

# 80-bus journal

Zeitschrift für NASCOM, GEMINI und andere  
Z80-Anwender

1. JAHRGANG \* NOVEMBER 1983 \* AUSGABE 10/11



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

# 80-bus journal

Intern

Liebe Leser,  
mit Mühe habe ich diese Doppelnummer nun fertiggestellt, damit Sie sie noch termingerecht vor Weihnachten erhalten. Heute Nacht wurde es 6.30, und am heutigen Sonntag ist es nun auch schon wieder 20h, gerade noch Zeit, das Editorial fertig zu machen und das Layout auf das Sonderpostamt zu bringen, damit es bis morgen bei der Druckerei ist.

Sie sehen, es ist keine Nachlässigkeit, daß nun noch ein Heft für 83 aussteht; es ist einfach nicht zu schaffen. Vielleicht bemüht sich der Günter Kreidl, wenigstens für die Ausgabe 12 noch etwas zu tun, damit ich die

letzte Ausgabe für dieses Jahr nicht auch noch alleine durchziehen muß,

Was das Journal für's nächste Jahr angeht, so habe ich eine Menge Post mit Zusagen und aufmunternden Worten erhalten, sogar einige großzügige Spenden sind schon eingegangen. Leider wurde die notwendige Mindestanzahl der Abos noch nicht erreicht. Bedauerlich ist auch, daß Leser aus dem Bekanntenkreis und sogar engere Mitarbeiter es versäumten, positiven Bescheid zu geben. Darauf angesprochen, wird immer reine Vergeßlichkeit genannt. Da ich annehmen kann, daß nicht nur die näher Bekannten so nachlässig sind, sollten wir den Zusagemodus ändern: Wenn Sie das Abonnement für 1984 nicht haben wollen, so kündigen Sie bitte bis Ende des Jahres 83. Vielleicht hätte man die Aktion gleich so beginnen sollen, denn im Impressum der Probehefte stand ja, daß bei versäumter Kündigung das Abo automatisch verlängert wird.

Nach augenblicklichem Stand der Dinge ist die Chance recht groß, daß das Journal weitermacht. Damit Sie die Zurückhaltung richtig verstehen, will ich Ihnen eine kleine Rechnung aufmachen:

die Druckkosten für 4 Doppelhefte DM 8000,-

100 Abonnements

DM 6000,-

Damit wäre auch noch lange kein Porto, geschweige denn Umschläge und die vielen Kosten für die Erstellung des Layouts gedeckt. Auch die monatlichen Mitteilungen sind hierbei noch lange nicht berücksichtigt. Bisher habe ich erst 50 Zusagen.

Ich warte also noch bis Jahresende auf Kündigungen und werde Ihnen dann im Januar mitteilen, wie sich die Sache entwickelt hat.

Zur Bestellung der Floppy- Karte und zur Neuauflage der 80x24-Zeichen- Karte beachten Sie bitte die Information bei den Leserbriefen.

Auch auf die CLD-DOS Liste möchte ich hinweisen (Übrigens ein weiteres Argument, die NASCOM- Gemeinde am Leben zu erhalten).

Nun bleibt mir nichts weiter, als Ihnen viel Spaß mit dieser Ausgabe zu wünschen und ein frohes Fest.

Vielleicht gibt es einen guten Rutsch in ein neues "Journal- Jahr".

Ihr Günter Böhm

Ich stelle mir z.Z. ein CP/M-System aus ECB-Modulen (Elektronikladen) zusammen. Als Laufwerke habe ich 8" vorgesehen, weil 8" single sided/single density das einzige zuverlässig genormte Format ist. Wenn's läuft, soll ein Programm 'rein, das sich verhält wie NAS/SYS. Im Prinzip kein Problem bis auf die Bildschirm-Ansteuerung. Denkbar wäre es, den Bildschirm doppelt zu führen damit auch Direktzugriffe noch funktionieren. ZEAP und BLS-Pascal würde ich gern weiterbenutzen, denn es gibt unter CP/M nichts ähnlich Komfortables für kleinere Programme. Hat jemand schon eine CP/M-Nas/Sys-Anpassung gemacht? Notfalls mach ich's halt selber (und veröffentliche es an dieser Stelle).

Michael Bach, Gundelfingen

# Inhalt

2	80-Bus Journal Intern	
3	Inhalt/Impressum	
4	Leserbriefe	
7	Spooler	Manfred Stoll
9	Seite(n) für Floppy-Einsteiger	Günter Böhm
14	EMDOS Floppyverwaltung	Helmut Emmelmann
16	Floppy Hilfsprogramme	Günter Böhm Helmut Emmelmann
19	EMDOS Anpassung	Helmut Emmelmann
21	Video- Verbesserung	Tom D. Rudebusch
22	Hochauflösende Grafik	Jörg Wittich
24	Hetzjagd	Peter Brendel
	EPROM Port II	Horst Dieckhoff
26	Kalender	Henrik Maier
27	Kalenderberechnung	H.u.R. Schulte
29	PASCAL- Drucker	Jürgen Weiermann
30	PASCAL- Spiel	Michael Bach
31	CLD-DOS Programmliste	
34	PASCAL- Funktionen	Jürgen Weiermann
35	OVERLAY	Gerhard Klement
	Spannungsüberwachung	K.Trust/G.Assmann
38	CLOCK	Gerhard Klement
39	CONMOD II	Jürgen Weiermann
40	nascomp1	
	2 BASIC Spiele	Eberhard Horch
42	Roulette	Klaus Mombaur
43	Mitarbeiter	
44	Hull-FORTH Review	Constantin Olbrich
45	Buchführung	Klaus Mombaur
48	RAM/EPROM Karte	Karl Schulmeister
52	Sonderangebote	

# IMPRESSUM

HERAUSGEBER:  
 Günter Böhm Ludwigshafener Str. 21d  
 75 Karlsruhe Tel. [REDACTED]  
 Redaktion  
 ebendort  
 Gabi Böhm Layout u. Versand  
 Pertenweg 18  
 Günter Kreidl Tel. [REDACTED]  
 4172 Straelen Buchhaltung

KORRESPONDENTEN:

Karl Georg Englmann [REDACTED]  
 Mütterstadt Tel. [REDACTED]  
 Reinzeichnungen

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

Clemens u. Max Ballarin [REDACTED]  
 Ueberlingen Tel. [REDACTED]

Michael Bach [REDACTED]  
 Stegen Tel. [REDACTED]

Peter Brendel [REDACTED]  
 Mannheim Tel. [REDACTED]

Hans-Jürgen Plath [REDACTED]  
 Kiel [REDACTED]

Hans Schneider [REDACTED]  
 Esens [REDACTED]

Oesterreich:  
 Gerhard Klement [REDACTED]  
 A-[REDACTED] Wien Tel. [REDACTED]

Niederlande:  
 Eric v.d.Vaart [REDACTED]  
 NL-[REDACTED] Waddixveen [REDACTED]

England:  
 Frank M. Butler [REDACTED]  
 Mansfield Woodhouse/Notts [REDACTED]

Luxemburg:  
 Rene Claus [REDACTED]

L-[REDACTED] Bonneweg [REDACTED]  
 Schweiz:  
 Markus Zimmer [REDACTED]  
 CH-[REDACTED] Basel Tel. [REDACTED]

Jugoslawien:  
 Gilvazi Istvan [REDACTED]  
 YU-[REDACTED] Becej [REDACTED]

VERLAG:  
 Günter Kreidl 4172 Straelen  
 VERTRIEBSWEISE 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 erschienenen Hefte nachgeliefert. Die Lieferung von Einzelheften durch den Verlag ist nicht möglich. Bitte zahlen Sie direkt bei der Bestellung auf das Postcheckkonto:  
 Günter Kreidl [REDACTED] PSchA Essen

HAFTUNG und RECHTE:  
 Für Fehler in Texten, Bildern, Programmen und Schaltungen und daraus entstehende Schäden 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 Auswertung ist nur mit Genehmigung des Verfassers 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 werde.

# Leserbriefe

Lieber Herr Böhm,  
das unwahrscheinlich lange Ausbleiben des Journals ließ schon darauf schließen, daß bei der Redaktion etwas nicht stimmt. Nach dem Erscheinen der September- Ausgabe wissen wir nun, daß der anscheinend um diese Jahreszeit unvermeidliche Kampf um das Fortbestehen der NASCOM- Zeitschrift entbrannt ist. Wir sind darüber etwas verwundert, denn die Redakteure waren ja keine Neulinge mehr, als sie sich zur Herausgabe des 80-Bus Journals entschlossen hatten, und wußten, was an Arbeit auf sie zukommt.

Wir hatten auch nicht den Eindruck, daß es der Redaktion an Beiträgen fehlt, da diese in dieser Hinsicht Optimismus verbreitete und ein von uns eingesandtes Programm nicht abgedruckt wurde. Wir hatten dieses allerdings an Herrn Kreidl gesandt, da dieser im Impressum als zuständiger Redakteur angegeben war. Uns hat es auch befremdet, daß wir nicht einmal die eingesandten Cassetten zurückerhalten haben. Auf einen längerfristigen Fortbestand des Journals haben wir keine große Hoffnung, denn die bisherigen NASCOM- Besitzer werden sich mit der Zeit etwas Moderneres zulegen; wir denken da an CP/M. Zu Beginn des Jahres hat ja Herr Kreidl noch große Hoffnungen auf einen CP/M-Monitor für den NASCOM verbreitet, aber dabei ist es dann auch geblieben. Da man aber von der Hoffnung allein nicht leben kann, haben wir uns einen Bausatz des mc CP/M Computers bestellt. Wir werden aber dem NASCOM trotzdem noch treu bleiben, da man ihn ohnehin nur noch verschenken könnte.

Wir werden auch dem 80-Bus Journal treu bleiben und nochmal DM 60 investieren.

Mit freundlichen Grüßen

Max u. Clemens Ballarin, 7770 Überlingen

Lieber Herr Ballarin und Clemens,  
der Kampf um's Überleben, an den man ja inzwischen wirklich schon gewöhnt ist, hat tatsächlich begonnen, und ich freue mich sehr, daß Sie beide mich weiterhin unterstützen.

In der Tat wußten wir, was auf uns zukam, aber ich wußte nicht, was auf mich zukommt; denn, wie schon in Heft 9 mitgeteilt, wurde

aus der Redaktion ein Ein-Mann-Betrieb durch die berufliche Überlastung von Günter Kreidl.

Letzteres ist auch der Grund, warum Sie und verschiedene andere Leser auf Cassetten und Antworten warten, Günter Kreidl schickt das schon alles zurück, aber eben sehr langsam und sporadisch. Ich warte auch schon seit geraumer Zeit auf Artikel, die an ihn geschickt wurden, von Ihrem Beitrag wußte ich gar nichts. Das erledigt sich alles, ich habe inzwischen auch Geduld lernen müssen.

Was den längerfristigen Fortbestand des Journals angeht, haben Sie recht: Über Jahre hinaus wird sich ein reines NASCOM- Heft nicht halten. Wie Sie aus der Entwicklung der Beiträge erkennen (ich weise nur auf die Schulmeister- Platinen hin) ist hier eine breite Tendenz zu ECB- Systemen zu erkennen (viele Leser bauen die Schaltungen nach) und damit hin zu CP/M. Der Erfolg des neuen NASCOM C zeigt, daß auch die Liebhaber des großzügigen 80-Bus-Formats nun ihr CP/M System erwerben. Damit schließt sich der Kreis, und im Laufe der Zeit wird es eben ein CP/M Journal geben.

Daß die Entwicklung relativ langsam geht, macht mich nicht traurig. Auch ich habe mir, wie Sie vielleicht wissen, die mc- CPU-Karte aufgebaut. Aber anstatt mir nun ein komplettes System zuzulegen, bin ich lieber etwas an der Entwicklung beteiligt; dann weiß ich auch, was im Rechner eigentlich abläuft. Ich kann mir vorstellen, daß es vielen Lesern ähnlich geht, sonst hätten sie sich schon längst eine Banane gekauft, oder wie das Ding heißt, und ihren Lötkeulen eingemottet.

Der alte NASCOM wird von mir in Ehren gehalten, denn er hat mir den Einstieg in die Computerei ermöglicht, und er tut in der Adreßverwaltung und einigen täglich anfallenden Arbeiten gute Dienste. Das wahre "Abenteuer" wartet aber mit der Weiterentwicklung des ECB- Rechners auf mich, und ich freue mich nun - nach dem Floppy Frust - schon auf die nächsten Probleme mit der Grafik-Karte. Man sagt "schön, wenn der Schmerz nachläßt"; und genauso sagen viele 80-Busler "schön, wenn es endlich läuft". Die nächsten Geburtswehen kommen bestimmt. Leiden Sie mit?

Mitfreundlichen Grüßen

Günter Böhm

Sehr geehrter Herr Böhm,  
bedanke mich sehr für die gute Gestaltung  
des 80-Bus Journals.

Damit Sie besser kalkulieren können, möchte  
ich hiermit mitteilen, daß ich das Journal  
auch weiterhin bestellen werde. Das Geld für  
die Floppy- Karte habe ich bereits überwie-  
sen. Die CRT- Karte habe ich mittlerweile  
von Günter Kreidl erhalten (Sehen Sie, Herr  
Ballarin G.B.). In Weinheim kaufte ich mir  
auf dem Flohmarkt eine ECB- Bus Karte und  
möchte in nächster Zeit den gesamten NAS auf  
ECB umstellen. Aus diesem Grund wollte ich  
alle Karten von Schulmeister beziehen. Wo  
kann man sie bestellen? Mir ist es bisher  
nicht geglückt, solche Karten selbst zu  
ätzen.

Ich würde mich freuen, wenn das 80-Bus  
Journal noch recht lange erscheinen würde,  
denn es stellt eine große Wissensbereiche-  
rung dar.

Mit freundlichen Grüßen  
Emil Dorsch , 6305 Buseck

Lieber Herr Dorsch,  
Sie sind nicht der einzige, der nach den  
Schulmeister-Karten fragt. Leider kamen die  
Anfragen so spät und zeitlich auseinanderge-  
zogen, sonst hätte sich die Herstellung  
einer durchkontaktierten Serie gelohnt. In-  
zwischen haben nun Herr Assmann und Herr  
Trust eine kleine "Produktion" aufgenommen  
und stellen die Karten her. Sie sind leider  
ungebohrt, aber in recht guter Qualität  
geätzt.

Sie können die Karten bei angegebener Adres-  
se bestellen. Allerdings sollten alle Be-  
stellungen bis Ende des Jahres eingehen,  
denn die Herren haben langsam keine Lust  
mehr, die eintröpfelnden Bestellungen ein-  
zeln zu bearbeiten.

Eine Karte erhalten Sie für DM 15,- inklu-  
sive Porto und Verpackung. (Ein guter Preis  
für die Menge Arbeit und Material).

Es sind folgende Karten erhältlich  
CPU- Karte

Video 48x16

Cassetteninterface und Tastaturanschl.

256K RAM

RAM/EPR0M Karte

PIO/SIO Karte

Zu bestellen sind die Karten (ungebohrt)  
bei

Georg Assmann oder Karl Trust

Herzhausen

Rosenthal

#### NASCOM TREFFEN

Die bisherigen lokalen Treffen der NASCOMer  
und anliegender Rechner- Benutzer waren bis-  
her immer ein Erfolg. Und wenn man nur ein  
Programm auf Cassette nach Hause nahm, so  
hatte der Tag immer noch etwas gebracht.  
Gedankenaustausch hat noch nie geschadet.

Nun hat Wolfgang Mayer- Gürr einmal die  
Initiative ergriffen und möchte ein Treffen  
in Recklinghausen an seiner Schule organi-  
sieren.

Wer in der Gegend wohnt (oder auch etwas  
außerhalb) und sich für ein Treffen am 5.  
Februar 1984 interessiert sollte sich melden  
bei

Wolfgang Mayer-Gürr

Recklinghausen

Tel. [Redacted]

Vielleicht wird dort auch so etwas allgemein  
Nützliches ausgekocht wie unser Floppy-  
Verwaltungsprogramm damals in Mannheim.

#### TIPS, TRICKS und KAFER

Zum Vorschlag in Heft 9/83 , die NMI Leitun-  
gen der Floppy-Karte und der Single-Step-  
Mimik über ein Gatter zusammenzuführen,  
hatte Georg Assmann eine elegantere Lösung  
gefunden. Auf der I/O- Karte von Karl Schul-  
meister ersetzte er die entsprechenden Gat-  
ter durch Open-Collector Typen (LS03) und  
legte jeweils einen Pull-up Widerstand an  
die Ausgänge. So kann NMI (und WAIT) von  
mehreren Karten bedient werden.

Auf der Floppy- Karte ist diese Möglichkeit  
schon gegeben, und der NASCOM II scheint sie  
auch schon vorgesehen zu haben. Beim NASCOM  
I müßte man an der Unterbrechung (7/8 Seite  
6) und ebenfalls auf der LIST/Niemann- Karte  
entsprechende Gatter einfügen.

