
5880 5890
5900 5910
5920 5930
5940 5950
5960 5970
5980 5990
6000 6010
6020 6030
6040 6050
6060 6070
6080 6090
6100 6110
6120 6130
6140 6150
6160 6170
6180 6190
6200 6210
6220 6230
6240 6250
6260 6270
6280 6290
6300 6310
6320 6330
6340 6350
6360 6370
6380 6390
6400 6410
6420 6430
6440 6450
6460 6470
6480 6490
6500 6510
6520 6530
6540 6550
6560 6570
6580 6590
6600 6610
6620 6630
6640 6650
6660 6670
6680 6690
6700 6710
6720 6730
6740 6750
6760 6770
6780 6790
6800 6810
6820 6830
6840 6850
6860 6870
6880 6890
6900 6910
6920 6930
6940 6950
6960 6970
6980 6990
7000 7010
7020 7030
7040 7050
7060 7070
7080 7090
7100 7110
7120 7130
7140 7150
7160 7170
7180 7190
7200 7210
7220 7230
7240 7250
7260 7270
7280 7290
7300 7310
7320 7330
7340 7350
7360 7370
7380 7390
7400 7410
7420 7430
7440 7450
7460 7470
7480 7490
7500 7510
7520 7530
7540 7550
7560 7570
7580 7590
7600 7610
7620 7630
7640 7650
7660 7670
7680 7690
7700 7710
7720 7730
7740 7750
7760 7770
7780 7790
7800 7810
7820 7830
7840 7850
7860 7870
7880 7890
7900 7910
7920 7930
7940 7950
7960 7970
7980 7990
8000 8010
8020 8030
8040 8050
8060 8070
8080 8090
8100 8110
8120 8130
8140 8150
8160 8170
8180 8190
8200 8210
8220 8230
8240 8250
8260 8270
8280 8290
8300 8310
8320 8330
8340 8350
8360 8370
8380 8390
8400 8410
8420 8430
8440 8450
8460 8470
8480 8490
8500 8510
8520 8530
8540 8550
8560 8570
8580 8590
8600 8610
8620 8630
8640 8650
8660 8670
8680 8690
8700 8710
8720 8730
8740 8750
8760 8770
8780 8790
8800 8810
8820 8830
8840 8850
8860 8870
8880 8890
8900 8910
8920 8930
8940 8950
8960 8970
8980 8990
9000 9010
9020 9030
9040 9050
9060 9070
9080 9090
9100 9110
9120 9130
9140 9150
9160 9170
9180 9190
9200 9210
9220 9230
9240 9250
9260 9270
9280 9290
9300 9310
9320 9330
9340 9350
9360 9370
9380 9390
9400 9410
9420 9430
9440 9450
9460 9470
9480 9490
9500 9510
9520 9530
9540 9550
9560 9570
9580 9590
9600 9610
9620 9630
9640 9650
9660 9670
9680 9690
9700 9710
9720 9730
9740 9750
9760 9770
9780 9790
9800 9810
9820 9830
9840 9850
9860 9870
9880 9890
9900 9910
9920 9930
9940 9950
9960 9970
9980 9990

```

ZEAP Z80 Assembler - Source Listing

```

0010 ;Anpassung von SORT an DATEI (NJ 3/4-82)
0020 ;27.12.82 G.B.
0030 ;Folgende Werte sind in den Zeilen zu
0040 ;ändern:
0050 ;Z 1280 Label BEGIN eintragen
0060 ;Z 1340 #0
0070 ;Z 1350 #28 ;Dateibeginn
0080 ;Z 1360 #7
0090 ;Z 1380 #20 oder #09 (Space od. NL)
0100 ;Z 1390 #A0
0110 ;Z 1410 DATEND EQU #2006
0120 ;Z 1411 LD HL,(DATEND)
0130 ;Z 1412 LD (IX+END),L
0140 ;Z 1420 LD (IX+END+1),H ;variabel
0150 ;Z 1435 RCAL MENI

```

```

0160 ;Z 3390 HPTPRG EQU #151A ;Hauptprogramm
0170 ;Z 3391 JP HPTPRG ;EXIT
0180 ;ORG und ENT im SORT weglassen
0190 ;Die im Text erwähnten Zeilen für den
0200 ;Zähler sind im Listing 3210 bis 3290
0210 ;
0220 ORG #9000 ;(an passende Adresse legen)
0230 JR BEGIN1
0240 RST #28
0250 DEFB #C
0260 DEFM *
SORTIEREN
0270 DEFW #D0D
0280 DEFM *Nachname N
0290 DEFB #D
0300 DEFM *Postleitzahl P
0310 DEFB #D
0320 DEFM *Wohnort W
0330 DEFW #D
0340 DEFW #7BDF ;BLINK
0350 CP -N
0360 JR NZ PLZ
0370 (IX+FLDCNT),1
0380 LD (IX+DISPLC),-1
0390 LD A,*
0400 LD (#9B),A
0410 RET
0415 JR BEGIN ;Brücke
0420 CP -P
0430 JR NZ WORT
0440 (IX+FLDCNT),2
0450 LD (IX+DISPLC),0
0460 LD A,*
0470 LD (#8B),A
0480 RET
0485 JR MENI ;Brücke
0490 CP -W
0500 JR NZ KEYB
0510 LD (IX+FLDCNT),2
0520 LD (IX+DISPLC),1
0530 LD A,*
0540 LD (#9B),A
0550 RET
0570 ;PROGRAM "SORT" VERS. 3.4
1010 ;

```

# Konfigurationen

von GÜNTER KREIDL

*Ist der NASCOM tot?*

Das 80-BUS-Journal ist aus der redaktionellen Tätigkeit der Herausgeber für das NASCOM-Journal hervorgegangen. Wie wohl die meisten Leser dieser Zeitschrift besitzen sie ein NASCOM-System und fragen sich besorgt, ob dem NASCOM-1/2 noch eine Zukunft beschieden sein wird oder ob dieser Veteran unter den Einplatinencomputern bald ganz aus dem Markt verschwindet. Dies hat weniger sentimentale als praktische Gründe: Während der letzten Jahre haben wir soviel Hardware und vor allem Software für dieses System entwickelt, daß der Umstieg auf ein anderes System mit sehr viel Arbeit und Kosten verbunden wäre. Andererseits ist der NASCOM heute technisch schon überholt (Standard wäre heute vielleicht ein 64K-CP/M-System) und man bekommt inzwischen erheblich mehr Computerleistung für sehr viel weniger Geld. Der Computermarkt im untersten Preisbereich wird heute einerseits von den billigen Konsumcomputern (zwischen 150,- und 1000,- DM) und andererseits von preiswerten Platinensystemen in Industriequalität beherrscht. Dazwischen gibt es aber sowohl preislich als auch inhaltlich eine Lücke, die früher einmal Computer wie der NASCOM ausgefüllt haben (sehr viele dieser Art gab es eigentlich nicht - mir fallen nur das OHJO SUPERBOARD und der ATM-65 ein). Die "Konsum-Computer" sind eigentlich, wenn man den Preis berücksichtigt, optimal ausgestattet: Farbgraphik, Sound, relativ viel Speicher, komfortables BASIC usw. Die große Verbreitung sorgt auch für einen entsprechenden Preisdruck bei den Erweiterungen und für einige Softwareunterstützung durch Microcomputerzeitschriften. Das "Innenleben" dieser Maschinen von bleibt aber für die meisten Anwender für immer im Dunkeln, sowohl was die Software (Betriebssystem) als auch was die Hardware angeht. Bei Platinensystemen ist das in der Regel anders; hier sind sowohl Hardware als auch Software meist gut dokumentiert und auch die Erweiterung ist meist einfacher. Ein preiswertes (Ein-)Platinen-Computer ist der ideale Einstieg für jeden, der mehr als ein "Knöpfchendrücker" sein will, den Funktionsweise und Anwendung (Interfacing und Programmierung) eines Mikrocomputers wirklich verstehen lernen will. Diese Funktion eines (erweiterbaren) Einstiegscom-