Im BASIC Listing von Black Jack (5/83)  
tauchten seltsame Zeichen auf, die einigen  
Lesern Schwierigkeiten bereiteten. Hier des  
Rätsels Lösung:

Exp 2 = Division /

Pfund = Gleich =

B.Hasko Günther, Ansbach, stellte einige Unstimmigkeiten beim Verschiebungsvektor des ZEAP (2/83) fest. Hier hat er uns die Korrekturen mitgeteilt:

Im Datenfeld sind diverse Adressen übersehen worden. In D01BH befindet sich der Zeiger auf dieses Datenfeld, welches die Adressen der Editor-Befehle enthält. Dieses Feld geht von D036H bis D075H. Dieser und die nachfolgenden Zeiger werden beim Kaltstart ins RAM (F00H) geladen und müssen ebenfalls geändert werden. Folgendes sind die geänderten Bytes im Verschiebungsvektor, mit dem ZEAP jetzt auf jedem Bereich einwandfrei läuft:

F600 24 08 00 0A A0 00 00 92  
 F608 49 24 92 49 24 92 48 00

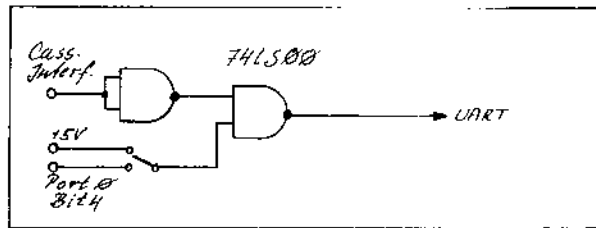
"Langsame" EPROMs 2732 oder 2764 lassen sich auch bei 4 MHz Taktfrequenz ohne WAIT-Zyklen betreiben, wenn der Chip-Select nicht mit MREQ verknüpft wird. Dafür müssen RD und MREQ über ein ODER-Gatter auf den OE-Eingang des EPROMs gelegt werden. Auf diese Weise läuft bei mir sogar ein 2732-6 bei 4 MHz problemlos. Eine andere einfachere Lösung wäre, das ursprüngliche CS-Signal direkt auf den OE-Eingang zu geben und den CS-Eingang ständig auf LOW zu lassen. Dies ist jedoch mit einer höheren Leistungsaufnahme des EPROMs verbunden.

Das automatische Laden und Starten eines Programms mit dem Generate-Befehl in NASSYS empfinde ich als recht praktisch. Ich habe daher fast alle meine Programme auf diese Weise abgespeichert. Welche Möglichkeit hat man nun aber, wenn man einmal den Execute-Befehl verhindern möchte?

Die einfachste Möglichkeit ist folgende: Bildschirm löschen und in die oberste Zeile "R" schreiben. Danach in die Mitte der Zeile(n), in der (denen) man den E-Befehl erwartet, z.B. ein "X" schreiben. Dann mit dem Cursor zurück in die oberste Zeile und mit New Line den R-Befehl ausführen. Nun kann man das Tonband starten.

Der ganze Trick dabei ist, daß der Execute-Befehl durch das zusätzliche fehlerhafte Zeichen mit ERROR abgebrochen wird. Diese Methode funktioniert natürlich nur bei Programmen, die kürzer als 13 KByte sind. Durch folgende kleine Schaltung kann man aber auch hardwaremäßig da Starten des Programms ver-

hindern:



Jörg Wittich, 771 Donaueschingen

PLATINENSERVICE

Die Nachfrage nach den Floppy-Karten war so groß, daß die erste Serie sofort vergriffen war. So müssen einige Besteller nun eben bis zur Woche vor Weihnachten warten, das ist der Termin, den uns die Lieferfirma versprochen hat.

Leider mußten wir die Lieferfirma wechseln, da der ursprüngliche Platinenhersteller sehr unzuverlässig in der Einhaltung der Termine war.

Nachdem uns dadurch neue Kosten für die Herstellung eines Bohrbandes etc. entstanden, und die Kosten für PROMs und Programmierung sowieso zu knapp kalkuliert waren, haben wir bei dieser Aktion bisher bedauerlicherweise 'draufbezahlt. Deshalb müssen wir den Preis für Neubestellungen auf DM 70,- erhöhen. (Die Erstbesteller lachen sich natürlich in's Fäustchen).

Wenn Sie in die augenblickliche Serie noch einsteigen wollen, so bestellen Sie bitte sofort. Es ist nicht gewiß, ob nochmals eine Serie aufgelegt wird. Momentan könnten wir die Stückzahl aber sicher noch erhöhen.

Für die 80x24-Karte haben sich inzwischen auch nochmals einige Interessenten gemeldet. Eine Mini-Serie ist jetzt in der Herstellung. (Termin wie oben). Je weniger Abnehmer, desto mehr Kosten pro Platine. Deshalb müssen wir für diese Platine DM 65,- verlangen. Ich bitte die Vorbesteller, den Betrag auf unten angegebenes Konto zu überweisen. Auch hier bestünde die Möglichkeit für Neubesteller, noch mit in die Serie einzusteigen. Dann aber bitte sofort melden. Die "Aufträge" werden prinzipiell nach Eingang der Bestellungen bearbeitet.

Grundsätzlich möchte ich darauf hinweisen,

daß an der ganzen Sache nichts zu verdienen ist. Interessant wäre so etwas, wenn man in große Serien gehen könnte. Seien Sie also versichert, daß Preiskalkulationen von den entstehenden Unkosten diktiert werden.

Inzwischen habe ich ein gesondertes Post-scheckkonto eingerichtet (so etwas verursacht zum Glück keine Unkosten). Ihre Überweisungen betreff Platinen- Service (und möglicherweise in Zukunft auch 80-Bus Journal) bitte nur noch auf folgendes Konto

Gabi Böhm  
 Hard- und Software  
 PSchA Kirh

**KLEINANZEIGE**

CLD-DOS Floppy- Disk für 1380,- DM (Neupreis 2000,- DM) vier Monate alt zu verkaufen. Mit Gehäuse, Netzteil, alle Anschlußschnüre, sofort an den 80-Bus anzuschließen und lauffähig; umfangreiche Software, CLD-DOS Betriebssystem, E-BASIC, Assembler, Debugger, Editor sowie 12 Disketten sind im Preis inbegriffen. Franz- Josef Bilden, Tel.

**KLEINANZEIGE**

Verkaufe: LO 15 mit Lochstreifen VB ; 300,- (Selbstabholer natürlich weniger) Suche: Wer kann mir helfen, meinem NASCOM1 T-Forth beizubringen? Rolf Lohmer, Tel. (nach 17. 30 h)

**KLEINANZEIGE**

Verkaufe BASF 6106 Floppy- Leiche Interessant für Tüftler, die es durch Austausch einiger (oder aller?) ICs wieder zum Laufen bringen oder als Ersatzteillager. Schaltpläne werden mit beerdigt. Mit 50,- sind Sie dabei (Porto knöpfe Ich Ihnen aber ab) Günter Böhm, Tel.

# Spooler

In vielen Zeitungen liest man etwas zum Thema "SPOOLER" oder "DRUCKERSPEICHER".

(Ich habe diese Artikel gelesen, ohne mich von diesem Thema angesprochen zu fühlen. Dieses galt jedoch nur so lange, bis ich zu einem Matrixdrucker mit parallelem Interface kam. Die Drucker-geschwindigkeit beträgt 50 Char./sec.

Aus diesem Grunde mußte ich mich mit dem Wandeln serieneller in parallele Daten beschäftigen. Normalerweise kann man das mit einem Wandler sehr einfach lösen, aber was macht man, wenn während des Druckens das Papier alle ist? Außerdem ist man beim Drucken langer Listen auch recht lange untätig. Gerade beim Programmieren in Assembler ist es hin- und wieder erforderlich, einmal ein kurzes Listing vom derzeitigen Programmteil zu erstellen und man fragt sich dann immer wieder: welches ist nun der aktuelle Stand. Aus diesem Grund kann man z.B. jeder Ausgabe an den Drucker eine laufende Nummer mitgeben. In meiner Spoolerlösung habe ich mich zugunsten einer Uhr entschlossen. Den Impuls für die Uhr lege ich dabei als 1/12 Takt an einen Eingang der P10. Eine Steuerung über Interrupt ist in diesem Falle nicht erforderlich, da man die Statussignale (UART, SUART oder P10) sowieso dauernd auf Datenbewegung prüfen muß (RDY, TXEMPTY).

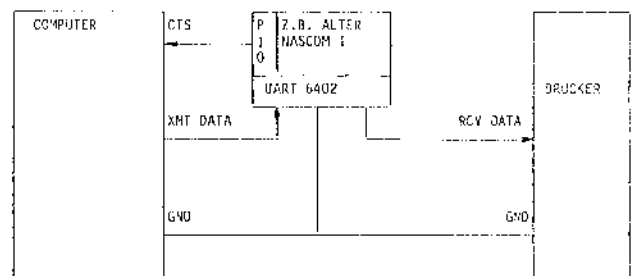
Wenn Daten zum Printen anstehen, so gibt der Spooler erst die aktuelle Uhrzeit an den Drucker aus, danach erfolgt ein Carriage Return und Linefeed. Erst dann werden die Daten abgedruckt. Bei Power "ON" wird die Uhr auf Null gesetzt. Wird am System ein "RESET" durchgeführt, bleibt der alte Stand der Uhrzeit erhalten. Eine Steuerung durch Modewort ermöglicht das Stellen der Uhr vom angeschlossenen Rechner aus. Es gibt folgende Modeworte:  
 1. Laden der Uhrzeit : Escape,1,1234 (Eingabe ohne Komma)  
 2. Initialisieren der Uhr: Escape,1 (Eingabe ohne Komma)  
 3. Printen der Uhrzeit : Escape,P (Eingabe ohne Komma)  
 Ausdruck der Uhrzeit: TIME 12:34:00

Vom DOS BASIC kann man die Uhr z.B. so stellen:  
 100 INPUT "UHR STELLEN Z.B. 1944 ";A\$  
 110 OPEN "AT:" FOR WRITE AS FILE #1  
 120 PRINT #1,CHR\$(27);"L";A\$  
 130 CLOSE #1

Ich finde es nicht gut, wenn in Zeitschriften seitenweise Programm-listings abgedruckt werden (mein Assemblerlisting hat 9 Seiten), deshalb möchte ich es bei dieser Systembeschreibung belassen.

Wer an diesem Thema interessiert ist, kann sich gerne mit mir in Verbindung setzen.

Manfred Stoll  
 Hamburg  
 Tel. Privat:  
 Gesch.:



Wie auf diesem Bild erkennbar, ist auch ein "alter", nicht mehr verwendeter NASCOM I, auch ohne große Änderungen, als "SPOOLER" zu verwenden. Es muß lediglich die Möglichkeit geschaffen werden, eine unterschiedliche XMT und RCV Speed zu verwenden.

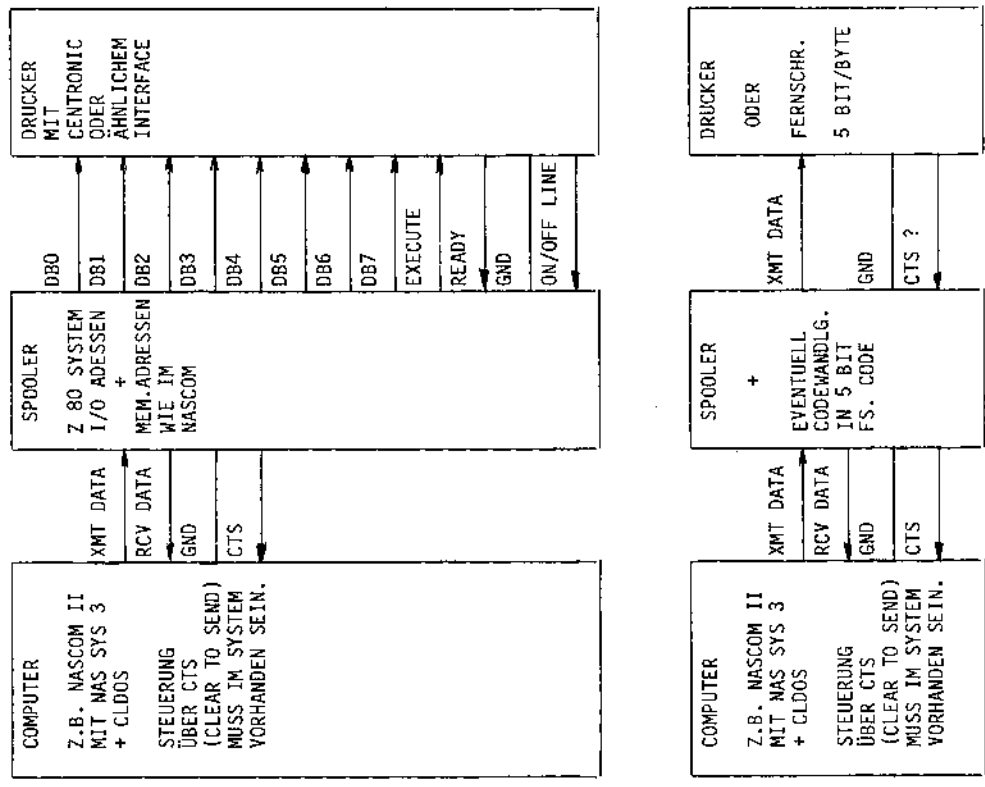
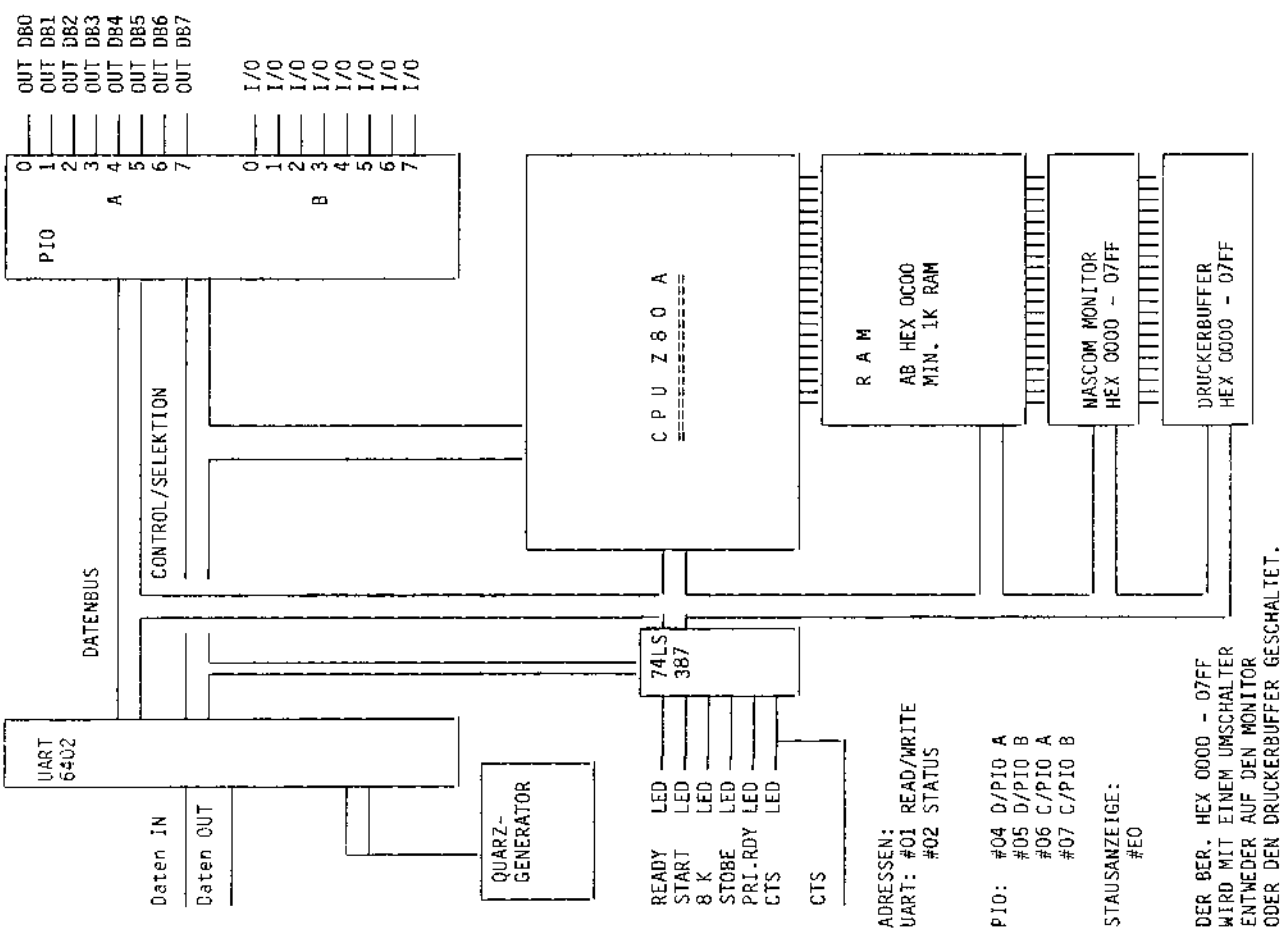
Zum Realisieren dieser oder ähnlicher Probleme biete ich gerne meine Mithilfe an.

- Folgende Fragen sollten geklärt sein:
1. Welcher I/O Baustein wird am Computer verwendet?
  2. Eingangs-chip am SPOOLER.
  3. Ausgabe über UART, PIO, SIO oder ähnliches.
  4. Kanaladressen der I/O Bausteine.
  5. Kanaladressen für CTS und Anzeigelampen (Status).
  6. Betrieb mit oder ohne Uhr.

Nach meiner Ansicht soll ein Spooler auch folgende "Intelligenz" des Druckers ausgleichen können. Folgende Anforderungen wurde ich an einen Spooler stellen (je nach eingesetztem Drucker):

1. Wandeln von Klein- in Großschreibung (wenn der Drucker nur "Groß" kann).
2. Wandeln der Interfacebedingungen (800 ODD Parity 2 Stopbit in z.B. 7DB ohne Parity).
3. Geschwindigkeitsadsption.
4. Wandeln von Seriell- in Paralleldaten.
5. Wandeln der Daten von ASCII in Baudot.
6. Ausgabe der Uhrzeit oder einer Laufnummer.
7. Ausführen eines Formfeed, wenn der Drucker es von sich aus nicht kann.
8. Drucken von 3,5 und 4 auch, wenn der Drucker diese Charakter nicht im Zeichenvorrat hat (z.B. a Backspace ).
9. Signalgabe bei "ERROR" oder Papierende.

\*\*\*\*\*  
 \* D R U C K E R B U F F E R M. STOLL SEPTEMBER 1983 \*  
 \*\*\*\*\*





# Seite(n) für Floppy-Einsteiger

von GÜNTER BÖHM

In der letzten Ausgabe habe ich Ihnen berichtet, welche Schwierigkeiten es gab, bis die Controller-Karte am Laufen war. Dabei handelte es sich um den Prototypen von Dieter Oberle. Als ich nun daran ging, meine eigene Karte aufzubauen, die aus unserer ersten Kleinserie stammt und ja auch schon bei einigen von Ihnen gelandet ist, war die Frustration sehr groß; denn die Karte lief zunächst nicht, und so mußte die bereits beschriebene Schweißarbeit von neuem beginnen.

*nichtiges "W" NASCOM PC*

Es wurden einige Unstimmigkeiten zwischen Schaltplan, Layout und Prototypen festgestellt, die leicht zu bereinigen waren, aber der hartnäckigste Fehler wurde erst gefunden, als der Dieter Metzler eigens aus Freiburg hier herauf fuhr, um mit einem sehr guten Oszi die minimalen Störungen an den Spitzen des 16 MHz - Taktes erkennbar zu machen. Das Einlöten eines winzigen Kondensators löste dann alle Probleme.

Inzwischen laufen zwei Karten aus der Serie, und somit dürften für alle Nachbauer keine Probleme mehr auftreten.

Der Übersichtlichkeit wegen möchte ich die korrigierte Stückliste hier nochmals abdrucken.

R6 - R13 150  $\Omega$   
R16, R22, R23 10K  
R3, R4, R5, R17, R24, R25, R26, R27, R28,  
R29, R30, R31 1k  
R18, R19, R20, R21 100  $\Omega$   
R14, R15 220  $\Omega$   
R1, R2 220K  
C1 47uF  
C2 4,7 uF  
C3 1 n  
C4 10 n  
C5, C6, C7, C8 100p  
C9 120p  
CB 4x100 n

R6 - R13 kann ein SIL-Widerstandsarray sein. Es können aber ebenso einzelne Widerstände stehend eingelötet werden. Oben verbinden und im untersten Lötauge gegen +5V verlöten.

Die 4x1K-Pullups bei Prom1 sollten Einzelwiderstände sein, deren gemeinsame Verbindung am 3. Lötauge von oben an +5V geht. Die Verwendung eines SIL-Arrays setzt hier eine Änderung der Leiterbahnen voraus.

Der unbezifferte Widerstand auf dem Bestückungsplan ist R19.

R30 auf Platinenrückseite zwischen Pin2 und Pin14 des 74LS38 einlöten.

R31 zwischen Pin1 und Pin16 von Prom2 auf der Rückseite einlöten.

C9 auf der Rückseite zwischen Pin3 des 74LS04 und Masse einlöten (ein 74S04 könnte von Vorteil sein).

Am 7438 müssen Pin12 und Pin13 verbunden werden.

Die Brücken an J2 müssen bei Verwendung eines 6106 Laufwerks offen sein. (Sie spiegeln ein READY vor, auch wenn das Laufwerk noch nicht bereit ist).

Vorsicht bei der Bestückung von R2 und R3. Sie dürfen nicht zu eng gebogen sein, sonst erwischen Sie das falsche Lötauge. (R2 muß an Kondensator und Brücke, R3 unterhalb Pin4 des 74LS123.

Bei Nichtverwendung der DMA brauchen I16 und I17 nicht bestückt zu werden. Die entsprechenden Kondensatoren CB können dann gleichfalls entfallen.

Es sind zwar noch einige Unterschiede zum Schaltplan auf der Karte, aber die haben keinerlei Einfluß auf die Funktion. So liegt z.B. das ENABLE des 74LS245 auf Masse, und einige Ein- und Ausgänge von Gattern wurden zur besseren Leitungsführung vertauscht. Dies erschwert zwar die Fehlersuche etwas, aber Fehler kommen ja bei uns nun nicht mehr vor, oder?

Nachdem also die Hardware soweit funktionierte (ein zeitlicher Rückfall von 14 Tagen und viel Nervensubstanz) konnte ich endlich an die Anpassung unseres Disketten-Verwaltungsprogramms (EMDOS) gehen. Dies hat mich zwar auch einem Magengeschwür nähergebracht, aber schließlich steht ein recht komfortables System zur Verfügung, das sich nicht mit den Cassettenverwaltungen vergleichen läßt, was Komfort und vor allem Geschwindigkeit angeht. Schon jetzt, wo alles noch mit Single Density abläuft, muß ich sagen, daß sich der Aufwand gelohnt hat.

An anderer Stelle ist von Helmut Emmelmann einiges über die Anpassung des Programms an einen Controller geschrieben. Das dürfte

aber für Floppy-Neulinge etwas zu hoch sein (ich konnte die Lücken auch nur durch einige Telefongespräche schließen). Deshalb möchte ich hier eine detaillierte Beschreibung anhand des Programmablaufs geben, wie ich es in der letzten Ausgabe mit den Testroutinen gemacht habe. Den Artikel haben allerdings einige Leser nicht aufmerksam verfolgt, sonst hätte sich nicht das Gerücht halten können, der Controller liefe nicht mit CP/M, weil auf den Speicher unterhalb 100H durch den NMI zugegriffen würde. Deshalb nochmals: der Bereich bei 66H wird vor den Floppy-Zugriffen gerettet und danach wieder restauriert. Im Prinzip wird der genannte Speicherbereich nicht angegriffen.

Nun aber zur Programmbeschreibung.

Das EMDOS wird bei A100H gestartet. Nun macht es zunächst eine Initialisierung der Controllerkarte und versetzt sich selbst in den Warmstart-Modus, d.h. bei erneuten Starts mit EA100 wird die Initialisierung nicht mehr durchgeführt. Beim Auftreten irgendwelcher Fehler könnte es aber von Vorteil sein, das EMDOS erneut kaltzustarten. Zu diesem Zweck können Sie die Speicherzelle A187H mit B7 laden, und ein Kaltstart wird wieder möglich. Wenn wir das DOS in Zukunft mit einem geeigneten Umlader von der Diskette "booten", wird diese Manipulation nicht mehr nötig sein. (Mehr dazu im nächsten Heft).

Im folgenden nochmals die möglichen EMDOS-Anweisungen mit gekürzten Erklärungen. Ausführlichere Beschreibungen können Sie in Heft 7/8 Seite 41 nachlesen.

#### EMDOS Anweisungen

```
L Filename [neue Anfangsadresse]
S Filename Anfang Ende [Start 0 nur laden
                        1 laden und RET.
                        2 starten
                        3 Call von EMDOS]
S Filename (abspeichern mit gleichen Param.)
B Filename (abspeichern des BASIC Programms)
D           (Inhaltsverzeichnis)
E Filename (löschen)
R Filename Anfang Länge
W Filename Anfang Länge
M           (NASSYS oder BASIC)
Filename   (laden und starten)
```

Durch die Anpassung haben sich noch zusätzliche Fehlermeldungen ergeben. Deshalb drucken wir hier nochmals den gesamten Katalog ab.

#### Fehlermeldungen des DOS:

Das DOS meldet die Fehler in Form einer Fehlernummer:

- 10 - Keine Diskette eingelegt
- 11 - Disketten Lesefehler
- 15 - Diskette ist schreibgeschützt
- 16 - Disketten Schreibfehler
- 17 - Falsche Laufwerksnummer
- 1F - Kein Laufwerk angeschlossen
- 20 - File nicht gefunden
- 21 - Diskette voll
- 22 - Fileende erreicht
- 23 - Falscher Filename
- 40 - Falscher Filetyp bei L (mit W abgespeichert)
- 41 - Falsche Anweisung
- 42 - Falscher Parameter bei Load
- 43 - Falscher Parameter bei Save
- 44 - Falscher Parameter bei Read/Write

Die folgenden Ausführungen laufen parallel zum Assemblerlisting. Ein Vergleich dürfte ihnen helfen, die Funktionen zu durchschauen.

EMDOS ruft PHEAS (die Schreib/Leseroutinen) auf und enthält in A die Funktion, 00 bedeutet dabei "Initiiieren". Den Programmteil INIT kennen Sie schon aus der letzten Ausgabe.

01 sollte das Laufwerk abstellen, wird aber nicht benutzt, da unsere Controllerkarte das hardwaremäßig selbst besorgt, wenn nicht mehr auf die Karte zugegriffen wird.

02 selektiert das Laufwerk. Im Register C ist die Laufwerksnummer enthalten. Nach dieser Nummer wird die entsprechende Routine angesprochen. Hier wird der P10A das Steuerwort übergeben (siehe Tabelle Journal 9/83 Seite 8) und HL auf die Tabelle des Laufwerks gerichtet (DSB). In dieser Tabelle findet EMDOS die Anzahl der Sektoren, die für das Betriebssystem reserviert sind (bei meinem augenblicklichen Format 3 Spuren mit je 10 Sektoren =30), die Anzahl der Sektoren, die auf Spur 3 für das Inhaltsverzeichnis vorgesehen sind (hier wurden 8 Sektoren angenommen) und die Anzahl der "Gruppen". Das sind jeweils 4 Sektoren (also zusammen 1KByte). Da wir mit unserem Format im Augenblick 92K speichern können, entspricht das 92 Gruppen.

Entsprechend PINIT und PSEL werden durch 03 und 04 im Akku die Lese- bzw. Schreibroutinen aufgerufen.

Die Routine TSTHOM testet, woher eine Fehlermeldung kommt. (gesetztes Carry-Flag weist immer auf einen Fehler hin. Deshalb auch immer das SCF/Set Carry Flag bei den

verschiedenen Testroutinen). Es wird zunächst ein RESTORE- Befehl an den Controller gegeben, der den Kopf auf Spur 0 setzen soll. Der Aufruf der Routine HOME wird deshalb nicht gewählt, weil diese eine Endlosschleife enthält, aus der das Programm nicht mehr herausfindet, wenn kein Interrupt erfolgt, d.h. wenn der Controller den Befehl nicht ausführen kann. Stattdessen wird eine Verzögerungsroutine aufgerufen, die dem Laufwerk Zeit gibt, den Motor anzuwerfen. Läuft die Diskette, kann der Controller an Bit1 das Vorhandensein des Indeximpulses feststellen. Fehlt dieser, so ist keine Diskette vorhanden. Bit2 zeigt an, daß der Kopf Spur 0 erreicht hat. Hat er das nicht, so ist wohl kein Laufwerk angeschlossen (oder die Zigarette des Programmierers ist im Laufwerk eingeklemmt). In diesen Fällen erfolgt eine Fehlermeldung nach Rückkehr zu EMDOS.

PREAD ruft wie PWRITE zunächst die Suchroutine PSEEK auf, die als erstes oben beschriebenen Test durchführt, falls das Laufwerk nicht bereit ist. Fällt dieser Test negativ aus, werden die PUSHs von PREAD/PWRITE rückgängig gemacht und ebenso die Returnadresse zu diesen Routinen "weggeworfen". Ein Return kehrt dann direkt zu EMDOS zurück.

Bei positivem Testverlauf wird Bit7 von HL untersucht. Dieses Bit ist rückgesetzt, wenn EMDOS absolute Sektornummern erwartet (es zählt die Sektoren auf der Diskette von Anfang bis Ende durch ohne Rücksicht auf die Spuren, auf denen sie sich befinden), d.h. der von EMDOS erwartete Sektor9 entspricht auch dem physikalischen Sektor9 auf der Diskette. Ist das Bit7 gesetzt, so erwartet EMDOS die Sektornummern ab Spur 3 (die ersten 3 Spuren sind ja für das System reserviert). In unserem Fall muß also zur relativen Sektornummer die Anzahl addiert werden, die sich auf den ersten 3 Spuren befindet, und das sind 10. So ergibt die relative Sektornummer 9 die physikalische 39.

Nun wird die Sektornummer durch die Anzahl der Sektoren pro Spur dividiert. (Beispiel:  $39/10=3$  Rest 9). Dabei entspricht das Ergebnis der Spur (die werden ja ab 0 gezählt) und der Rest dem Sektor. Da (bei Rest 0) der Sektor 0 nicht existiert, wird jeweils noch 1 addiert.

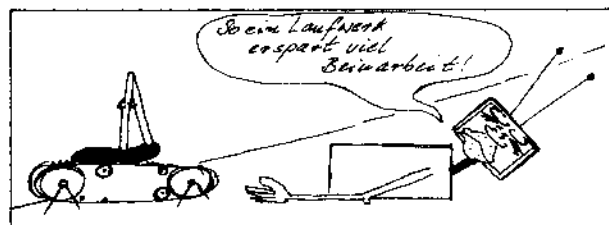
In Seek wird nun die entsprechende Spur

aufgesucht, in Setsek die Sektornummer gespeichert. Dann geht's zurück zur aufrufenden Routine.

Da man nie sicher sein kann, ob ein Sektor auf Anhieb richtig gelesen werden kann, sollen mehrere Versuche gemacht werden. 10 "Retries" sollen üblich sein, und so werden zunächst einmal 5 Versuche gemacht und dann ein "HOME", denn die Spur könnte sich ja leicht verstellt haben. Sind weitere 5 Versuche erfolglos, wird mit der Fehlermeldung 11H, gesetztem Carry und hängendem Kopf zu EMDOS zurückgekehrt. (Verstehen Sie das mit dem Kopf nicht falsch; es ist nicht der Schreib/Lesekopf gemeint).

Die PWRITE Routine macht es etwas einfacher: sie versucht einfach 10 mal zu schreiben und macht dann ihre Fehlermeldung (Schreibschutz oder was anderes).

Man könnte die Tests sicher noch viel aufwendiger machen, indem man statt HOME versucht, einen Schritt des Motors nach innen und außen vorzunehmen etc. Hierzu fehlt mir aber der Vergleich mit professionellen Programmen. Bis jetzt geht es gut, und ich hoffe nicht, Disketten lesen zu müssen, die von total verstellten Laufwerken beschrieben wurden.



Die eigentlichen Controllerroutinen sind weitgehend gleich geblieben. Hier und da mußten einige PUSHs und POPs eingebaut werden, die Bildschirmanzeige beim Interrupt ist entfernt. Wenn Sie die Routinen im letzten Heft vergleichen, werden Sie die kleinen Änderungen erkennen. An der grundlegenden Funktion hat sich nichts geändert. Wenn Ihre Controllerkarte getestet ist, brauchen Sie nur noch das EMDOS einzutippen, die Anpassungsroutine zu assemblieren (oder auch im Maschinencode einzugeben) und dann können Sie Ihre ersten Speicherbereiche zum Test auf Diskette schreiben. (Hier kann man sich die tollsten Filenamen ausdenken. Die wurden bei mir z.T. recht unflätig, als die Anpassung noch nicht so recht funktionieren wollte).

Auf daß Ihre Filenamen positiv sind!