puters könnte der NASCOM immer noch haben, wenn der Preis der NASCOM-Systeme sich an den heutigen Marktbedingungen orientieren würde. Zumindest in Deutschland ist das aber nicht der Fall. Die seltsame und unverständliche Preispolitik des deutschen Generalimporteurs (Michael Klein Systemtechnik, später MKV) führte zu einer Verteuerung des NASCOM, während gleichzeitig immer billigere und leistungsfähigere "Konsum-Computer" auf den Markt kamen. Der NASCOM ist zu den dort verlangten Preisen überteuert und damit in der heutigen Marktsituation unverkäuflich. Ein "nichtautorisierter" Anbieter konnte den NASCOM erheblich billiger anbieten, obwohl er ihn in England um 15% teurer einkaufte als der offizielle Importeur. Doch auch zu diesem Preis hat der NASCOM heute keine Marktchance in Deutschland mehr. Ich persönlich glaube allerdings, daß dies nicht so sein müßte. Ein NASCOM-1 für ca. 400,- DM als Bausatz bzw. ca. 500,- DM als Fertigplatine wäre m. E. immer noch ein interessanter Einstiegscomputer und evtl. auch für manche Anwendungen interessant. Ein noch größere Chance hätte aber in meinen Augen ein NASCOM-2 in folgender Konfiguration: 8K stat. RAM (4 x 6116), komplette Firmware (ZEAP, NASOPS, DEBUG, TOOLKIT, NASPEN) in Eproms und das 8k-ROM-BASIC sowie der Graphik-Zeichengenerator für einen Preis um die 1000,- DM. Das ließe sich mit minimalen Änderungen auf der NASCOM-2-Platine verwirklichen! Ich weiß nicht, ob das finanziell machbar ist, aber ohne eine solche oder ähnliche Lösung sehe ich schwarz für die Zukunft des NASCOM in Deutschland. Und was ist dann mit uns, den NASCOM-Anwendern? Sollen wir unseren NASCOM wegschmeißen?

GEMINI und RP/M

Hardwaremäßig ist Unterstützung, Ausbau und Modernisierung natürlich mit den GEMINI-Karten möglich durch die Buskompatibilität, aber GEMINI-Programme sind auf dem NASCOM genauso wenig lauffähig wie umgekehrt NASCOM-Programme auf dem GEMINI. Das Betriebssystem der GEMINI Computer heißt aus naheliegenden Gründen RP/M (Rom Program for Microcomputers). Es ist natürlich auf die GEMINI-Hardware (CPU-Karte, busparallele Videokarte und Floppycontroller) zugeschnitten, unterscheidet sich aber nicht nur dadurch von einem Betriebssystem wie NAS-SYS, sondern vor allem auch durch die ganze Systemarchitektur. NAS-SYS ist ein typischer Monitor/Debugger und vor allem zum Testen und

Entwickeln von Programmen gedacht. RP/M hingegen ist im wesentlichen ein mit CP/M kompatibles Dateiverwaltungssystem. Speicheraufteilung, Softwareschnittstelle und Behandlung von Dateien (Files) ist soweit kompatibel mit CP/M, daß z.B. der Microsoft Disk Basic Interpreter ohne eine einzige Änderung mit RP/M läuft. Der "Trick" besteht darin, daß das Cassettensystem softwaremäßig wie ein Floppy-Laufwerk angesprochen wird, wobei natürlich nur sequentielles Lesen und Schreiben möglich ist. Für die eigene Programmierarbeit bedeutet dies, daß die mit diesem System erstellten Programme (zunächst auf ein Cassettensystem zugeschnitten) beim Umsteigen auf ein CP/M-Floppy-System (was natürlich auch wieder ein GEMINI-System sein kann - RP/M enthält bereits die BOOT-Routine!) praktisch ohne Änderung lauffähig bleiben. Mit RP/M wird also eine Aufwärtskompatibilität von Cassetten- zu Floppy-Betriebssystemen erreicht. (Vergl. zu CP/M und RP/M auch den "Monitor"-Artikel in diesem Heft!) Darüber hinaus verfügt RP/M nun noch über einige Befehle zur Anzeige und Veränderung des Speichers. Pont Exp/Ausgabe-Befehle, 2 Befehle zum Starten eines Programms, den BOOT-Befehl für die GEMINI-Floppy-Systeme und einen "Screen-Edit-Mode". RP/M belegt die oberen 4K (FOOO-FFFF) im Speicher und benutzt natürlich ebenso wie CP/M die Zero-Page (0-100H) als Arbeitsspeicher. Benutzerprogramme können dann den Bereich ab 100H benutzen, ja bei 100H liegt gewöhnlich der Programmstart. Man sieht also: mit der tollen Kompatibilität zu CP/M geht andererseits die Softwarekompatibilität zum hardwaremäßig kompatiblen Vorläufer NASCOM verloren. Speicherbelegung, Systemaufrufe und Cassetten-Aufzeichnung (obwohl ebenso wie beim NASCOM-2 1200 Bd. Kansas-City!) passen nicht zusammen und das heißt: beim Übergang vom einen System zum anderen müssen alle Programme umgeschrieben werden und auch Cassettentausch ist vorläufig nicht möglich. Aber da kann man ja was dran tun!