ZEAP 280 Assembler - Source Listing

```

0010 ;-----
0020 ;PHEAS-ANPASSUNG VER. 1.8 / 26.11.83
0030 ;GÜENTER BOEHM KARLSRUHE
0040 ;nach Vorgaben v. H.Emmelmann 10/83
0050 ;-----
A969          0060      ORG #A969
A969 B7      0070 PHEAS OR A
A96A CA2DAB 0080      JP Z PINIT ;KARTE INITIIEREN
A96D 3D      0090      DEC A
A96E C8      0100      RET Z ;NICHT BENUTZT
A96F 3D      0110      DEC A
A970 280C    0120      JR Z, PSEL ;LAUFWERK SELEKTIEREN
A972 3D      0130      DEC A
A973 CAB6A9 0140      JP Z PREAD ;SEKTOR LESEN
A976 3D      0150      DEC A
A977 CA19AA 0160      JP Z PWRITE;SEKTOR SCHREIBEN
A97A 37      0170      SCF
A97B C9      0180      RET
0190 ;-----
A97C 08AB    0200 INTVEK DEFV FLPINT ;INTERRUPT TABELLE
0210 ;-----
A97E 79      0220 PSEL LD A,C
A97F B7      0230      OR A
A980 2808    0240      JR Z PSELA
A982 FE01    0250      CP 1
A984 280C    0260      JR Z PSELB ;USW. BIS LAUFWERK Z
A986 3E17    0270      LD A,#17
A988 37      0280      SCF
A989 C9      0290      RET
0300 ;
A98A 3E31    0310 PSELA LD A,#31 ;LAUFW. A SD
A98C D310    0320      OUT (10H),A
A98E 219AA9 0330      LD HL,DSBA
A991 C9      0340      RET
0350 ;
0360 ;DSBB EQU DSBA;EIN ANDERES FORMAT
0370 ;WUERDE EINE EIGENE DSB TABELLE
0380 ;ERFORDERN
0390 ;
A992 3E31    0400 PSELB LD A,#31 ;LD A,#32 LAUFW. B SD
0410 ;HIER NOCH NICHT ANGESCHLOSSEN
A994 D310    0420      OUT (10H),A
A996 219AA9 0430      LD HL,DSBA ;DSBB FUER EIGENES FORMA
A999 C9      0440      RET
0450 ;
A99A 1E      0460 DSBA DEFV 30;3 X SEKTORANZAHL (res.System)
A99B 08      0470 DEFV 8 ;8 Sekt. DIRECTORY
A99C 5B      0480 DEFV 91 ;max.Gruppenanzahl-1
0490 ;
0500 ;
A99D 3E03    0510 TSTHOM LD A,PHOME ;WARUM IST LAUFWERK
A99F F3      0520      DI ; NICHT BEREIT?
A9A0 D30C    0530      OUT (FDCCMD),A
A9A2 CDDFA9 0540      CALL CPDEL
A9A5 DB0C    0550      IN A,(FDCSTA)
A9A7 CB57    0560      BIT 2,A
A9A9 2004    0570      JR NZ NODISK
A9AB 3E1F    0580      LD A,#1F ;KEIN LAUFWERK
A9AD 1805    0590      JR TSTERR
A9AF CB4F    0600 NODISK BIT 1,A
A9B1 C8      0610      RET Z
A9B2 3E10    0620      LD A,#10 ;KEINE DISKETTE
A9B4 37      0630 TSTERR SCF
A9B5 C9      0640      RET
0650 ;
0660 ;-----
A9B6 C5      0670 PREAD PUSH BC
A9B7 E5      0680      PUSH HL
A9B8 CDECA9 0690      CALL PSEEK
A9BB D5      0700      PUSH DE
A9BC E1      0710      POP HL ;BUFFER
A9BD 010205 0720      LD BC,#0502 ;2 X 5 RETRIES
A9C0 CD90AA 0730 RRETRY CALL READ
A9C3 B7      0740      OR A
A9C4 2813    0750      JR Z OKREAD
A9C6 10F8    0760      DJNZ RRETRY
A9C8 0D      0770      DEC C
A9C9 280E    0780      JR Z OKREAD
A9CB CD77AA 0790      CALL HOME
A9CE E1      0800      POP HL ;SEKTORNUMMER
A9CF E5      0810      PUSH HL
A9D0 CDECA9 0820      CALL PSEEK
A9D3 D5      0830      PUSH DE
A9D4 E1      0840      POP HL ;BUFFER
A9D5 0605    0850      LD B,5 ;NOCH 5 RETRIES
A9D7 18E7    0860      JR RRETRY
A9D9 E1      0870 OKREAD POP HL
A9DA C1      0880      POP BC
A9DB B7      0890      OR A
A9DC 200A    0900      JR NZ RDERR
A9DE C9      0910      RET
0920 ;-----
A9DF F5      0930 CTDEL PUSH AF
A9E0 C5      0940      PUSH BC
A9E1 DB0C    0950      IN A,(#C) ;FDC ANSPRECHEN=MOTON
A9E3 DF5D    0960      DEFV #5DDF ;TDEL NASSYS 1 SEC DELAY
A9E5 C1      0970      POP BC
A9E6 F1      0980      POP AF
A9E7 C9      0990      RET
1000 ;-----
A9E8 3E11    1010 RDERR LD A,11H;Lesefehler
A9EA 37      1020      SCF
A9EB C9      1030      RET
1040 ;-----
A9EC DB0C    1050 PSEEK IN A,(FDCSTA)
A9EE CB7F    1060      BIT 7,A ;READY?
A9F0 C49DA9 1070      CALL NZ TSTHOM
A9F3 3004    1080      JR NC PSE050
A9F5 E1      1090      POP HL ;RETURN ADDRESS
A9F6 E1      1100      POP HL ; "
A9F7 E1      1110      POP HL ; "
A9F8 C9      1120      RET
A9F9 C5      1130 PSE050 PUSH BC
A9FA CB7C    1140      BIT 7,H
A9FC 2804    1150      JR Z PSEL00 ;phys.=log. Sektornummer
A9FE 011E80 1160      LD BC,30-8000H ;Bit 7 rücksetzen
AA01 09      1170      ADD HL,BC ;phys.=relat.+Anzahl Syst.Sekt
AA02 D5      1180 PSEL00 PUSH DE
AA03 110A00 1190      LD DE,10 ;Sekt.pro Spur
AA06 3EFF    1200      LD A,0FFH
AA08 B7      1210      OR A ;Reset Carry
AA09 45      1220 PSE200 LD B,L ;Rest
AA0A ED52    1230      SBC HL,DE ;DIVISION
AA0C 3C      1240      INC A ;Ergebnis=Spur
AA0D 30FA    1250      JR NC PSE200
AA0F D1      1260      POP DE
AA10 CD81AA 1270      CALL SEEK
AA13 78      1280      LD A,B
AA14 3C      1290      INC A ;Sekt.0-9=Sekt.1-10
AA15 C1      1300      POP BC
AA16 C3DAA   1310      JP SETSEK
1320 ;-----
AA19 C5      1330 PWRITE PUSH BC
AA1A E5      1340      PUSH HL
AA1B CDECA9 1350      CALL PSEEK
AA1E EB      1360      EX DE,HL
AA1F 060A    1370      LD B,10
AA21 CDB0AA 1380 WRETRY CALL WRITE
AA24 B7      1390      OR A
AA25 2802    1400      JR Z OKWRIT
AA27 10F8    1410      DJNZ WRETRY
AA29 E1      1420 OKWRIT POP HL
AA2A C1      1430      POP BC
AA2B B7      1440      OR A
AA2C 2001    1450      JR NZ WTERR
AA2E C9      1460      RET
1470 ;
AA2F CB77    1480 WTERR BIT 6,A ;WRITE PROTECT?
AA31 3E15    1490      LD A,#15
AA33 2002    1500      JR NZ WTERR1
AA35 3E16    1510      LD A,#16
AA37 37      1520 WTERR1 SCF
AA38 C9      1530      RET
1540 ;-----
1550 ;FDC CONTR. VER 1.7 19.11.83
1560 ;G.Bohm/H.Emmelmann
1570 ;nach Routinen von Oberle/Zippel
AA39 0C0E    1580 ARG2 EQU 0C0EH
AA39 000C    1590 FBCCMD EQU 000CH
AA39 000D    1600 FDCTRK EQU 000DH

```

AA39 000E	1610	FDCSEK EQU 0FH	AA9C E1	2420	POP HL
AA39 000F	1620	FDCDAT EQU 0FH	AA9D 0E0F	2430	LD C, FDCDAT
AA39 000C	1630	FDCSTA EQU 0CH	AA9F 3E8C	2440	LD A, FREAD
	1640	;	AAA1 FB	2450	EI
AA39 0010	1650	FLOAD EQU 10H	AAA2 E5	2460	PUSH HL
AA39 0011	1660	PIOAC EQU 11H	AAA3 D30C	2470	OUT (FDCCMD), A
AA39 0012	1670	PIOBD EQU 12H	AAA5 18FE	2480	READW JR READW
AA39 0013	1680	PIOBC EQU 13H	AAA7 DB0C	2490	IN A, (FDCSTA)
	1690	;	AAA9 CDDCAA	2500	CALL RE66
AA39 0003	1700	HOME EQU 03 ;SPEED STEPPER MOTOR	AAAC E1	2510	POP HL
AA39 001B	1710	FSEEK EQU 1BH;SPUR SUCHEN	AAAD D1	2520	POP DE
AA39 001F	1720	FSEEKV EQU 1FH;SUCHEN MIT VERIFY	AAAE C1	2530	POP BC
AA39 008C	1730	FREAD EQU 8CH;SEKTOR LESEN	AAAF C9	2540	RET
AA39 00AC	1740	FWRITE EQU 0ACH;SEKTOR SCHREIBEN		2550	;
AA39 00D0	1750	FINT0 EQU 0D0H ;FDC RÜCKSETZEN	AAB0 C5	2560	WRITE PUSH BC
AA39 00F4	1760	FWRTRK EQU 0F4H;GANZE SPUR SCHREIBEN	AAB1 D5	2570	PUSH DE
AA39 00E4	1770	FRDTRK EQU 0E4H;GANZE SPUR LESEN	AAB2 E5	2580	PUSH HL
	1780	;	AAB3 CDD0AA	2590	CALL SAV66
	1790	;	AAB6 21F0AA	2600	LD HL, NMW
AA39 F3	1800	INIT DI ;PIOs u. FDC Initialisieren	AAAB CDDFAA	2610	CALL RF66HL
AA3A 3ECF	1810	LD A, 0CFH	AABC E1	2620	POP HL
AA3C D311	1820	OUT (PIOAC), A ;CONTROLMODE	AABD 0E0F	2630	LD C, FDCDAT
AA3E 3E00	1830	LD A, 0C0H	AABF 3EAC	2640	LD A, FWRITE
AA40 D311	1840	OUT (PIOAC), A ;I/O Maske	AAAC1 FB	2650	EI
	1850	;	AAC2 E5	2660	PUSH HL
AA42 3ECF	1860	LD A, 0CFH	AAC3 D30C	2670	OUT (FDCCMD), A
AA44 D313	1870	OUT (PIOBC), A ;CONTROLMODE	AAC5 18FE	2680	WRITEW JR WRITEW
AA46 3E00	1880	LD A, 0C0H	AAC7 DB0C	2690	IN A, (FDCSTA)
AA48 D313	1890	OUT (PIOBC), A ;I/O Maske	AAC9 CDDCAA	2700	CALL RE66
	1900	;	AACC E1	2710	POP HL
AA4A 3EB7	1910	LD A, 0B7H ;INTERRUPT CONTROL MODE	AACD D1	2720	POP DE
AA4C D311	1920	OUT (PIOAC), A	AACE C1	2730	POP BC
AA4E 3E7F	1930	LD A, 7FH	AACF C9	2740	RET
AA50 D311	1940	OUT (PIOAC), A ;MASKE:BIT7 macht Int		2750	;
	1950	;	AAD0 216600	2760	SAV66 LD HL, 66H ;RAMBEREICH 66H RETTEN
AA52 3E08	1960	LD A, 8 ;FDC RÜCKSETZEN	AAD3 11E8AA	2770	LD DE, SAVE
AA54 D310	1970	OUT (FLOAD), A	AAD6 010400	2780	LD BC, 4
AA56 0E01	1980	LD C, 1	AAD9 EDB0	2790	LDIR
AA58 CDF4AA	1990	CALL DELAY	AADB C9	2800	RET
AA5B 3E28	2000	LD A, 28H		2810	;
AA5D D310	2010	OUT (FLOAD), A	AADC 21E8AA	2820	RE66 LD HL, SAVE ;UND WIEDER HERSTELLEN
AA5F 3E00	2020	LD A, FINT0	AADF 116600	2830	RE66HL LD DE, 66H
AA61 D30C	2030	OUT (FDCCMD), A	AAE2 010400	2840	LD BC, 4
AA63 E3	2040	EX (SP), HL	AAE5 EDB0	2850	LDIR
AA64 E3	2050	EX (SP), HL	AAE7 C9	2860	RET
AA65 DB0C	2060	IN A, (FDCSTA)		2870	;
AA67 3EA9	2070	LD A, #A9 ;INTERR.VECTOR MSB	AAE8 0004	2880	SAVE DEFS 4
AA69 ED47	2080	LD I, A		2890	;
AA6B 3E7C	2090	LD A, 7CH ;LSB	AAEC EDA2	2900	NMIR INI
AA6D D311	2100	OUT (PIOAC), A	AAEE ED45	2910	RETN
AA6F 3E31	2110	LD A, 31H ;LAUFWERKNUMMER HIER A	AAF0 EDA3	2920	NMW OUTI
AA71 D310	2120	OUT (FLOAD), A	AAF2 ED45	2930	RETN
AA73 FB	2130	EI		2940	;
AA74 ED5E	2140	IM 2 ;INTERRUPT MODE	AAF4 E5	2950	DELAY PUSH HL
AA76 C9	2150	RET	AAF5 D5	2960	PUSH DE
	2160	;	AAF6 C5	2970	PUSH BC
AA77 3E03	2170	HOME LD A, HOME	AAF7 0664	2980	DELAY1 LD B, 100
AA79 D30C	2180	OUT (FDCCMD), A	AAF9 17	2990	DELAY2 RLA
AA7B FB	2190	EI	AAFA 29	3000	ADD HL, HL
AA7C 18FE	2200	HOMEW JR HOMEW	AAFB 29	3010	ADD HL, HL
AA7E DB0C	2210	IN A, (FDCSTA)	AAFC 05	3020	DEC B
AA80 C9	2220	RET	AAFD C2F9AA	3030	JF NZ, DELAY2
	2230	;	AB00 0D	3040	DEC C
	2240	;	AB01 C2F7AA	3050	JF NZ, DELAY1
AA81 D30F	2250	SEEK OUT (FDCDAT), A ;TRACK NUMBER	AB04 C1	3060	POP BC
AA83 3E1B	2260	LD A, FSEEK	AB05 D1	3070	POP DE
AA85 FB	2270	EI	AB06 E1	3080	POP HL
AA86 D30C	2280	OUT (FDCCMD), A	AB07 C9	3090	RET
AA88 18FE	2290	SEEKW JR SEEKW		3100	;
AA8A DB0C	2300	IN A, (FDCSTA)	AB08 F5	3130	FLPINT PUSH AF
AA8C C9	2310	RET	AB09 C5	3140	PUSH BC
	2320	;	AB0A E5	3150	PUSH HL
AA8D D30E	2330	SETSEK OUT (FDCSEK), A ;SEKTORNUMMER	AB0B DB0C	3160	IN A, (FDCSTA)
AA8F C9	2340	RET	AB0D 322CAB	3170	LD (FLPSTA), A
	2350	;	AB10 210600	3180	LD HL, 6
AA90 C5	2360	READ PUSH BC	AB13 39	3190	ADD HL, SP
AA91 D5	2370	PUSH DE	AB14 4E	3200	LD C, (HL)
AA92 E5	2380	PUSH HL	AB15 23	3210	JNC HL
AA93 CDD0AA	2390	CALL SAV66			
AA96 21ECAA	2400	LD HL, NMW			
AA99 CDDFAA	2410	CALL RE66HL			

```

AB16 46      322Q      LD   B,(HL)
AB17 0A      323Q      LD   A,(BC)
AB18 FE18    324Q      CP   18H
AB1A 200A    325Q      JR   NZ,NOLOOP
AB1C 03      326Q      INC  BC
AB1D 0A      327Q      LD   A,(BC)
AB1E FEFE    328Q      CP   0FEH
AB20 2004    329Q      JR   NZ,NOLOOP
AB22 03      330Q      INC  BC
AB23 70      331Q      LD   (HL),B
AB24 2B      332Q      DEC  HL
AB25 71      333Q      LD   (HL),C
AB26 E1      334Q      NOLOOP POP HL
AB27 C1      335Q      POP  BC
AB28 F1      336Q      POP  AF
AB29 FB      337Q      EI
AB2A ED4D    338Q      RETI
          339Q      ;-----
AB2C 00      340Q      FLPSTA DEF B 0
          341Q      ;-----
AB2D DB0C    342Q      PINIT IN  A,(FDCSTA);MOTOR ON
AB2F DF5D    343Q      DEFW #5DDF ;TDEL / NASSYS DELAY 1 SEC
AB31 CD39AA  344Q      CALL INIT
AB34 CD77AA  345Q      CALL HOME
AB37 C9      346Q      RET
          347Q      ;-----

```

ZEAP 28Q Assembler - Symbol Table

```

0C0EH 158Q  ARG2          A9DFH 093Q  CTDEL
AAF4H 295Q  DELAY          AAF7H 298Q  DELAY1
AAF9H 299Q  DELAY2          A99AH 046Q  DSBA
000CH 159Q  FDCCMD          000FH 162Q  FDCDAT
000EH 161Q  FDCSEK          000CH 163Q  FDCSTA
000DH 160Q  FDCTRK          0003H 170Q  FHOME
00D0H 175Q  FINTQ          AB08H 313Q  FLPINT
AB2CH 340Q  FLPSTA          00E4H 177Q  FRDRK
008CH 173Q  FREAD          001BH 171Q  FSEEK
001FH 172Q  FSEEKV          00ACH 174Q  FWRITE
00F4H 176Q  FWRTRK          AA77H 217Q  HOME
AA7CH 220Q  HOMEW          AA39H 180Q  INIT
A97CH 020Q  INTVEK          AAECB 290Q  NMIR
AAF0H 292Q  NMIW          A9AFH 060Q  NODISK
AB26H 334Q  NOLOOP          A9D9H 087Q  OKREAD
AA29H 142Q  OKWRIT          A969H 007Q  PHEAS
AB2DH 342Q  PINIT          0011H 166Q  PIOAC
0010H 165Q  PLOAD          0013H 168Q  PIOBC
0012H 167Q  PIOBD          A9B6H 067Q  PREAD
A9F9H 113Q  PSE05Q          AA02H 118Q  PSEL0Q
AA09H 122Q  PSE20Q          A9E0H 105Q  PSEEK
A97EH 022Q  PSEL          A98AH 031Q  PSELA
A992H 040Q  PSELB          AA19H 133Q  PWRITE
A9E8H 101Q  RDERR          AADCH 282Q  RE66
AADFH 283Q  RE66HL          AA90H 236Q  READ
AAA5H 248Q  READW          A9C0H 073Q  RRETRY
AD0H 276Q  SAV66          AA88H 288Q  SAVE
A81H 225Q  SEEK          AA88H 229Q  SEEKW
A8DH 233Q  SETSEK          A9B4H 063Q  TSTERR
A99DH 051Q  TSTROM          AA21H 138Q  WRETRY
AB0H 256Q  WRITE          AAC5H 268Q  WRITEW
A2FH 148Q  WTRR          AA37H 152Q  WTRR1

```

# EMDOS

von HELMUT EMMELMANN

Hier nun endlich der Abdruck des Hexlistings von Helmut Emmelmann. Leider steht das Assemblerlisting noch nicht zur Verfügung.

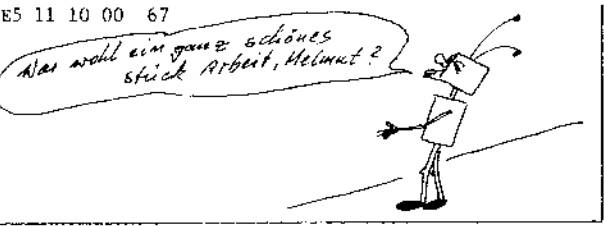
wird aber zu gegebener Zeit veröffentlicht. Ich konnte das Listing nicht vom Originalspeicherbereich machen, da der alte NASCOM ab A000 kein RAM mehr hat.

Sie sollten das Listing ab 1100 eintippen (dann stimmen auch die Prüfsummen) und dann später auf A100 kopieren. Dort ist das Programm lauffähig. Mehr zur Funktion auf der Einstiegersseite.

In Zukunft werden wir das EMDOS auch auf Diskette anbieten, wenn der Kaltstarter im DD Format so weit ist. Wer so lange warten möchte, kann sich die Mühe der Eintipperei sparen.

1100	C3	82	A1	C3	3E	A4	C3	61	C0
1108	A4	C3	70	A4	C3	69	A9	45	AE
1110	4D	44	4F	53	20	28	43	29	08
1118	20	31	39	38	33	20	42	59	D9
1120	20	48	45	4C	4D	55	54	20	40
1128	45	4D	4D	45	4C	4D	41	4E	85
1130	4E	00	84	22	21	22	23	24	BF
1138	25	00	00	00	00	00	00	00	6E
1140	00	00	00	00	00	00	00	00	51
1148	00	00	00	00	00	00	00	00	59
1150	00	00	00	00	00	00	FF	FF	5F
1158	FF	FF	FF	FF	FF	FC	07	FF	66
1160	FE	0F	FF	FF	FF	FF	FF	FF	78
1168	FF	FF	FF	FF	FF	FF	FF	FF	71
1170	FF	FF	FF	FF	FF	FF	6F	6F	59
1178	B7	C2	39	A2	C3	4B	A2	6F	FC
1180	79	AB	CD	87	A1	DF	5B	B7	9B
1188	38	50	EF	0C	0D	0D	20	20	76
1190	45	20	4D	20	44	20	4F	20	46
1198	53	20	2D	20	66	75	65	72	1B
11A0	20	20	4E	41	53	43	4F	4D	B2
11A8	20	20	49	2C	49	49	0D	0D	1A
11B0	20	20	28	43	29	20	31	39	1F
11B8	38	33	20	42	59	20	48	45	9C
11C0	4C	4D	55	54	20	45	4D	4D	12
11C8	45	4C	4D	41	4E	4E	0D	0D	AE
11D0	00	3E	37	32	87	A1	AF	CD	2C
11D8	69	A9	EF	45	4D	44	4F	53	62
11E0	20	0D	00	3E	C3	32	1D	0C	7A
11E8	21	03	A1	22	1E	0C	21	31	5C
11F0	A1	3E	04	CD	80	A4	DF	63	17
11F8	1A	FE	20	13	28	FA	47	1A	D7
1200	FE	20	20	4C	78	FE	4D	20	7F
1208	06	3E	01	CD	69	A9	C9	FE	05
1210	44	CA	A1	A2	FE	46	CA	02	83
1218	A3	FE	53	CA	65	A3	FE	45	33
1220	CA	0A	A4	FE	4C	CA	30	A3	91
1228	FE	52	CA	DD	A3	FE	57	CA	F3
1230	E1	A3	FE	42	CA	C8	A3	3E	79
1238	41	F5	EF	45	4D	44	4F	53	E7
1240	2D	45	72	72	6F	72	20	00	A9
1248	F1	DF	67	EF	0D	00	18	8A	2F
1250	1B	CD	14	A4	38	E3	3E	03	5E
1258	11	FF	FF	CD	0A	A8	38	D9	09
1260	3E	07	CD	0A	A8	B7	28	E6	FB
1268	3D	C8	3D	20	07	E1	C5	3E	C7
1270	01	C3	69	A9	3D	20	0A	21	E0
1278	DA	A1	E5	C5	3E	01	C3	69	1A
1280	A9	DF	55	21	8F	A2	22	7B	5E
1288	0C	ED	43	9F	A2	18	BF	2A	18
1290	9F	A2	23	22	9F	A2	2B	7E	12
1298	B7	37	C0	DF	78	B7	C9	00	2F
12A0	00	1A	13	FE	20	28	FA	B7	D6
12A8	28	02	D6	41	4F	3E	02	CD	57
12B0	69	A9	38	85	3E	05	CD	80	21
12B8	A4	3E	06	CD	80	A4	38	3D	18
12C0	21	00	A0	E5	7E	B7	20	2A	F7

12C8	EB 21 OF 00 19 CB 7E 20 77	14F8	3E 04 C3 69 A9 22 08 A8 F5	1738	19 06 10 7E CD 49 A7 23 DC
12D0	21 EB 23 06 08 7E B7 20 74	1500	C9 CD 7D A7 D8 2A 08 A8 81	1740	10 F9 E1 11 20 00 19 18 A3
12D8	02 3E 20 F7 23 10 F6 3E A8	1508	71 23 23 36 00 23 22 6A B9	1748	E3 E5 C5 47 1F 1F 1F E6 76
12E0	2E F7 06 03 7E B7 20 00 75	1510	A5 36 FF 06 20 23 36 00 7E	1750	1F 21 56 A1 CD 00 A8 78 8B
12E8	F7 23 10 F8 EF 20 3A 20 85	1518	10 FB CD A7 A4 CD BD A4 7E	1758	E6 07 47 3E 80 0F 10 FD 7D
12F0	20 00 E1 11 20 00 19 2C 79	1520	38 34 21 00 A0 CD 98 A7 6E	1760	47 B6 4F 7E A0 71 C1 E1 F4
12F8	2D 20 C8 18 BC C3 78 A1 CF	1528	B7 28 F2 3D 20 36 11 0F C1	1768	B7 C9 4F 47 CD 49 A7 28 7A
1300	18 2B 1A 13 FE 20 28 FA C3	1530	00 19 ED 5B 08 A8 13 13 7C	1770	09 04 78 B9 20 F6 3E 21 3A
1308	21 31 A1 3E 04 CD 80 A4 41	1538	1A 86 12 ED 5B 6A A5 06 5C	1778	37 C9 78 B7 C9 22 06 A8 57
1310	EB 2B 3E 01 CD 80 A4 DA 43	1540	10 4F 23 7E 12 13 10 FA 84	1780	23 7E 2B 0E 00 FE 3A 20 C9
1318	39 A2 11 00 50 21 00 00 88	1548	AF 12 ED 53 6A A5 79 FE E4	1788	F2 7E 23 23 22 06 A8 D6 FB
1320	E5 3E 02 CD 80 A4 E1 23 4D	1550	80 23 28 D1 18 0C B7 37 13	1790	41 4F FE 1A 3E 23 3F C9 B8
1328	DA 39 A2 18 F3 C3 DA A1 39	1558	C0 2A 6A A5 7E B7 3E 20 F9	1798	AF CB 44 C0 E5 EB 1A B7 CE
1330	CD 14 A4 3A 3E 07 DF 79 D5	1560	37 C0 B7 C9 11 20 00 19 36	17A0	20 54 13 2A 06 A8 06 08 24
1338	3E 42 38 F7 11 FF FF 3A 43	1568	18 BB 44 A1 7D 1F 1F E6 D6	17A8	1A BE 20 4A 13 23 7E FE B3
1340	0B 0C FE 02 3F 38 F1 B7 89	1570	1F 45 2A 08 A8 4E C6 03 DA	17B0	2E 28 08 FE 20 28 04 10 7F
1348	28 04 ED 5B 0C 0C 21 2F 37	1578	CD 00 A8 6E 7D 26 00 29 3C	17B8	EF 18 3B 05 28 08 1A FE 5E
1350	A4 3E 03 CD 0A A8 38 DB DA	1580	29 B7 3E 22 37 C8 78 E6 32	17C0	20 20 33 13 18 F5 7E FE B6
1358	3E 07 CD 0A A8 C5 DF 6C 3F	1588	03 CD 00 A8 CB FC B7 C9 5C	17C8	20 20 03 21 FC A7 23 06 0F
1360	E1 DF 66 18 C8 CD 14 A4 FE	1590	CD 7D A7 D8 CD A7 A4 CD 53	17D0	03 D5 1A BE 20 09 13 23 F6
1368	38 C9 21 00 00 22 10 0C DB	1598	BD A4 38 17 21 00 A0 CD EB	17D8	10 F8 E1 E1 AF 3C C9 E1 4E
1370	22 12 0C DF 79 3E 43 D8 74	15A0	98 A7 3D 20 F2 36 E5 11 6F	17E0	7E FE 42 20 11 23 7E FE 85
1378	3A 0B 0C FE 02 38 29 2A 67	15A8	10 00 19 E5 CD D6 A4 E1 F3	17E8	41 20 0B 23 7E FE 4B 20 75
1380	0C 0C EB 2A 0E 0C ED 52 19	15B0	D8 18 EC B7 C8 37 C9 CD ED	17F0	05 E1 B7 3E 02 C9 E1 11 9F
1388	EB 3E 05 CD 0A A8 EB 2A 5D	15B8	7D A7 D8 2A 08 A8 71 23 37	17F8	20 00 19 18 9B 4E 41 53 DD
1390	10 0C 3A 12 0C 4F 3E 06 AA	15C0	23 06 22 36 00 23 10 FB 84	1800	85 6F D0 24 C9 38 2F A4 D4
1398	CD 0A A8 3A 0B 0C FE 05 7E	15C8	CD A7 A4 CD 02 A7 CD BD F5	1808	31 A1 3D CA 44 A8 3D CA EC
13A0	20 04 ED 5B 14 0C 18 0D 64	15D0	A4 38 3E 21 00 A0 CD 98 25	1810	65 A8 3D CA 93 A8 3D CA 7E
13A8	B7 20 CA 3E 07 CD 0A A8 20	15D8	A7 FE 01 20 1D E5 11 09 CF	1818	1F A9 3D 28 08 3D 28 0D D6
13B0	7A B3 28 C1 EB 21 2F A4 B8	15E0	00 19 36 42 23 36 41 23 43	1820	3D 28 12 37 C9 22 62 A9 DC
13B8	3E 04 CD 0A A8 21 00 00 AD	15E8	36 4B 11 04 00 19 7E FE 28	1828	ED 53 64 A9 C9 22 66 A9 87
13C0	22 12 0C DA 39 A2 18 9B 7B	15F0	80 28 0E 23 7E 32 05 A8 3B	1830	79 32 68 A9 C9 2A 62 A9 02
13C8	CD 14 A4 38 F6 11 D6 10 85	15F8	18 07 FE 02 20 0E E5 36 75	1838	ED 5B 64 A9 ED 4B 66 A9 EC
13D0	2A D6 10 3E 01 32 0B 0C 7B	1600	E5 CD D6 A4 E1 D8 11 20 2C	1840	3A 68 A9 C9 3E 01 CD 89 01
13D8	32 12 0C 18 A9 3E 01 18 53	1608	00 19 18 CA CD 29 A7 18 CE	1848	A8 D8 21 00 00 CD 57 A8 CD
13E0	02 3E 02 32 01 A4 CD 14 ED	1610	BD B7 37 C0 B7 C9 D5 2A 10	1850	D0 FE 22 37 C0 B7 C9 78 47
13E8	A4 38 D8 DF 79 3E 44 38 C1	1618	08 A8 4E 23 23 46 3E 02 F8	1858	B7 C8 3E 02 CD 89 A8 D8 05
13F0	D2 3A 0B 0C FE 02 20 F5 3B	1620	80 77 78 1F 1F 1F E6 1F 07	1860	23 05 14 18 F2 3E 08 CD D1
13F8	ED 5B 0C 0C ED 4B 0E 0C BD	1628	CD 00 A8 23 7E B7 28 0F 42	1868	89 A8 D8 27 7A A8 CD 7C 16
1400	3E 02 21 2F A4 CD 0A A8 C7	1630	6F 26 00 29 29 78 1F E6 AA	1870	A8 D8 2A 7A A8 3E 09 C3 5E
1408	18 B9 CD 14 A4 3E 0B CD 88	1638	03 CD 00 A8 D1 18 14 78 3B	1878	80 A4 2F A4 78 B7 C8 3E BC
1410	80 A4 18 F4 1A 13 FE 20 9F	1640	B7 28 03 2B 7E 23 F5 CD B6	1880	0A CD 89 A8 D8 14 05 18 A9
1418	28 FA 21 2F A4 06 0E 77 CD	1648	6A A7 E1 D1 D8 77 6F 26 05	1888	F3 E5 D5 C5 CD 80 A4 C1 C4
1420	23 1A 13 FE 20 28 02 10 DC	1650	00 29 29 CB FC 3E 04 C3 84	1890	D1 E1 C9 3E 01 CD 89 A8 60
1428	F6 36 20 21 2F A4 C9 47 8C	1658	69 A9 CD 7D A7 CD A7 A4 89	1898	D8 ED 53 1C A9 11 00 A0 3E
1430	41 42 49 20 33 20 2E 42 F3	1660	3E B7 32 B3 A6 CD BD A4 24	18A0	21 00 00 3E 02 CD 80 A4 0A
1438	41 4B 20 94 84 94 D6 60 DA	1668	38 12 21 00 A0 7E FE E5 EA	18A8	D8 21 00 A0 3E 11 06 04 B2
1440	D2 0A A8 6E 20 DA 80 A4 BC	1670	28 10 11 20 00 19 CB 44 17	18B0	BE 28 04 3E 40 37 C9 87 B7
1448	C6 20 DA 69 A9 C6 20 3F 53	1678	28 F3 18 F9 B7 37 3E 24 YA	18B8	23 10 F5 11 62 A9 01 08 1D
1450	D8 B7 CA 87 A1 3D 28 02 4C	1680	C0 C9 36 00 23 EB 2A 06 93	18C0	00 ED B0 2A 1C A9 24 20 A8
1458	37 C9 21 00 A0 11 55 A1 34	1688	A8 06 08 7E FE 20 28 0B 23	18C8	0B 2C 20 05 2A 62 A9 18 89
1460	C9 3A 0E 0C 2A 10 0C ED C4	1690	FE 2E 28 07 12 23 13 10 59	18D0	06 2C 28 46 2A 1C A9 E5 5C
1468	5B 12 0C CD 69 A9 DF 5B 0E	1698	F2 18 06 3E 20 12 13 10 51	18D8	2A 64 A9 11 10 FF B7 ED EB
1470	7E FE 3A C2 40 FF E5 D5 F5	16A0	FA 7E FE 2E 28 03 21 FC A2	18E0	5A 28 02 38 10 ED 4B 64 60
1478	C5 CD 87 A1 C1 D1 E1 C9 82	16A8	A7 06 03 23 7E 12 23 13 57	18E8	A9 78 B1 28 2D 21 10 A0 F8
1480	B7 C8 3D 28 7C 3D 28 5F B8	16B0	10 FA EB B7 38 2B 36 00 0B	18F0	D1 ED B0 B7 C9 D1 E5 21 CD
1488	3D 28 67 3D 28 6F 3D 28 A1	16B8	23 36 00 23 36 00 23 ED 90	18F8	10 A0 03 F0 00 ED B0 C1 0F
1490	16 3D 28 29 3D 28 3F 3D 29	16C0	5B 08 A8 13 13 1A CB 7F 6B	1900	21 01 00 CD 57 A8 D8 79 58
1498	CA B7 A5 3D CA 5A A6 3D 16	16C8	28 02 3E 80 77 23 13 01 74	1908	B7 C8 D5 11 00 A0 3E 02 66
14A0	CA 16 A6 3D CA 90 A5 79 EF	16D0	10 00 EB ED B0 B7 F2 D6 FD	1910	CD 89 A8 D1 D8 21 00 A0 91
14A8	32 E6 A4 3E 02 CD 69 A9 97	16D8	A4 3E 37 32 B3 A6 EB 18 95	1918	ED B0 B7 C9 FF FF 3E 08 92
14B0	D8 7E 3D 32 F4 A4 3C 23 70	16E0	95 36 01 23 36 00 23 36 74	1920	CD 89 A8 D8 27 7A A8 21 74
14B8	86 32 E5 A4 C9 21 E4 A4 7F	16E8	08 23 E5 2A 08 A8 23 23 26	1928	00 A0 3E 11 06 04 77 23 D4
14C0	34 3A E5 A4 BE 28 0B 6E 2A	16F0	7E CB BF 11 11 00 19 D1 1A	1930	87 10 FB D5 EB 21 62 A9 C7
14C8	26 00 11 00 A0 3E 03 C3 B7	16F8	12 13 01 10 00 ED B0 C3 A4	1938	01 08 00 ED B0 E1 11 10 F9
14D0	69 A9 35 AF 37 C9 21 E4 DF	1700	D6 A4 21 55 A1 71 23 06 42	1940	A0 01 F0 00 ED B0 11 00 98
14D8	A4 6E 26 00 11 00 A0 3E 13	1708	1E 36 00 23 10 FB 3E 02 E1	1948	A0 3E 0A CD 89 A8 D8 E5 04
14E0	04 C3 69 A9 25 26 00 D5 ED	1710	CD 69 A9 23 7E 0F 0F 47 0C	1950	2A 64 A9 11 10 FF 19 2B 04
14E8	CD 6C A5 D1 D8 3E 03 C3 87	1718	78 3D CD 49 A7 10 F9 23 CD	1958	24 44 D1 CD 7C A8 D8 C3 36
14F0	69 A9 D5 CD 6C A5 D1 D8 72	1720	46 04 C8 78 CD 49 A7 18 96	1960	72 A8 00 10 00 10 00 00 B3
		1728	F8 21 00 A0 CB 44 C0 7E 45	1968	00
		1730	FE E5 28 0F E5 11 10 00 67		



# Floppy- Hilfsprogramme

Wie Helmut Emmelmann schon in Heft 7/8 erwähnt hat, sollen im Laufe der Zeit Hilfsprogramme erstellt werden, die per Befehl von der Diskette geladen und sofort ausgeführt werden. So wird der Befehlsvorrat des Betriebssystems gewaltig erweitert, ohne daß dafür auf Dauer Speicherplatz zur Verfügung gestellt werden müßte. So erübrigt sich auch der Vorschlag, den Günter Kreidl anfang des Jahres machte, einen erweiterten Monitor zu programmieren. Die erwünschten Funktionen können als selbständige Programme auf der Systemdiskette auf ihren Aufruf warten. Zwei dieser Programme sind bereits fertig.