Wie wär's mit "RP/M"

Noch'n Monitor? Nach T2 kam T4, dann NAS-SYS-1 jetzt NAS-SYS-3 (zwischen durch auch noch NAS-SYS-2) - und so ganz kompatibel waren die nie! Bitte nicht erschrecken. Wir wollen die Verwirrung nicht vergrößern, sondern nur die Kompatibilität zwischen verschiedenen Systemen herstellen. Wir (die Redaktion) möchten zusammen mit den Lesern ein Betriebssystem ent-

wickeln, das für den NASCOM eine ähnliche Kompatibilität zu CP/M herstellt, wie sie zwischen RP/M und CP/M besteht. Durch die gleiche Normung der Betriebssystemaufrufe bei CP/M und RP/M werden dann auch unter RP/M geschriebene Programme auf dem NASCOM lauffähig; entsprechend können mit dem neuen Betriebssystem Programme auf dem NASCOM entwickelt werden, die dann ohne Änderung auf GEMINI-Systemen laufen (wenn's nicht gerade um das Bildschirmformat geht) und wenn wir schon dabei sind, werden wir durch eine entsprechende Sprungleiste auch eine Kompatibilität mit dem ZAPPLE-Monitor herstellen; dies ist einer der Großväter aller Z80-Monitorprogramme, der von R. D. Klein in seinem Buch "MikroComputer, Hard- und Softwarepraxis" veröffentlicht wurde und in modifizierter Form auch im "MC-CP/M-Computer" verwendet wird. Es ist deshalb damit zu rechnen, daß in Zukunft mit diesem Monitor lauffähige Programme in der Zeitschrift MC veröffentlicht werden. Unter diesem Monitor läuft z.B. auch das 12K-BASIC von TDL (ebenfalls von R. D. Klein veröffentlicht in "Basic-Interpreter"), von denen auch der ZAPPLE-Monitor stammt. Wenn sich dann Benutzer des MC-CP/M-Computers noch ein Cassetteninterface (1200 Bd. Kansas-City) einbauen, ist auch mit ihnen ein Programmaustausch möglich. Natürlich soll auch NAS-SYS mit allen dafür geschriebenen Programmen weiter verfügbar bleiben und so weit als möglich benutzt werden. Der Monitor soll nicht im ROM stehen, sondern von der Cassette mit NAS-SYS in den oberen Speicherbereich geladen werden. Er soll modular aufgebaut sein, damit die Anpassung an verschiedene Hardwarekonfigurationen leicht möglich ist. So kann z.B. jeder Benutzer seine eigenen Treiberprogramme selbst einfügen, etwa für den Mini-Digital-Recorder, und damit andere Massenspeicher für CP/M-kompatible Dateien benutzen, oder auch die Software für die im NASCOM-Journal vorgestellte 80-Zeichen-Karte. Wir haben tatsächlich eine ganz verrückte Vorstellung; jeder NASCOM-Besitzer hat letztendlich seine eigene Monitor-Version im Speicher, aber alle sind sie kompatibel miteinander - und mit RP/M und CP/M (und ZAPPLE und ZDOS ...).

Ausblenden

Der NASCOM paßt aus 2 Gründen schlecht mit CP/M zusammen: CP/M ist Terminal-orientiert, setzt also voraus, daß sich der Bildschirmspeicher außerhalb des Adressierungsbereichs des Pro-

zessors befindet und über ein Port angesprochen wird. Üblich waren Terminals mit serieller Schnittstelle, zumeist mit eigenem Prozessor. Heute sind wohl busparallele intelligente Videokarten zum Standard geworden (sowohl beim 80-Bus als auch beim ECB-Bus), die über Parallel-Ports angesprochen werden, und eine schnellere Datenübertragung ermöglichen. Der schnellste Zugriff auf den Bildschirm ist aber in Systemen möglich, in denen der Bildschirm-Speicher ein Teil des Arbeitsspeichers ist, wie auch beim NASCOM. Dies wird jedoch von CP/M nicht unterstützt und außerdem liegen sowohl der Bildschirmspeicher als auch das Betriebssystem NAS-SYS in einem Speicherbereich, der von CP/M für andere Funktionen reserviert ist. Die CP/M-Kompatibilität wird also nicht ohne eine kleine Hardwaremodifikation erreicht, mit der wir aber plötzlich die Vorteile beider Systeme (Terminal oder Memorymapped) im NASCOM miteinander vereinen können. Die unteren 4K des Speicherraums mit NAS-SYS, dem Video-RAM und dem Workspace COOH bis FFFH müssen einer eigenen Speicher-Bank zugeordnet werden, die über ein Bit eines I/O-Ports beliebig ein- oder ausgeblendet werden kann, wobei gleichzeitig 4K-RAM einer anderen Karte (z.B. die sonst ausgeblendeten unteren 4K einer 64K-Karte) eingeblendet werden. Verwendet man z.B. Bit 5 von Port 0, dann kann man das mit 2 oder 3 IC's bewerkstelligen (im einzelnen hängt das von der verwendeten Speicherkarte ab). Man muß nur dafür sorgen, daß bei RESET NAS-SYS eingeblendet wird. Dann kann das Betriebssystem von Cassette in den oberen Speicherbereich geladen werden. Bei CP/M-Betrieb wird von diesem Betriebssystem aus dann NAS-SYS mit dem Bildschirm ausgeblendet und nur bei Bedarf eingeschaltet, wenn z.B. den Bildschirm beschrieben oder Unterprogramme von NAS-SYS verwendet werden sollen. Wer das 8K-Basic im ROM behalten möchte, kann es ebenso ausblendbar machen. Natürlich ist auch einfach Betrieb unter NAS-SYS möglich.

#### BUS-Schnittstellen

Ebenso wie bei Betriebssystemen haben sich auch bei den Bus-Systemen gewisse Standards etabliert, meist weniger durch ihre Qualität als durch die Entwicklung der Mikrocomputertechnik bedingt. Wenn man sich z.B. den in Deutschland verbreitetsten Bus für Z80-Systeme, den Kontron- bzw. ECB-Bus, auf seine Konzeption hin ansieht, kann man kaum Ordnung oder Systematik

darin erkennen. Aber diese Busbelegung mit dem Europakartenformat hat sich trotzdem durchsetzen können. Wenn erst einmal genügend Systeme und Erweiterungskarten für einen Bus auf dem Markt sind, kann sich kaum ein Anbieter leisten, ein anderes Bus-System herauszubringen, auch wenn es erheblich besser ist. Der Kunde profitiert natürlich von dem Wettbewerb der vielen Anbieter: Für den ECB-Bus werden die verschiedensten Karten als Leerkarten, Bauelemente oder Fertigplatinen angeboten zu ständig sinkenden Preisen, sowie eine Menge Zubehör - Gehäuse, Netzteile, Busplatinen und all das Kleinzug, das manchmal mehr kostet als der eigentliche Computer. Der aus England stammende 80-BUS zeigt eine ordentlichere Struktur als der ECB-Bus und bietet noch viel Platz für spätere Erweiterungen. Weniger schön ist das unhandliche Platinenformat. So haben viele NASCOM-Anwender ihr System auf den ECB-Bus umgerüstet (von allem zu der Zeit, als wegen der NASCOM-Pleite kaum noch Erweiterungen zu bekommen waren), andere hatten bereits ihr System mit dem 80-BUS ausgebaut. Für diese Anwender erwies sich die Kompatibilität der GEMINI-Karten mit dem NASCOM als nützlich. Mit den GEMINI- und MICROSYSTEM-Anwendern ist natürlich auch der Kreis der 80-BUS-Benutzer hienzulande wieder angewachsen. Wir wollen hier weder für das eine noch für das andere Bus-System Stellung beziehen, aber wir haben wieder das Problem der Kompatibilität: In welchem Kartenformat und mit welcher Busbelegung sollen die Hardware-Erweiterungen der Leser und Mitarbeiter des 80-BUS-JOURNALS entwickelt werden? Man kann es niemals allen recht machen und die Gewaltlösung (handverdrahtete Busanpassung) ist sicher auch nicht jedermanns Sache. Wir lassen deshalb derzeit eine Bus-schnittstelle entwickeln, einen ECB-Bus-Adapter für den 80-BUS. Die Adapterplatine wird in eine Steckerleiste des 80-BUS gesteckt und kann 2 ECB-Bus-Karten aufnehmen. Mit einem kleinen Trick lassen sich dann auch zwei Buskarten der beiden Systeme direkt miteinander verbinden, z.B. auch ein Mini-Mother-Board im 80-BUS-Format mit einem ECB-System, wenn jemand mit ECB-Bus 80-BUS-Karten in seinem System verwenden will. Darüber hinaus werden wir in den nächsten Ausgaben des 80-BUS-JOURNALS die Kopplung des NASCOM-Grundsystems mit ECB-Bus-Systemen beschreiben, z.B. mit dem MC-CP/M-Computer oder mit Karten von Janich&Klass. Dem NASCOM-Anwender steht damit das ganze Angebot

an ECB-Bus-Karten offen, ohne daß er auf die mit dem NASCOM entwickelte oder für den NASCOM gekaufte Software verzichten muß. Und wer nur noch mit 19-Zoll-Gehäusen arbeiten möchte, für den gibt es demnächst auch eine NASCOM-kompatible Hardware auf Europakarten. Hat jemand gesagt, der NASCOM sei tot? - Es lebe der FRCOB-BUS!