1. Ein Formatierprogramm für Single Density.

## EAP Z80 Assembler - Source Listing

```

0010 ;FORMATIERPROGRAMM FDC VER.1.1
0020 ;NACH ROUTINEN AUS JOURNAL 9/83
0030 ;10 SEKTOREN SINGLE DENSITY (OSBORNE)
0040 ;GUENTER BOHM 26.11.83
0050 ;-----
3168 000C 0060 FDCCMD EQU 0CH
3168 000D 0070 FDCTR EQU 0DH
3168 000E 0080 FDCSEK EQU 0EH
3168 000F 0090 FDCDAT EQU 0FH
3168 000C 0100 FDCSTA EQU 0CH
0110 ;
3168 0010 0120 PLOAD EQU 10H
3168 0011 0130 PIOAC EQU 11H
3168 0012 0140 PIOBD EQU 12H
3168 0013 0150 PIOBC EQU 13H
0160 ;
3168 0003 0170 FHOME EQU 03 ;SPEED STEPPER MOTOR
3168 001B 0180 FSEEK EQU 1BH
3168 001F 0190 FSEEKV EQU 1FH
3168 000D 0200 FINT0 EQU 0D0H
3168 00F4 0210 FWRTRK EQU 0F4H
0220 ;
0230 ;-----
8000 0240 ORG 8000H
8000 A680 0250 DEFW FLPINT ;INTERRUPTVEKTOR
8002 C36580 0260 START JP FORMAT
8005 F3 0270 INIT DJ
8006 3ECF 0280 LD A,0CFH
8008 D311 0290 OUT (PIOAC),A
800A 3EC0 0300 LD A,0C0H
800C D311 0310 OUT (PIOAC),A
0320 ;
800E 3ECF 0330 LD A,0CFH
8010 D313 0340 OUT (PIOBC),A
8012 3EF0 0350 LD A,0F0H
8014 D313 0360 OUT (PIOBC),A
0370 ;
8016 3EB7 0380 LD A,0B7H
8018 D311 0390 OUT (PIOAC),A
801A 3E7F 0400 LD A,7FH
801C D311 0410 OUT (PIOAC),A
0420 ;
801E 3E08 0430 LD A,8
8020 D310 0440 OUT (PIOAD),A
8022 0E01 0450 LD C,1
8024 CD9280 0460 CALL DELAY
8027 3E28 0470 LD A,28H
8029 D310 0480 OUT (PIOAD),A
802B 3ED0 0490 LD A,FINT0
802D D30C 0500 OUT (FDCCMD),A
802F E3 0510 EX (SP),HL
8030 E3 0520 EX (SP),HL
8031 DB0C 0530 IN A,(FDCSTA)
8033 3E80 0540 LD A,80H
8035 ED47 0550 LD I,A
8037 3E90 0560 LD A,00
8039 D311 0570 OUT (PIOAC),A
803B 3E31 0580 LD A,31H ;LAUFWERKNUMMER HIER 1
803D D310 0590 OUT (PIOAD),A
803F FB 0600 EI
8040 ED5E 0610 IM 2;INTERRUPT MODE
8042 C9 0620 RET
0630 ;
8043 3E03 0640 HOME LD A,FHOME
8045 D30C 0650 OUT (FDCCMD),A
8047 FB 0660 EI
8048 18FE 0670 HOMEW JR HOMEW
804A DB0C 0680 IN A,(FDCSTA)
804C C9 0690 RET
0700 ;
804D D30F 0710 SEEK OUT (FDCDAT),A ;TRACK NUMBER
804F 3E1B 0720 LD A,FSEEK
8051 FB 0730 EI
8052 D30C 0740 OUT (FDCCMD),A
8054 18FE 0750 SEERW JR SEERW
8056 DB0C 0760 IN A,(FDCSTA)
8058 C9 0770 RET
0780 ;
8059 D30F 0790 SETSEK OUT (FDCSEK),A
805B C9 0800 RET
0810 ;
805C C5 0820 SETDEN PUSH BC
805D CB47 0830 BIT 0,A
805F 0E10 0840 LD C,10H
8061 2802 0850 JR Z,SETDEL
8063 0E00 0860 LD C,0
8065 DB10 0870 SETDEL IN A,(PIOAD)
8067 E6EF 0880 AND 0EFH
8069 B1 0890 OR C
806A D310 0900 OUT (PIOAD),A
806C C1 0910 POP BC
806D C9 0920 RET
0930 ;
806E 216600 0940 SAV66 LD HL,66H
8071 118680 0950 LD DE,SAVE
8074 010400 0960 LD BC,4
8077 EDB0 0970 LDIR
8079 C9 0980 RET
0990 ;
807A 218680 1000 RE66 LD HL,SAVE
807D 116600 1010 RE66HL LD DE,66H
8080 010400 1020 LD BC,4
8083 EDB0 1030 LDIR
8085 C9 1040 RET
1050 ;
8086 0004 1060 SAVE DEFS 4
808A EDA2 1070 NMJR INI
808C ED45 1080 RETN
808E EDA3 1090 NMJW OUTI
8090 ED45 1100 RETN
1110 ;
8092 E5 1120 DELAY PUSH HL
8093 D5 1130 PUSH DE
8094 C5 1140 PUSH BC
8095 0664 1150 DELAY1 LD B,100
8097 17 1160 DELAY2 RLA
8098 29 1170 ADD HL,HL
8099 29 1180 ADD HL,HL
809A 05 1190 DEC B
809B C29780 1200 JP NZ,DELAY2
809E 0D 1210 DEC C
809F C29580 1220 JP NZ,DELAY1
80A2 C1 1230 POP BC
80A3 D1 1240 POP DE
80A4 E1 1250 POP HI
80A5 C9 1260 RET
1270 ;
1280 ; INTERRUPTROUTINE AUSGEL0EST VOM FDC
1290 ;

```



80A6 F5	1300	FLPINT	PUSH AF	813A CD7D80	2030	CALL KE66HL
80A7 C5	1310		PUSH BC	813D AF	2040 F050	XOR A ; BEGINNE MIT SPUR 0
80A8 E5	1320		PUSH HL	813E 329481	2050 F100	LD (FTRK),A ; ZWISCHENSPP.
80A9 210600	1330		LD HL,6	8141 CD4D80	2060	CALL SEEK
80AC 39	1340		ADD HL,SP	8144 CD5C81	2070	CALL EXPTRK ; SPUR IM SP. AUFBAUEN
80AD 4E	1350		LD C,(HL)	8147 CDBF81	2080	CALL WRTRK ; SPUR SCHREIBEN
80AE 23	1360		INC HL	814A DB0C	2090	IN A (FDCSTA)
80AF 46	1370		LD B,(HL)	814C B7	2100	OR A
80B0 0A	1380		LD A,(BC)	814D C21C81	2110	JP NZ RETERR ;WRITE TRACK ERROR
80B1 FE18	1390		CP 18H	8150 3A9481	2120	LD A,(FTRK) ; NAECHSTE SPUR
80B3 200A	1400		JR NZ,NOLOOP	8153 3C	2130	INC A
80B5 03	1410		INC BC	8154 FE28	2140	CP 40
80B6 0A	1420		LD A,(BC)	8156 20F6	2150	JR NZ,F100
80B7 FEFE	1430		CP 0FEH	8158 CD7A80	2160	CALL RE66
80B9 2004	1440		JR NZ,NOLOOP	815B C9	2170	RET ;ZU EMDOS
80BB 03	1450		INC BC		2180 ;	
80BC 70	1460		LD (HL),B		2190 ;	EXPAND THE FORMAT-TABLE IN MEMORY
80BD 2B	1470		DEC HL		2200 ;	
80BE 71	1480		LD (HL),C	815C 110010	2210	EXPTRK LD DE,BUF
80BF E1	1490	NOLOOP	POP HL	815F 218C81	2220	LD HL,TRKTAB
80C0 C1	1500		POP BC	8162 CD7C81	2230	CALL EXPAND ; INDEX PART
80C1 F1	1510		POP AF	8165 010100	2240	LD BC,1 ; SEKT. 1
80C2 FB	1520		EI	8168 21AE81	2250	TRKSEC LD HL,SEKTAF-1
80C3 ED4D	1530		RETI	816B 09	2260	ADD HL,BC
	1540 ;			816C 7E	2270	LD A,(HL)
80C5 CD0580	1550	FORMAT	CALL INIT	816D 329881	2280	LD (SECNR),A
80C8 EF	1560		RST #28	8170 218F81	2290	LD HL,TRKSTB
80C9 0D	1570		DEFB #D	8173 CD7C81	2300	CALL EXPAND ; EXPAND SEKTOR PART
80CA 464F524D	1580		DEFM "FORMAT	8176 0C	2310	INC C
4154				8177 79	2320	LD A,C
80D0 0D	1590		DEFB #D	8178 FE0B	2330	CP 10+1 ;10 SEKTOREN FERTIG?
80D1 4C415546	1600		DEFM "LAUFWERK (A-C)	817A 38EC	2340	JR C,TRKSEC
5745524B					2350 ;	
2028412D				817C 7E	2360	EXPAND LD A,(HL)
4329				817D 23	2370	INC HL
80DF 0D00	1610		DEFW #D	817E B7	2380	OR A
80E1 DF7B	1620		DEFW #7BDF ;BLINK/ TASTATUREINGABE	817F C8	2390	RET Z
80E3 FE41	1630		CP "A	8180 C5	2400	PUSH BC
80E5 2004	1640		JR NZ LFW2	8181 47	2410	LD B,A
				8182 7E	2420	LD A,(HL)
80E7 3E31	1650		LD A,#31 ;BEACHT LAUFWERK TAB.9/83	8183 23	2430	INC HL
80E9 180C	1660		JR DRIVE	8184 12	2440	EXPA LD (DE),A
80EB FE42	1670	LFW2	CP "B	8185 13	2450	INC DE
80ED 2004	1680		JR NZ LFW3	8186 05	2460	DEC B
80EF 3E32	1690		LD A,#32	8187 20FB	2470	JR NZ,EXPA
80F1 1804	1700		JR DRIVE	8189 C1	2480	POP BC
80F3 FE43	1710	LFW3	CP "C	818A 18F0	2490	JR EXPAND
80F5 20CE	1720		JR NZ FORMAT		2500 ;	
80F7 D310	1730	DRIVE	OUT (PIOD),A	818C 20FF	2510	TRKTAB DEFB 32,#FF
	1740 ;			818E 00	2520	DEFB 0
80F9 3E00	1750		LD A,0 ;SINGLE DENSITY	818F 0600	2530	TRKSTB DEFB 6,0
80FB CD5C80	1760		CALL SETDEN	8191 01FE	2540	DEFB 1,0FEH
80FE 3E03	1770		LD A,PHOME	8193 01	2550	DEFB 1
8100 DB0C	1780		OUT (FDCSTA),A ;HOME	8194 00	2560	FTRK DEFB 0
8102 DF5D	1790		DEFW #5DDF ;NASSYS DELAY 1 SEC TDEL	8195 0100	2570	DEFB 1,0
8104 DB0C	1800		IN A,(FDCSTA)	8197 01	2580	DEFB 1
8106 CB57	1810		BIT 2,A ;KEIN LAUFWERK DA	8198 00	2590	SECNR DEFB 0
8108 2004	1820		JR NZ LFWOK	8199 0101	2600	DEFB 1,1 ;SEKTORLAENGE 256
810A 3E1F	1830		LD A,#1F	819B 01F7	2610	DEFB 1,0F7H
810C 180E	1840		JR RETERR	819D 00BF	2620	DEFB 11,#FF
810E CB4F	1850	LFWOK	BIT 1,A	819F 0600	2630	DEFB 6,0
8110 2804	1860		JR Z DISKOK	81A1 01FB	2640	DEFB 1,0FBH
8112 3E10	1870		LD A,#10 ;KEINE DISKETTE DA	81A3 FFE5	2650	DEFB 255,0E5H ; DATA
8114 1806	1880		JR RETERR	81A5 01E5	2660	DEFB 1,0E5H ;256.
8116 CB77	1890	DISKOK	BIT 6,A	81A7 01F7	2670	DEFB 1,0F7H
8118 281A	1900		JR Z TSTOK	81A9 11FF	2680	DEFB 17,#FF
811A 3E15	1910		LD A,#15 ;WRITE PROTECT	81AB 00	2690	DEFB 0
811C F5	1920	RETERR	PUSH AF	81AC 6EVE	2700	DEFB 110,#FF
811D EF	1930		RST #28	81AE 00	2710	DEFB 0
811E 0D	1940		DEFB #D		2720 ;	
811F 464F524D	1950		DEFM "FORMATIERFEHLER "		2730 ;	
41544945				81AF 01020304	2740	SEKTAF DEFB 1,2,3,4,5,6,7,8,9,10
52464548				05060708		
4C455220				090A		
812F 00	1960		DEFB 0		2750 ;	
8130 F1	1970		POP AF		2760 ;	GANZE SPUR ZUM FORMATIEREN SCHREIBEN
8131 DF68	1980		DEFW #68DF ;PRINT A		2770 ;	
8133 C9	1990		RET ;ZU EMDOS	81B9 0E0F	2780	WRTRK LD C,FDCDAT
	2000 ;			81BB 3EFA	2790	LD A,0F4H ;WRITE A TRACK COMMAND
8134 CD6E80	2010	TSTOK	CALL SAY66	81BD FB	2800	EI
8137 218E80	2020		LD HL,NM1W	81BE 210010	2810	LD HL,BUF

```

81C1 130C 2820 OUT (FDCCMD),A
81C3 18FE 2830 WRTTRW JR WRTTRW
81C5 C9 2840 RET
81C6 1000 2850 BUF EQU 1000H

```

Es könnte zwar gewaltig kürzer sein, wenn es die Unterprogramme aus PHEAS benutzen würde. Da PHEAS aber sicher noch einige Änderungen erfährt, ist es besser, wenn das Formatierprogramm nicht in Abhängigkeit immer wieder neu assembliert werden muß.

Wenn EMDOS bei Ihnen läuft, wird das Formatierprogramm mit

```
S FORMAT.COM 8000 81C6 8002 3
```

auf der Diskette abgespeichert. Der Befehl FORMAT.COM lädt es dann automatisch ein und startet es. Falls Sie nur ein Laufwerk angeschlossen haben, dürfen Sie nicht vergessen, die Systemdiskette aus dem Laufwerk zu nehmen, sonst wird sie formatiert und Ihre Programme sind futsch (ist mir beim Tetsten passiert). Bei zwei Laufwerken wird einfach bei der Auswahl B gedrückt, und der Systemdiskette in Laufwerk A, kann nichts geschehen.

Wenn wir die benötigten Informationen von mc haben, wird das Programm für das mc DD Format umgestrickt, und steht Ihnen selbstverständlich wieder zur Verfügung.

2.STAT.COM von Helmut Emmelmann gibt eine sehr genaue Auskunft über die Diskette: Filenamen, Speicherbereich, verbleibender Platz etc. Sie müssen sich das selbst anschauen, ist wirklich sehr komfortabel.