# Der neue Monitor

von GÜNTER KREIDL

## SOFTWARE-SCHNITTSTELLEN

Wie schon an anderen Stellen in diesem Heft beschrieben (siehe "Konfigurationen"), soll die Entwicklung eines neuen Monitors der Austauschbarkeit von Programmen dienen und das spätere Umsteigen auf ein Floppy-Disk-System mit CP/M erleichtern. Erst in zweiter Linie sollen zusätzliche Fähigkeiten, über die NAS-SYS nicht verfügt, eingebaut werden (ein Beispiel dafür ist der unten beschriebene Relocator). Wenn man etwas zu CP/M kompatibles entwickeln will, muß man natürlich zunächst einmal CP/M und seine Verwandten näher kennenlernen. Das ist in diesem Beitrag nur recht oberflächlich möglich, und es muß wegen der Einzelheiten auf die vielfältige Literatur verwiesen werden. Zur Zeit erscheint auch in der Zeitschrift MC eine Artikelreihe über CP/M von Rolf Dieter Klein. Wer sich heute CP/M (oder CDOS, ZDOS und wie sie alle heißen) zulegt, der erhält ein umfangreiches Programmpaket, von dem uns hier jedoch nur der Kern, das eigentliche Betriebssystem, interessiert. Das besteht aus drei Teilen, dem CCP (Console Command Prozessor), dem BDOS (Basic Disk Operating System) und dem BIOS (Basic Input Output System). Die Bezeichnungen sind eigentlich etwas irreführend; mir gefallen die (deutschen) Bezeichnungen von ZDOS besser: KI (Kommando-Interpreter), LEAS (Logisches Ein/Ausgabe-System) und HEAS (Hardwareabhängiges Ein/Ausgabe-System). Das entscheidende Kriterium ist die Hardware-Unabhängigkeit von LEAS bzw. BDOS. Nur das BIOS ist systemabhängig, muß also für jeden Computer neu geschrieben werden. Es läßt sich auch vom Benutzer verändern und ergänzen, um eine Anpassung an die jeweilige Systemkonfiguration zu ermöglichen. Das BIOS bzw. HEAS enthält also die Treiber- (Ausgabe) bzw. Monitor-Software (Eingabe) für sämtliche Peripheriegeräte: Tastatur, Video, externes

Terminal, Cassetten-Ein/Ausgabe, Lochstreifenleser und -Stanzen, Drucker usw. sowie bei einem Diskettensystem alle Ansteuernroutinen für die Diskettenlaufwerke (bei RP/M sind diese durch entsprechende Cassetten-Routinen ersetzt). HEAS bzw. BDOS besteht im wesentlichen aus zwei Funktionsblöcken: einer logischen Ein/Ausgabesteuerung und einer Dateiverwaltung für die Diskettenlaufwerke (oder andere Massenspeicher). Alle Einzelfunktionen sind über ein standardisiertes Verfahren, ein "Weichwaren-Interface", vom Benutzer oder von "CP/M-kompatiblen" Software aufzurufen.

## CALL FIVE

CP/M benutzt den Speicherbereich von 0 bis FFH als Arbeitsbereich. An der Stelle 5 steht dabei ein Sprung nach BDOS bzw. LEAS. Über diese Adresse kann man die Funktionen von LEAS als Unterprogramme aufrufen. Im Register C wird dabei eine Kennziffer für das gewünschte Unterprogramm übergeben. Eingangsparameter werden in D und E übergeben, während Ausgangsparameter in A bzw. HL übergeben werden. Diesmal sollen uns nur die grundsätzlichen Ein/Ausgabefunktionen interessieren, die mit den Kennziffern 1 bis 11 aufgerufen werden (Kennziffer 0 bewirkt einen Kaltstart des Systems). Die genaue Funktion der einzelnen Routinen kann der beigefügten Tabelle entnommen werden. Logische Eingabegeräte sind die "Konsole" und ein "Streifenleser"; logische Ausgabegeräte ebenfalls die Konsole, ein "Streifenstanzen" und ein Drucker ("List Device"). Die meisten dieser Geräte sucht man an einem NASCOM oder GEMINI vergebens. Schon die Bezeichnungen verraten, daß das Betriebssystem CP/M aus den Anfangszeit der Mikrocomputerei stammt. Aber keine Angst; man muß sich nicht so einen altmodischen "Rittermann" anschaffen, um mit CP/M arbeiten zu können. Den logischen Ein/Ausgabefunktionen können physikalische Geräte auf vielfache Weise zugeordnet werden, sofern die zugehörigen Routinen im BIOS implementiert sind.

## DAS JOBYTE

Jedem logischen Ein/Ausgabegerät können 4 physikalische Geräte zugeordnet werden. Die Zuordnung ist ebenfalls standardisiert und erfolgt über das "JOBYTE", das an der Speicherstelle 3 abgelegt ist. Je 2 Bit sind einem logischen Gerät zugeordnet in der folgenden Weise: 000000XX: Konsole  
0000XX00: Streifenleser



00XX0000: Streifenstanzen

XX000000: Drucker

"XX" bestimmt jeweils, welches Gerät bei Aufruf der entsprechenden Funktion angesteuert wird. Je nach System ist immer nur ein Teil der Gerätenummern belegt; der Rest kann vom Benutzer festgelegt werden (z.B. für einen Cassettenrekorder). Im einzelnen gelten folgende Zuordnungen:

Konsole: 00=TTY

01=CRT

10=BATCH (s.u.)

11=USER

Leser: 00=TTY

01=Streifenleser

10=USER

11=USER

Stanzen: 00=TTY            01=Streifenstanzen

10=USER

11=USER

Drucker: 00=TTY

01=CRT

10=Zeilendrucker

11=USER

Alle mit "USER" gekennzeichneten Geräte können vom Benutzer oder in manchen Fällen auch vom Konstrukteur eines Computers frei definiert werden. Oft tritt auch heute schon ein Cassettenrekorder an die Stelle des Streifenlesers bzw. -stanzen. Unter "BATCH" versteht man eine bestimmte Betriebsart, bei der der "Leser" als Eingabegerät und der "Drucker" (logisch!) als Ausgabegerät dient. Damit hat man wohl früher Lochstreifen in Klartext übersetzt. Es ist nun überhaupt nicht mehr schwierig, eine entsprechende Software-Schnittstelle für NAS-SYS zu schreiben, sofern wir vorher ein kleines Hardware-Problem lösen: Der Speicherbereich von 0 bis 4K muß zwischen RAM einerseits und EPROM/RAM auf der Grundplatte des NASCOM umschaltbar gemacht werden. Vorschläge dazu werden im nächsten 80-BUS-Journal folgen.