Auf die Diskette wird das Programm geladen mit

```
S STAT.COM 1200 15D0 1200 3
```

Der Aufruf erfolgt wiederum mit dem Programmnamen.

```

1200 15E0
1200 EF 0C 20 20 53 5F 54 5F B2
1208 41 5F 54 5F 5F 5F 5F 2D B7
1210 5F 5F 5F 5F 66 75 65 72 50
1218 5F 5F 45 5F 4D 5F 44 5F DB
1220 4F 5F 53 0D 0D 20 20 28 B5
1228 43 29 20 20 31 39 38 33 BB
1230 20 48 2E 45 4D 4D 45 4C 48
1238 4D 41 4E 4E 0D 0D 4C 61 3B
1240 75 66 77 65 72 6B 20 3F 45
1248 20 28 41 2C 42 2E 2E 2E DB
1250 29 2E 2E 2E 2E 2E 2E 00 9F
1258 DF 7B FE 0D 20 02 3E 41 70
1260 D6 41 4F 32 A5 15 3E 22 24

```

```

1268 CD 1D 0C 30 20 FE 17 C2 97
1270 51 15 EF 46 61 6C 73 63 C0
1278 68 65 20 4C 61 61 75 66 77 76
1280 65 72 6B 73 61 6E 67 61 DE
1288 62 65 0D 00 C9 EF 0D 0D 40
1290 20 20 20 4C 61 75 66 77 01
1298 65 72 6B 20 20 00 3A A5 0B
12A0 15 C6 41 F7 EF 3A 0D 0D 08
12A8 4C 61 65 6E 67 65 20 53 79
12B0 79 73 74 65 6D 73 70 75 4C
12B8 72 65 6E 2E 2E 2E 3A 20 F3
12C0 20 00 7E 57 1E 00 CD E7 99
12C8 14 EF 0D 4C 61 65 6E 67 D1
12D0 65 20 44 69 72 65 63 74 C2
12D8 6F 72 79 2E 2E 2E 2E 2E 2A
12E0 2E 3A 20 20 00 23 7E 57 92
12E8 1E 00 CD E7 14 7E 1F 1F 9C
12F0 E6 3F 32 A3 15 7E EB 6F E9
12F8 26 00 29 29 29 EF 0D 41 E8
1300 6E 7A 61 68 6C 20 44 69 FD
1308 72 2E 20 45 69 6E 74 72 DD
1310 61 65 67 65 2E 3A 20 20 5B
1318 00 CD 18 15 EF 0D 44 69 CE
1320 73 6B 65 74 74 65 6E 67 98
1328 72 6F 65 73 73 65 2E 2E 28
1330 2E 2E 2E 2E 3A 20 20 00 75
1338 13 1A 3C 32 A4 15 6F 26 34
1340 00 CD 18 15 EF 6B 0D 0D C1
1348 00 CD B9 14 AF 32 A2 15 8D
1350 32 9C 15 EF 0C 49 6E 68 60
1358 61 6C 74 20 64 65 72 20 27
1360 44 69 73 6B 65 74 74 65 B0
1368 20 69 6E 20 20 4C 57 20 75
1370 00 3A A5 15 C6 41 F7 EF 64
1378 3A 0D 0D 00 3E 01 CD 1D 08
1380 0C 22 9E 15 3A A5 15 4F B7
1388 3E 45 CD 1D 0C DA 51 15 54
1390 3E 46 CD 1D 0C 30 06 B7 0A
1398 C2 51 15 18 63 2A 9E 15 2B
13A0 22 A0 15 22 A0 15 7E FE DD
13A8 E5 28 48 E5 11 0F 00 19 2E
13B0 EB E1 1A CB 7F 20 3C 23 72
13B8 06 08 CD A1 14 EF 2E 00 78
13C0 06 03 CD A1 14 EF 20 20 8D
13C8 20 00 7E 0E 00 B7 28 02 68
13D0 0E 80 63 23 23 7E 81 B7 D0
13D8 1F 1E 00 CB 1B 57 21 A3 29
13E0 15 C6 03 1F 1F E6 3F 86 BA
13E8 77 CD E7 14 21 9C 15 34 40
13F0 CD AC 14 2A A0 15 11 20 A0
13F8 00 19 CB 44 28 A5 18 90 A8
1400 21 A2 15 7E FE 05 D4 B9 FA
1408 14 EF 0D 44 69 65 73 20 D1
1410 73 69 6E 64 20 00 2A 9C B8
1418 15 26 00 CD 18 15 EF 20 70
1420 46 69 6C 65 73 0D 0D 42 83
1428 65 6C 65 67 74 65 72 20 44
1430 53 70 65 69 63 68 65 72 77
1438 70 6C 61 74 7A 20 3A 20 F1
1440 20 00 3A A3 15 26 00 6F FB
1448 CD 18 15 EF 6B 0D 0D 46 10
1450 72 65 69 65 72 20 20 20 DB
1458 53 70 65 69 63 68 65 72 9F
1460 70 6C 61 74 7A 20 3A 20 19
1468 20 00 3A A4 15 21 A3 15 68
1470 96 6F 26 00 CD 18 15 EF 98
1478 6B 0D 0D 2A 2A 2A 2A 2A E3
1480 5F 5F 53 20 54 20 41 20 9A
1488 54 20 20 20 20 20 62 65 57
1490 65 6E 64 65 74 5F 5F 2A 9C
1498 2A 2A 2A 2A 00 CD B9 14 EE
14A0 C9 7E B7 20 02 3E 20 F7 29
14A8 23 10 F6 C9 EF 0D 00 21 CB
14B0 A2 15 34 7E FE 0D D8 36 46
14B8 00 EF 0D 2A 2A 2A 5F 04
14C0 57 65 69 74 65 72 20 6D D1
14C8 69 74 20 62 65 6C 69 65 DA
14D0 62 69 67 65 72 20 54 61 C2
14D8 73 74 65 5F 5F 2A 2A 2A 74
14E0 00 DF 7B EF 0D 00 C9 E5 F8
14E8 EB E5 DF 66 EF 48 20 20 88
14F0 00 E1 CD 1D 15 EF 20 20 13
14F8 00 7C C6 03 1F 1F E6 3F B4

```

```

1500 26 00 6F CD 18 15 EF 6B FE
1508 00 E1 C9 11 00 00 B7 ED 7C
1510 42 38 03 13 18 F8 09 C9 97
1518 E5 D5 C5 18 15 E5 D5 C5 58
1520 01 10 27 CD 0B 15 CD 4C 73
1528 15 01 E8 03 CD 0B 15 CD F8
1530 4C 15 01 64 00 CD 0B 15 F8
1538 CD 4C 15 01 0A 00 CD 0B 5E
1540 15 CD 4C 15 5D CD 4C 15 23
1548 C1 D1 E1 C9 7B C6 30 F7 01
1550 C9 F5 EF 0D 45 6D 64 6F A4
1558 73 20 20 46 65 68 6C 65 04
1560 72 20 4E 72 2E 20 00 F1 06
1568 DF 67 EF 20 62 65 69 6D 6F
1570 20 53 54 41 54 2D 42 65 B5
1578 66 65 68 6C 0D 0D 2A 2A 9A
1580 2A 2A 5F 5F 53 54 41 54 E3
1588 20 20 20 20 41 42 42 52 34
1590 55 43 48 5F 5F 2A 2A 2A C1
1598 2A 0D 00 C9 00 00 00 00 AD
15A0 00 00 00 00 00 00 00 A8 5D
15A8 A9 AA AB B2 AD AE AF B0 27
15B0 B1 00 B3 B8 B5 B6 B7 BC BF
15B8 B9 BA BB BF BD BE 00 00 35
15C0 C1 C2 C3 3A C5 C6 C7 1B C2
15C8 FF FF FF FF FF FF FF FF D5
15D0 FF FF FF FF FF FF FF FF DD
15D8 FF FF FF FF FF FF FF FF E5

```

# EMDOS- Anpassung

von HELMUT EMMELMANN

Anpassung von EMDOS an das jeweilige Floppy-System

Folgende Aenderungen sind notwendig:

- relocieren des DOS (dazu liegen 2 Versionen des DOS mit verschiedenen Startadressen vor)
- schreiben bzw. anpassen des Moduls PHEAS. Es liegt am Ende des DOS und kann so einfach ersetzt werden.
- schreiben eines Bootstrap-Loaders, der das DOS und NASSYS in den Speicher laedt.
- schreiben eines Formatierungsprogramms zum Initialisieren der Disketten
- evt. aendern einiger Anwendungsprogramme zum direkten Aufruf des DOS. (z.B. BASIC, ZEAP, NASSYS, PASCAL ...)

Relocieren des DOS:

Das DOS liegt in Versionen fuer D000-E000 und A000-B000 vor. Es kann durch Vergleich der beiden Versionen an jeden 4k-Block gelegt werden.

Modul PHEAS:

Das Modul PHEAS uebernimmt die Funktionen des Sektorschreibens- und lesens.

Diskettenformat: n Sektoren je 256 Bytes

n/4 Groups je 4 Sektoren=1k

Ein Sektor wird durch eine fortlaufend vergebene Sektornummer identifiziert. PHEAS muss daraus Spur- und Sektornummer innerhalb der Spur berechnen. ACHTUNG !!! bei CP/M-kompatiblen Formaten muss der Skew-Faktor beruecksichtigt werden (=versetzte Sektornummern auf der Spur).

Die ersten m Sektoren einer Diskette sind fuer das System reserviert und werden nur beim Booten verwendet. Die folgenden Groups enthalten das Inhaltsverzeichnis. Das DOS verwendet oft Sektornummern relativ zum 1. Directory- Sektor. Diese Sektornummern sind durch ein gesetztes BIT 15 gekennzeichnet.

Pheas hat nur einen Einsprungpunkt; die jeweilige Funktion wird durch eine Nummer im Akku angegeben.

Pheas-Aufrufe :

A=0 Initialisiere Pheas, beim Systemstart aufgerufen

A=1 Schalte Motoren und evt. Interrupt aus (DOS verlassen)

A=2 Select Drive (C= Drivenumber 0h, 1h .. , ergibt -> HL Adresse DSB Disk- Steuer - Block s,u.)

A=3 Lese Sektor (DE Adresse; HL Sektornummer (Bit 7 des H-Registers zeigt relative Sektornummer an))

A=4 Schreibe Sektor (DE Adresse; HL Sektornummer sh, oben)

Aufbau des DSB (fuer jeden Laufwerkstyp einen)

1. Byte Sektornummer des 1. Sektors des Directories

2. Byte Laenge des Inhaltsverzeichnisses in Sektoren

3. Byte Nummer der letzten Group auf der Diskette

1. Group enthaelt den Beginn des Directories und hat die Nummer 0.

Weitere Informationen ueber Pheas sh. Pheas-Geruest.

Fehlerbehandlung :

Im ganzen DOS, so auch im Pheas, werden auftretende Fehler folgendermassen behan-

delt:

Carry = 1 wenn Fehler sonst 0

bei Fehler A=Fehlernummer (fuer Pheas stehen die Fehlernummern 10-1F zur Verfuegung).

Der Bootstap-Loader liest aus den ersten Sektoren der Diskette das DOS in den dafuer vorgesehenen Speicherbereich. Moeglicherweise wird auch NASSYS mitgeladen (sh. beiliegenden Sourcecode).

Das Formatierungs-Programm

Die Diskette wird initialisiert, die Sektoren des Inhaltsverzeichnisses muessen mit 0E5H aufgefuellt werden.

Anpassung BASIC :

Nach dem Initialisieren des Basic ist in die Adresse 1055h-56h anstelle von FF40h die Adresse BASCAL (sh. Maschinensprache-Schnittstelle) einzutragen. Vor der Initialisierung des Basics kann BASCAL auch an die Adresse E335 eingetragen werden (bei BASIC im RAM sinnvoll).

Anpassung anderer Software:

Die Moeglichkeiten, das DOS von Maschinensprache aus aufzurufen sind gesondert beschrieben.

Hardware Aenderungen:

Neben dem Anschluss des Controllers wird eine kleine Schaltung zur Ausblendung des Monitor-Roms benoetigt. (ueber diese Moeglichkeiten koennen Sie einiges in Heft 6 nachlesen).

Fuer EMDOS :

Boot-Eprom → Monitor Sockel

Ausblendung des Boot-Eproms und ersetzen durch RAM

entweder statisch oder 4k Page der dynamischen RAM-Karte

Fuer CP/M :

Ersetzen der gesamten unteren 4k Page 0h-1000h durch RAM

Bei Bildschirmzugriffen wird kurzzeitig zurueckgeschaltet.

EMDOS Speicherbelegung:

D000-D100 Puffer frei, solange DOS nicht aktiv

D100 Einsprungpunkt von NASSYS aus

D103 Einsprungpunkt fuer DOS-Aufrufe (sh.

EMDOS.MAC)

D106 NASSYS-PHEAS-Aufrufe

E D109 A ; HL; DE Register entsprechend ARG2-ARG4 versorgen und Aufruf von PHEAS (zum Test von PHEAS und zum direkten Sektorzugriff.)

D109 BASCAL Einsprungpunkt von BASIC

D10C Sprung zu PHEAS

PHEAS DIENT ZUM LESEN UND SCHREIBEN DER EINZELNEN SEKTOREN

ENTSPRECHEND SEKTORNUMMER. DIE DISKETTE BESTEHT AUS VIELEN 256-BYTES LANGEN LOGISCHEN SEKTOREN. DIE GRUPPENGROESSE UMFASST 4 SEKTOREN. BEI CP/M KOMPATIBLEN DISKETTENFORMATEN MUSS DER SKEW FAKTOR BERUECKSICHTIGT WERDEN !

PHEAS AUFRUFE ;

A= AUFRUFSNUMMER

A=0 INIT PHEAS (NACH LADEN AUFGERUFEN)

A=1 ABORT PHEAS (INTERRUPT LOESCHEN, SODASS PHEAS GELOESCHT WERDEN KANN)

A=2 SELECT DISK (C=DISKNO.; HL WIRD ADRESSE DSB

A=3 SEKTOR LESEN; HL SEKTORNUMMER ;BIT 15 = 1 ; RELATIV

DE ADRESSE DATEN

A=4 SEKTOR SCHREIBEN

HL SEKTORNUMMER

DE SPEICHERADRESSE

DSB DISKETTEN STEUER BLOCK

FUER JEDEN LAUFWERKSTYP EINEN, FORMAT S.H. UNTEN

\*\*\*\*\*

Z80

PHEAS:: OR A; JP Z,PINIT \* INIT PHEAS

DEC A; JP Z,PABORT \* ABORT PHEAS

DEC A; JP Z,PSEL \* SELECT DISK

DEC A; JP Z,READ \* SEKTOR LESEN

DEC A; JP Z,WRITE \* SEKTOR SCHREIBEN

SCF; RET

PINIT: RET \* NICHT VERWENDET

PSEL: \* LAUFWERK SELEKTIEREN

LD A,C; CP NDRIVE; LD A,1; CCF; RET C

LD A,C; LD (INDRV),A

LD HL,DSB; RET

READ: \* SEKTOR LESEN

BIT 7,H; CALL NZ,DIRADD; JP DREAD

ABORT: RET \* HIER MOTOREN AUSSCHALTEN !!!

DIRADD: RES 7,H; LD A,30; JP DADA \*  
DIRECTORY-ANFANG ADDIEREN

DREAD: \* HL SEKTORNUMMER DE ADRESSE  
SEKTOR LESEN

DADA: ADD A,L ;Addition HL=HL+A  
LD L,A  
RET NC  
INC H

UNTERPROGRAMM SEKT  
SPUR UND SEKTOR BERECHENEN HL: SEKTOR-  
NUMMER

SEKT: PUSH BC  
LD BC,-10  
LD A,B  
SEKT1: INC A  
ADD HL,BC  
JR C,SEKT9  
LD (TRACK),A ; SPURNUMMER ABSPEICHERN  
CP 3  
JR NC,SEKT8 ;BEI SPUR 0,1,2 KEIN SKEW  
FAKTOR  
LD A,L  
ADD A,10  
SEKT6: LD (SECT),A  
POP BC  
JR SEKT9  
SEKT8: LD BC,SKEWTAB+10  
ADD HL,BC  
LD A,(HL) ;PHYSIKALISCHE SEKTORNUMMER  
JR SEEK6  
SEKT9: RET ; ....

SKEWTAB:DB \* LOGISCHER -> PHYSIKALISCHER  
SEKTOR

; SEKTOR SCHREIBEN

WRITE: ; DE ADRESSE HL SEKTORNUMMER  
BIT 7,H  
CALL NZ,DIRADD

RET

DSB: DB 30 ;1. SEKTOR DIRECTORY  
DB 8 ;LAENGE INHALTSVERZEICHNIS IN SEKTO-  
REN  
DB 91 ;LETZTE GROUP  
;  
END

# NASCOM-Video

VON TOM D. RÜDEBUSCH

NASCOM 2 (1) Video- Verbesserung

Zwei einfache Änderungen am Video- Teil des NASCOM 2 werten die Bildschirmdarstellung so auf, daß außer der geringen Auflösung von 48 Zeichen/Zeile wirklich keine Wünsche mehr offen bleiben.

1. Bei längerem Arbeiten äußerst unangenehm ist das "Schwimmen" des Bildschirms, das besonders im rechten unteren Teil auffällt. Es wird dadurch verursacht, daß das Bild nicht mit exakt 50 Hz sondern mit 50,15 Hz aufgefrischt wird.

Außerdem werden von den einzelnen Zeichen nicht 16 sondern nur 14 Zeilen ausgegeben. Den Männchen aus dem Grafik-Zeichensatz fehlen deshalb die Füße. Beide Schönheitsfehler wurden von NASCOM in Kauf genommen, um eine Umschaltung von 50 auf 60 Hz zu ermöglichen und so Kompatibilität mit zwei Fernsehnormen zu gewährleisten.

Mit der Festlegung auf 50 Hz bewirkt folgende Änderung ein absolut ruhiges Bild, auf dem alle Zeichen vollständig ausgegeben werden:

Folgende IC- Beinchen sind herauszubiegen

IC 53, Pin 1 und 11

IC 56, Pin 5 und 6

IC 68, Pin 1 und 10

Außerdem sind diese Verbindungen herzustellen:

IC 68, Pin 1 - Pin 8

IC 68, Pin 10 - Pin 16

IC 53, Pin 11 - IC 68, Pin 5 - IC 56, Pin 5

IC 56, Pin 6 - IC 44, Pin 11

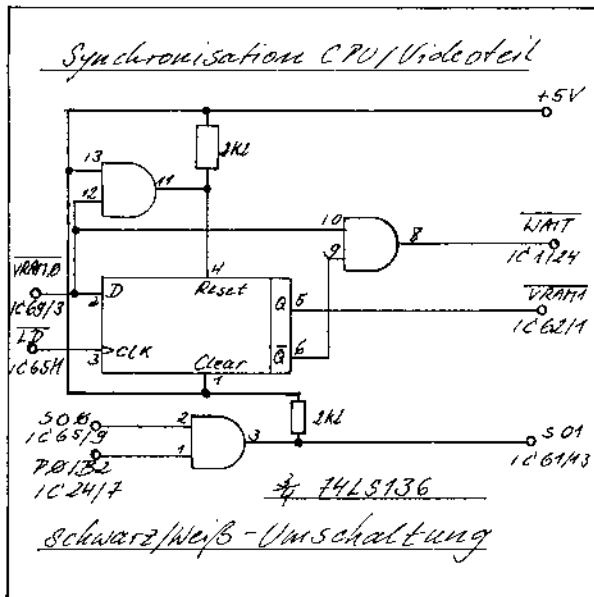
IC 53, Pin 1 - IC 68, Pin 11

2. Sowohl bei NASCOM 1 als auch bei NASCOM 2 sind der Zugriff von CPU und Video- Teil auf das Video- RAM nicht synchronisiert.