#### ZAPPLE'S LEISTE

Manchem Leser werden die oben beschriebenen Unterprogramme und Funktionen recht bekannt vorkommen, auch wenn er noch nie etwas mit CP/M zu tun hatte. Der von der Firma TDL in den USA vor vielen Jahren herausgebrachte "ZAPPLE Monitor" verfügt im wesentlichen über die gleichen Unterprogramme incl. der JOBYTE-Funktion. Der Monitor ist inzwischen in Deutschland mehrfach veröffentlicht worden von R. D. Klein. Eine

erste Version ist in seinem Buch "Mikrocomputer Hard- und Software-Praxis" enthalten, eine erweiterte Fassung ist neulich als Monitor für den MC-CP/M-Computer in dieser Zeitschrift erschienen. Der Monitor belegt den Speicher ab F000H und beginnt mit einer "Sprungleiste" zu den entsprechenden Ein/Ausgaberroutinen. Die Registerbelegung der verschiedenen Funktionen scheint auf den ersten Blick unterschiedlich zu sein, aber das liegt nur an der Software-Schnittstelle. Intern sind es genau die gleichen Funktionen. Ein Beispiel zeigt dies deutlicher als komplizierte Erklärungen: die Routine C0 gibt das Zeichen, das im Register C steht, auf die Konsole aus. Bei CP/M steht aber in C die Funktionsnummer, weshalb das Zeichen in E übergeben wird. Intern passiert nun aber folgendes: es wird auch bei CP/M eine Routine aufgerufen, die das Zeichen in C, das natürlich vorher von E dorthin geladen wurde, auf die Konsole ausgibt. D.h. nichts anderes, als daß die einfachsten Ein/Ausgaberroutinen bei beiden Betriebssystemen gleich sind. Nur der Zugriff erfolgt einmal über einen Unterprogrammaufruf bei 5, im anderen Fall über die Sprungleiste bei F000H. Es bereitet nun keinerlei Mühe, unseren neuen Monitor sowohl mit dieser Sprungleiste als auch mit der Einsprungstelle bei 5 zu versehen. Damit erreichen wir, daß auch Programme, die unter diesem Monitor laufen, direkt auf dem NASCOM einsatzfähig sind.

#### GARY'S RELOCATOR

Damit nicht alles Theorie bleibt, will ich mit einem kleinen Programm schließen, das wir später in den Monitor übernehmen können, das aber auch jetzt schon unter NAS-SYS benutzt werden kann. Es beruht auf einem Aufsatz von Gary A. Kildall, dem Entwickler von CP/M, in Dr. Dobbs's, Februar 1978. Er beschreibt dort ein Verfahren, Programme voll verschieblich zu machen, das dann auch in CP/M benutzt wird. Das Betriebssystem CP/M selbst kann mit diesem Verfahren den unterschiedlichsten Speicherkonfigurationen angepaßt werden. Wir können es z.B. auch benutzen, um NASCOM-Firmware an anderer Stelle als der ursprünglich vorgesehenen lauffähig zu machen. Als Beispiel wird der "Verschiebungsvektor" für NASDOS mit abgedruckt. Damit ist es möglich, den Disassembler in jedem Speicherbereich laufen zu lassen. Weitere Verschiebungsvektoren (z.B. für DEBUG, ZEAP und NASPEN) werden in den nächsten Heften folgen. (Ich wollte schon immer ZEAP nach F000H

verbannen!) RELOCATOR-Programme haben wir schon im NASCOM-Journal 6/81 veröffentlicht. Beide Programme hatten den Nachteil, daß der verschiebliche Code "von Hand" erzeugt werden mußte. Das von Gary Kildall verwendete Verfahren hat den großen Vorteil, daß dieser Code von einem Programm erzeugt wird. Voraussetzung ist, daß ein Programm in zwei Versionen vorliegt, die sich nur im Speicherbereich unterscheiden, in dem sie lauffähig sind. Dieser Unterschied darf keinen "krummen Wert" annehmen, sondern muß stets das Vielfache einer "ganzen PAGE" (=100H) betragen. Immer dann, wenn ein Programm im Assembler-Quellcode vorliegt, kann man auf einfache Weise zwei solche Programmversionen erzeugen (durch entsprechende "ORG"-Vorgaben). NASD75 habe ich hingegen erst in ein ZEAP-Format disassemblieren und editieren müssen, bevor diese Bedingung erfüllt war. Das Verfahren ist im Grunde sehr einfach. Die Routine RELVEC vergleicht die beiden Programmversionen Byte für Byte und erzeugt einen "Verschiebungsvektor", in dem für jedes Byte des Programms ein Bit vorgesehen ist. Wird ein Unterschied festgestellt, wird das entsprechende Bit gesetzt, sonst nicht. Der Unterschied muß immer gleich dem "Page-Unterschied" sein, sonst erfolgt eine Fehlermeldung. Der Verschiebungsvektor zu einem Programm kann dann separat gespeichert und bei Bedarf geladen werden. Die Routine RELOC verrechnet dann diesen Verschiebungsvektor mit einer beliebigen Version des Programms und verändert es dabei so, daß es an einer vorgeählten Adresse lauffähig wird. (Die Verschiebung muß man dann mit dem Copy-Befehl vornehmen!) Das Programm muß nicht an einer Stelle stehen, wo es auch lauffähig ist. Es muß nur eine Bedingung erfüllt sein: Der Abstand der zwei Programmversionen muß gleich dem Unterschied in "Pages" sein. Alles weitere kann man wohl dem Assemblerlisting entnehmen. (wird fortgesetzt)

#### CP/M-I/O-FUNKTIONEN

Nr.	Bedeutung
0	System rüchsetzen
1	Konsolen-Eingabe
2	Konsolen-Ausgabe
3	Leser-Eingabe
4	Stanger-Ausgabe
5	Drucker-Ausgabe
6	Dirkte Konsolen Ein/Ausgabe (s.u.)

- 7 JOBYTE holen
- 8 JOBYTE setzen
- 9 Buffer drucken bis "\$"
- 10 Buffer von Konsole füllen
- 11 A=FFH falls Konsoleingabe, sonst=0

#### Anmerkungen:

Einzelzeichen werden für die Ausgabe immer in E übergeben, Pointer (Nr. 9 und 10) in DE Eingaben erfolgen immer über A, ebenso das JOBYTE zu 6: bei Aufruf in E=FFH oder Zeichen, bei Rückkehr in A Zeichen oder 0

F000	1000	ORG	LF000
	1010	;Unterprogramme für ?P/M	
	1020	;	
	1030	;G.K. 1/83	
	1040	;Page-Relocation nach	
	1050	;einer Idee von	
	1060	;Gary A. Kildall	
	1070	;Dr. Dobb's Vol. 3, Iss. 2	
	1080	;verallgemeinert auf beliebige	
	1090	;Speicherbereiche	
	1100	;	
	1110	;RELVEC erzeugt den	
	1120	;Verschiebungsvektor	
	1130	;Registerbelegung:	
	1140	;JX: Anfangsadresse 1	
	1150	;JY: Anfangsadresse 2	
	1160	;HL: Anfang Verschiebungsvektor	
	1170	;DE: Programmlänge	
	1180	;	
F000	FDE5	1190	RELVEC PUSH JY
F002	C1	1200	POP BC
F003	78	1210	LD A,B
F004	DD55	1220	PUSH JX
F006	C1	1230	-POP BC
F007	90	1240	SUB B
F008	4F	1250	LD C,A;Page-Differenz
F009	2B	1260	DEC HL
F00A	23	1270	RELV1 JNC HL
F00B	0608	1280	LD B,8
F00D	3600	1290	LD (HL),0
F00F	CB26	1300	RELV2 SLA (HL)
F011	F5	1310	PUSH AF
F012	F07E00	1320	LD A,(JY)
F015	DD9600	1330	SUB (JX)
F018	2805	1340	JR Z,RELV3
F01A	B9	1350	CP C
F01B	201D	1360	JR NZ,RELERR
F01D	CBCE	1370	SET 1,(HL)
F01F	DD23	1380	RELV3 JNC JX
F021	FD23	1390	JNC JY
F023	1B	1400	DEC DE
F024	7B	1410	LD A,E
F025	B2	1420	OR D
F026	2809	1430	JR Z,RELV5
F028	F1	1440	POP AF
F029	10E4	1450	DJNZ RELV2
F02B	CB1E	1460	RR (HL)
F02D	18DB	1470	JR RELV1
F02F	CB26	1480	RELV4 SLA (HL)
F031	10FC	1490	RELV5 DJNZ RELV4
F033	CB1E	1500	RR (HL)
F035	F1	1510	POP AF
F036	23	1520	JNC HL
F037	37	1530	SCF
F038	3F	1540	CCF
F039	C9	1550	RET

1560 ;HL zeigt auf das höchste  
 1570 ;Byte+1 des Verschiebungsvektors  
 1580 ;Carry zurückgesetzt  
 F03A F1 1590 RELERR POP AF  
 F03B 37 1600 SCF  
 F03C C9 1610 RET  
 1620 ;JX, JY zeigen auf die Bytes,  
 1630 ;die weder gleich sind,  
 1640 ;noch die zulässige Differenz  
 1650 ;haben: Programme sind nicht  
 1660 ;gleichwertig!  
 1670 ;  
 1680 ;RELOC verrechnet den  
 1690 ;Verschiebungsvektor mit  
 1700 ;dem Programm und den  
 1710 ;gewünschten Verschiebung.  
 1720 ;Das Programm muß nicht  
 1730 ;dort stehen, wo es lauft.  
 1740 ;fähig ist.  
 1750 ;  
 1760 ;Registerbelegung:  
 1770 ;JX: Anfangsadresse Programm  
 1780 ;HL: Anfang Verschiebungsvektor  
 1790 ;DE: Länge "  
 1800 ;B: Page, auf den das Programm  
 1810 ; läuft  
 1820 ;C: Page, auf den es laufen soll  
 1830 ;  
 F03D 78 1840 RELOC LD A,B  
 F03E 91 1850 SUB C  
 F03F 4F 1860 LD C,A  
 F040 0608 1870 RELOC1 LD B,8  
 F042 CB06 1880 RELOC2 RLC (HL)  
 F044 3007 1890 JR NC,RELOC3  
 F046 DD7E0C 1900 LD A,(JX)  
 F049 91 1910 SUB C  
 F04A DD7701 1920 LD (JX),A  
 F04D DD23 1930 RELOC3 JNC JX  
 F04F 10F1 1940 DJNZ RELOC2  
 F051 23 1950 JNC HL  
 F052 1B 1960 DEC DE  
 F053 7B 1970 LD A,E  
 F054 B2 1980 OR D  
 F055 20E9 1990 JR NZ,RELOC1  
 F057 C9 2000 RET  
 2010 ;  
 2020 ;  
 2030 ;TESTPROGRAMM  
 2040 ;läuft unter NASSYS-1/3  
 2050 ;  
 2060 ;1) Vektorezeugung  
 2070 ;Auflauf mit E Startadresse  
 2080 ;+ 4 Argumente:  
 2090 ;Programmadresse 1 + 2  
 2100 ;Programmlänge + Adresse  
 2110 ;des Verschiebungsvektors  
 2120 ;  
 F058 0C0C 2130 ARG EQU ECOC  
 F058 3A0B0C 2140 LD A,(ARG-1)  
 F05B FE05 2150 CP 5  
 F05D 201F 2160 JR NZ,ERRMSG  
 F05F DD2A0E0C 2170 LD JX,(ARG+2)  
 F063 FD2A100C 2180 LD JY,(ARG+4)  
 F067 ED5B120C 2190 LD DE,(ARG+6)  
 F06B 2A140C 2200 LD HL,(ARG+8)  
 F06E CD00F0 2210 CALL RELVEC  
 F071 3806 2220 JR C,ERR1  
 F073 DF66 2230 SCAL E66  
 F075 DF6A 2240 SCAL E6A  
 F077 DF5B 2250 SCAL E5B  
 F079 DDE5 2260 ERR1 PUSH JX  
 F07B E1 2270 POP HL  
 F07C DF66 2280 SCAL E66  
 F07E DF6B 2290 ERRMSG SCAL E6B  
 F080 DF5E 2300 SCAL E5B  
 2310 ;  
 2320 ;2) Verschieben  
 2330 ;Auflauf mit E Startadresse  
 2340 ;+ 5 Argumenten:

2350 ;Anfangsadresse Programm  
 2360 ;Anfangsadresse + Länge des  
 2370 ;Verschiebungsvektors  
 2380 ;Page 1 = lauffähige Adresse  
 2390 ;Page 2 = Zieladresse der  
 2400 ;Relokation (jeweils 1 Byte!)  
 2410 ;Das Programm wird nur umgerechnet  
 2420 ;nicht verschoben - Verschiebung  
 2430 ;mit "J"- oder "C"-Befehl

2440 ;  
 F082 3A0B0C 2450 LD A,(ARG-1)  
 F085 FE06 2460 CP 6  
 F087 20F5 2470 JR NZ,ERRMSG  
 F089 DD2A0E0C 2480 LD JX,(ARG+2)  
 F08D 2A100C 2490 LD HL,(ARG+4)  
 F090 ED5B120C 2500 LD DE,(ARG+6)  
 F094 3A140C 2510 LD A,(ARG+8)  
 F097 47 2520 LD B,A  
 F098 3A160C 2530 LD A,(ARG+10)  
 F09B 4F 2540 LD C,A  
 F09C CD30F0 2550 CALL RELOC  
 F09F DF5B 2560 SCAL E5B

#### NASDJS VERSCHIEBUNGSVEKTOR

F400 00 20 04 00 00 00 00 00 18  
 F408 00 00 02 48 04 00 00 00 4A  
 F410 04 00 00 00 80 40 00 00 C8  
 F418 00 00 80 00 00 00 00 00 8C  
 F420 00 00 00 08 40 24 04 24 A8  
 F428 00 22 48 80 09 24 24 00 57  
 F430 20 20 00 48 04 12 00 04 C6  
 F438 80 20 08 10 10 01 01 24 1A  
 F440 90 00 48 90 84 21 22 08 68  
 F448 00 08 02 02 48 00 40 90 60  
 F450 24 09 24 82 00 00 20 00 37  
 F458 01 08 00 89 02 00 48 80 A8  
 F460 00 00 00 00 12 40 09 20 CF  
 F468 90 02 48 24 09 21 00 00 84  
 F470 48 12 49 04 02 40 49 09 9F  
 F478 02 00 90 10 09 00 12 49 72  
 F480 22 40 00 40 40 12 00 78  
 F488 08 20 20 04 12 40 81 20 B8  
 F490 04 40 90 09 02 48 90 00 38  
 F498 22 10 84 02 00 00 24 89 F1  
 F4A0 10 01 20 88 10 00 40 92 2F  
 F4A8 49 20 04 10 02 40 80 00 08  
 F4B0 AA AA AA AA AA AA AA AA F4  
 F4B8 00 00 00 00 00 00 00 00 AC  
 F4C0 00 00 00 00 00 00 00 00 B4  
 F4C8 00 00 00 00 00 00 00 00 BC  
 F4D0 00 00 00 00 00 00 00 00 C4  
 F4D8 80 00 00 00 00 00 04 00 50  
 F4E0 00 00 00 00 00 00 00 10 E4  
 F4E8 00 00 00 00 00 00 00 00 0C  
 F4F0 00 00 00 00 00 00 00 00 E4  
 F4F8 00 00 00 40 00 00 02 00 2E  
 F500 00 00 00 00 00 00 00 00 F5  
 F508 00 00 00 09 00 00 00 00 06  
 F510 00 02 40 08 00 01 00 00 50  
 F518 00 00 02 00 00 40 00 00 4F  
 F520 00 00 00 00 00 00 00 00 15  
 F528 00 82 40 00 00 42 00 10 31  
 F530 40 00 00 00 00 00 00 20 85  
 F538 00 00 00 02 00 00 00 00 2F  
 F540 00 00 00 00 20 00 00 00 55  
 F548 20 00 00 00 20 00 00 00 7D  
 F550 00 00 00 00 00 00 80 00 C5  
 F558 00 00 00 00 00 01 10 5E  
 F560 00 00 00 00 00 00 42 97  
 F568 00 00 00 00 04 02 00 20 83  
 F570 00 10 00 00 01 00 20 10 A6  
 F578 00 01 00 00 00 00 01 01 6F

# Gemini Microcomputer

Vertriebs - GmbH

GEMINI bietet Ihnen die Unterstützung, die Sie brauchen, wenn Sie Ihr NASCOM - System für ernsthafte Anwendungen einsetzen wollen.

Wir liefern ein breites Spektrum hochwertiger Platinen für den 80 - Bus, zum Beispiel:

PLUTO Farbgrafik mit 192 KBytes RAM, eigenem 8088 Prozessor, 640x288 Bildpunkte in 8 Farben	DM 2471,90
I/O - Erweiterung mit 3 PIOs, 1 CTC, 1 RTC	DM 918,90
Floppydisk - Controller für 5 1/4 und 8", double side, double density	DM 880,--
IVC Video Platine mit eigener CPU, 80x24, 128 ASCII- Zeichen, 128 Zeichen frei programmierbar	DM 880,--
CPU/RAM Platine mit Z80 A CPU, 64 KBytes RAM, PIO, UART und Cass. Interface	DM 1523,30
CMOS - RAM, 32 KBytes mit Akku für ca. 1 Jahr	DM 880,--
CP/M 2.2 Betriebssystem für NASCOM	DM 685,--
POLY-DOS Betriebssystem, läuft unter NAS-SYS, mit Basic- Erw., Editor und Assembler	DM 620,--

## Sonderangebote gültig bis 28.2.1983

Floppydisk Erweiterung für NASCOM, komplett mit 2 Laufwerken im Gehäuse, mit Netzteil, Kabel, Controller und nach Wahl POLY-DOS oder CP/M 2.2	DM 3885,--
BASIC MICROSYSTEM mit CPU/RAM, IVC, ASCII-Tastatur, RP/M - Betriebssystem und 12 KBytes BASIC Interpreter	DM 1995,--
NASCOM 2 mit 64 KBytes RAM, aufg. u. getest.	DM 2100,--
Matrix - Drucker C. ITOH 8510 A, mit 3K - Buffer und par. Schnittstelle und Software für NAS-SYS	DM 1695,--
PRINCE Monitor, grün 24 MHz	DM 395,--

Alle Preise verstehen sich einschließlich Mehrwertsteuer, zuzügl. Porto und Verpackung.  
Zwischenverkauf vorbehalten.

Schluderstr. 10 • 8000 München 19

Tel. 089/168595