Bei häufigen Zugriffen der CPU resultieren daraus beim NASCOM 1 die bekannten weißen und beim NASCOM 2 die schwarzen Streifen auf dem Bildschirm, die jede noch so schöne Grafikdarstellung verstümmeln.

Mit unten beschriebener Schaltung bekommt man eine saubere bewegte Grafik bei Spielen; und auch durchlaufende Listings lassen sich besser mitlesen.

Außerdem wurde ein freies Gatter dazu verwendet, mit Port0/B2 den Bildschirm zwischen Normal- und Inversdarstellung umschalten zu können. Dabei ist zu beachten, daß NASSYS bei jeder Tastaturabfrage Port0 auf den Wert in 0C00H zurücksetzt. Für Testzwecke trägt man dort mit dem Modify-Befehl "4" für invers und "0" für Normaldarstellung ein. RESET bewirkt ebenfalls Normaldarstellung. Oben erwähnte Synchronisation von CPU und Videoteil wird durch die Ausgabe von WAIT-Zyklen gewährleistet. Da der Prozessor warten muß, bis der Videoteil eine neue Zeile eines Zeichens in das Schieberegister zur seriellen Ausgabe geladen hat, ist die Geschwindigkeitseinbuße auch bei häufigem Bildschirmzugriff kaum meßbar.



Sollten bei Einsatz der Schaltung ab und zu irgendwelche unerwünschten Zeichen auf dem Bildschirm erscheinen, so bekommt das Videoram, bedingt durch die jetzt leicht verzögerte Adreßumschaltung, seine Freigabe zu früh. Einfache Abhilfe für dieses Timing-Problem: ein 22pF-Kondensator von IC 69, Pin 8 nach Masse.

Alle IC- und Pinangaben beziehen sich übrigens auf den NASCOM 2, beim NASCOM 1 sind sie analog umzusetzen.

Nun aber zur Schaltung:

IC 65, Pin 9 und IC 69, Pin 3 sind herauszubiegen, IC 58 wird nicht mehr benötigt und wandert in die Bastelkiste. Dann sind Ver-

bindungen zur Hauptplatine entsprechend der Schaltungsskizze herzustellen.

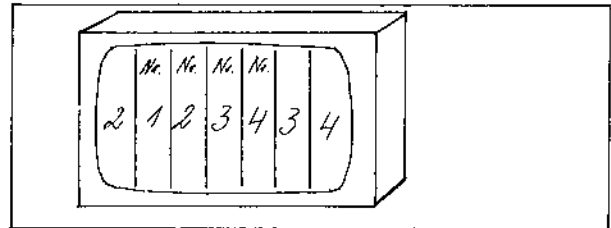
Wer beide Änderungen durchgeführt hat, kann sich nun an einer wirklich schönen Bildschirmdarstellung erfreuen. (Natürlich funktioniert jede Modifikation auch für sich allein).

## Grafik

VON JÖRG WITTICH

Mit der folgenden Schaltung, deren Aufbau weniger als DM 80,- kosten soll, haben Sie gleichzeitig eine 8K Speichererweiterung, einen programmierbaren Zeichengenerator und - wenn Sie über ein Bit von Port0 auf 4 Generatoren umschalten - eine hochauflösende Grafik von 256x256 Punkten.

Jedem der vier Zeichengeneratoren sind auf dem Bildschirm abwechselnd senkrechte Streifen von 8 Zeichen Breite zugeordnet. Dadurch lassen sich auf dem Bildschirm 65536 einzelne Bildpunkte ansprechen.



Es sind auch folgende Formate möglich:

320x160 Punkte + 6 Zeilen

384x128 Punkte + 8 Zeilen Text.

Im Vergleich dazu:

280x160 Punkte + 4 Zeilen oder

280x192 Punkte bei APPLE

320x200 Punkte bei VC 64

Falls die Schaltung von einigen Lesern nachgebaut wird und weiteres Interesse besteht, können wir die entsprechende Grafik-Software dazu liefern. Es existiert auch eine Anpassung an X-TAL BASIC, ähnlich wie in früheren Heften beschrieben, jedoch mit der oben erwähnten feinen Auflösung.

Auch eine Erweiterung auf Grauwerte könnten wir liefern, die beim Autor schon seit einiger Zeit für SC-Television benutzt wird und ausgezeichnete Bildqualität liefert.

Sollte bei einem Nachbau sogar ein Platinenlayout herausbringen, wären wir sehr dankbar für eine Veröffentlichung im Journal.



# Hetzjagd

VON PETER BRENDDEL

```
10 REM * persoenlicher Rekord:Staerke 23 *
20 REM P, B,
30 FORI=3200TO3218STEP2:READA:DOKEI, A:NEXT
40 DOKE4100, 3200:CLS
50 SCREEN15, 5:PRINT" H E T Z J A G D
60 PRINT
70 PRINT"Du wirst von PNKTEN gejagt. Mit den Cur
so";
80 PRINT"rtasten":PRINT"musst du Ihnen ausweich
en, ";
90 PRINT" und in die Nische":PRINT"rechts gel
angen
110 PRINT:INPUT" SCHWIERIGKEITSGRAD";S
120 N=1/S*300
130 A=2058:B=2105:C=2058:D=2954:G=183:CLS
140 FORI=ATODSTEP11:POKEI, 128:NEXT
150 FORI=ATODSTEP17:POKEI, 128:NEXT
160 FORI=1TO2
170 FORJ=ATOBSTEP2:DOKEJ, -1:NEXT
180 A=A+896:B=B+896
190 FORK=CTODSTEP64:POKEK, 255:NEXT
200 C=C+47:D=D+47:NEXT:POKE2554, 248:POKE2553, 32
210 M(0)=2220:M(1)=2230:M(2)=2731:M(3)=2741
220 P=2451:POKEP, 6
230 FORI=0TO3
240 IFM(I)=PTHEN260
250 GOTO280
260 IFM(I)-P=23THEND=-1:GOTO300
270 D=-64:GOTO300
280 IFP-M(I)=23THEND=1:GOTO300
290 D=64:GOTO300
300 IFPEEK(M(I)+D)=32THEN440
310 POKEM(I), 32:M(I)=M(I)+D:POKEM(I), 185
320 NEXT
330 S=USR(0)
340 IFS=17THENE=-1:G=181
350 IFS=18THENE=1:G=182
360 IFS=19THENE=-64:G=183
370 IFS=20THENE=64:G=183
380 IFS=0THENE=0
390 IFPEEK(P+E)=255THEN230
400 IFPEEK(P+E)=128THEN230
410 IFPEEK(P+E)=248THEN530
420 POKEP, 32:P=P+E:POKEP, 6
430 GOTO230
```

```
440 IFPEEK(M(I)+D)=GTHEN480
450 POKEM(I)+D, 255
460 IFQ=NTHENG=0:GOTO310
470 D=0:Q=Q+1:GOTO320
480 SCREEN12, 1:PRINT" VERLOREN - nochmal? "
490 S=USR(0)
500 IFS=74THEN130
510 IFS=78THENEND
520 GOTO490
530 SCREEN15, 15:PRINT" GEWONNEN! - nochmal? ";
540 GOTO490
550 DATA289, 1548, 13833, 8960, -1264, 25311, 288
560 DATA-20665, -3389, 240
```

## EPROM Port II

VON HORST DIECKHOFF

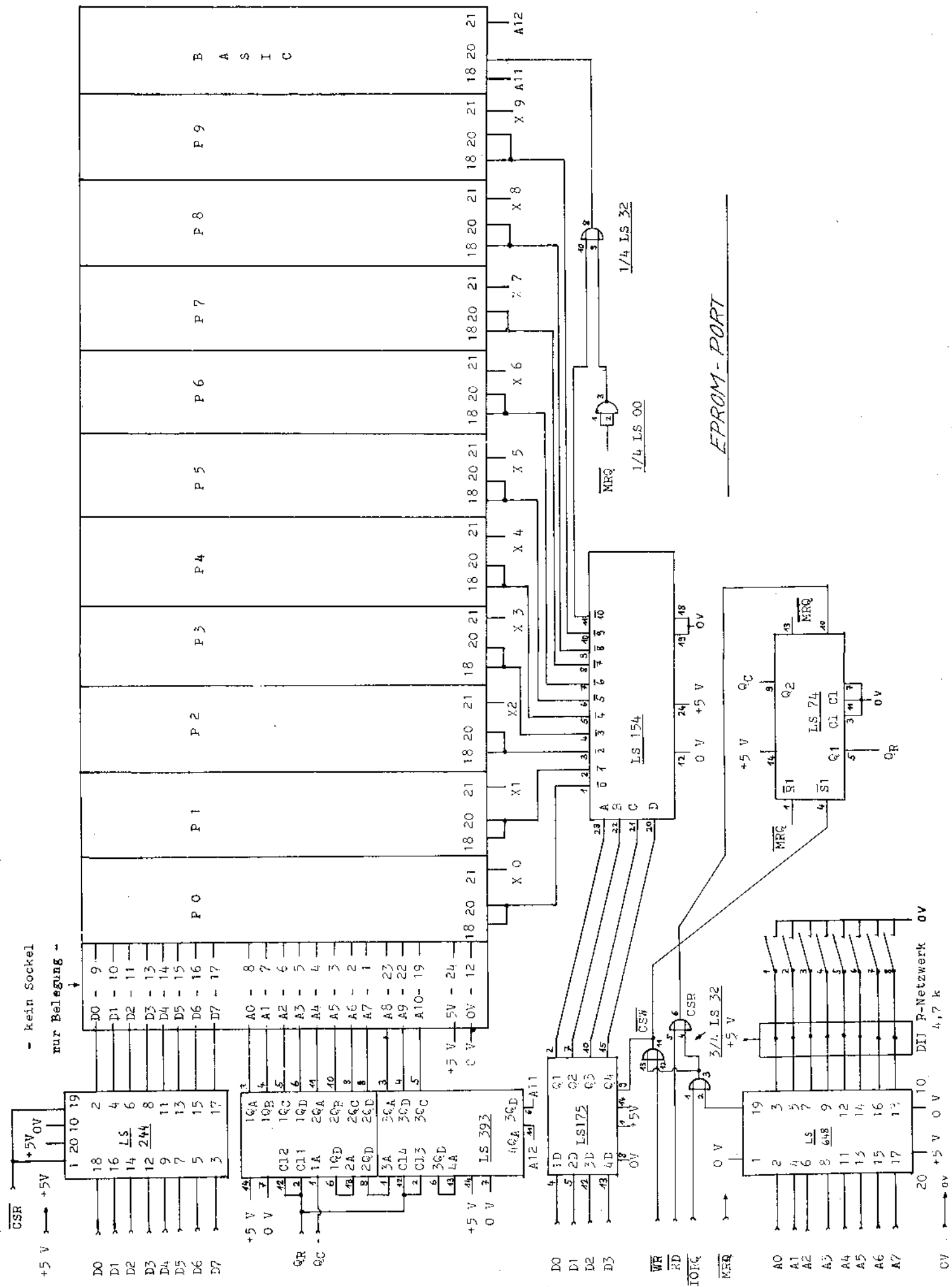
In Heft 6/83 wurde der erste Teil der EPROM-Port Erweiterung veröffentlicht. Der Beschreibung folgt hiermit der Schaltplan und die Kernroutine zum Lesen der ECB-Port Karte. Horst Dieckhoff hat auch ein komplettes Menuprogramm geschrieben, das aber von der Cassette leider nicht zu entziffern war. Wir verweisen desha lb bei Interesse auf den Autor oder werden, falls größere Nachfrage besteht, das Assemblerlisting zu späterem Zeitpunkt nachdrucken.

Das Menu ruft ein externes Uhrenprogramm auf, das die aktuelle Zeit und Datum aus einer gepufferten ECB- Uhrenkarte liest und in die Kopfzeile einträgt.

ZEAP Z80 Assembler - Source Listing

```
1000          9000          ORG #1000
1000 00F1      9005 IOPOEP EQU #F1 ; Portadresse der Karte
          9010 ; Umladen von der EPROM-Karte in den
          9020 ; RAM Bereich des Systems
          9030 ; Unterprog. Parameter bei der Übergabe :
          9040 ; HL - Adresse an die geladen werden soll
          9050 ; D - EPROMadresse auf der Karte
          9060 ; E - Länge des EPROMs in #100
          9070 ;
1000 AF       9100 UMLADE XOR A
1001 47       9105 LD B,A
1002 0EF1     9110 LD C,IOPOEP ; IO Adresse EPROM
1004 ED51     9130 OUT (C),D ; PORT Karte
1006 EDB2     9140 UMLADL INIR
1008 1D       9150 DEC E
1009 20FB     9160 JR NZ,UMLADL
100B 3EFF     9170 LD A,#FF
100D ED79     9180 OUT (C),A
100F C9       9190 RET
```





# Kalender

```

10 REM GREGORIANISCHER KALENDER
20 REM AUS NUOVA ELETTRONICA NR.84/85
30 REM ANGEPASST AUF NASCOM VON
40 REM HENRIK MAIER, BADEN-BADEN
50 REM ██████████

90 CLEAR 4000: DIM M$(12), L(12), A$(12,50)
100 CLS : PRINT "GREGORIANISCHER KALENDER : "
110 PRINT: PRINT: INPUT "Fuer welches Jahr "; Y
120 IF Y=0 THEN Y=1983 : REM LAUFENDES JAHR
130 IF Y < 100 THEN Y=Y+1900: GOTO150
140 IF Y<1583 THEN PRINT: PRINT "Erst ab 1583 !": SCR
EEN1,2: GOTO110
150 SCREEN30,1: PRINT "anno "; Y: SCREEN1,5
160 PRINT: INPUT "Ausdruck auf UART-Drucker "; A$
170 IF LEFT$(A$,1)="J" THEN UART=1
180 INPUT "Bild erwuenscht "; A$
190 IF LEFT$(A$,1)="N" THEN BILD=1
200 CLS
210 IF UART THEN PRINT: INPUT "Drucker bereit "; A$
220 CLS: SCREEN1,15
230 IF UART THEN POKE3187,125: POKE3189,133
240 FOR I=1 TO 10: PRINT: NEXT I
250 SCREEN1,6: G$="MO DI MI DO FR SA SO "
260 M$(1)="+ J A N U A R +"
270 M$(2)="+ F E B R U A R +"
280 M$(3)="+ M A E R Z +"
290 M$(4)="+ A P R I L +"
300 M$(5)="+ M A I +"
310 M$(6)="+ J U N I +"
320 M$(7)="+ J U L I +"
330 M$(8)="+ A U G U S T +"
340 M$(9)="+ S E P T E M B E R +"
350 M$(10)="+ O K T O B E R +"
360 M$(11)="+ N O V E M B E R +"
370 M$(12)="+ D E Z E M B E R +"
380 L(1)=31 : L(2)=28 : L(3)=31: L(4)=30
390 L(5)=31 : L(6)=30 : L(7)=31: L(8)=31
400 L(9)=30 : L(10)=31 : L(11)=30 : L(12)=31
410 V1=365*Y+INT((Y-1)/4)-INT(3*(INT((Y-1)/100)
+1)/4)
420 B=V1-INT(V1/7)*7 : IF B=0 THEN B=7
430 Y1=Y: R=Y1/100: S=INT(R): IF (R-S)=0 THEN Y1=R
440 D=Y1/4: C=INT(D): IF D-C=0 THEN L(2)=29
450 DATA " 1 ", " 2 ", " 3 ", " 4 ", " 5 ", " 6 "
460 DATA " 7 ", " 8 ", " 9 ", "10 ", "11 ", "12 "
470 DATA "13 ", "14 ", "15 ", "16 ", "17 "
480 DATA "18 ", "19 ", "20 ", "21 ", "22 "
490 DATA "23 ", "24 ", "25 ", "26 "
500 DATA "27 ", "28 ", "29 ", "30 ", "31 "
510 GOSUB680
520 IF BILD=0 THEN GOSUB730
530 REM **KALENDER AUSDRUCK**
540 PRINTAB(26) "*****": PRINTAB(26) " Y "
550 PRINTAB(26) "*****"
560 PRINT: PRINT
570 FOR I=1 TO 12: PRINT: PRINT: PRINT: PRINT: PRINTAB
(21)M$(I)
580 PRINT: PRINT: PRINTAB(21): G$
590 PRINT: F=1: H=7: FOR C=1 TO 6: PRINTAB(21): : FOR N
=F TO H
600 IF A$(I,N)=" " THEN A$(I,N)=" "
610 PRINTAB(I,N): : NEXT N: PRINT: F=H+1: H=H+7
620 NEXT C: NEXT I
630 FOR I=1 TO 10: PRINT: NEXT I
640 IF UART THEN POKE3187,127: POKE3189,130
650 PRINT: PRINT: PRINT: PRINT: INPUT "Noch ei
n Ausdruck "; A$
660 IF LEFT$(A$,1)="N" THEN CLS: END
670 RUN
680 FOR X=1 TO 12: T=L(X)+B-1: RESTORE
690 FOR I=8 TO 6: READAS(X,I): NEXT I
700 IFT<35 THEN T1=27
710 IFT>34 THEN T1=34
720 B=T-T1: NEXT X: RETURN
730 READ0

```

```

740 IF N= -99 THEN1370
750 IF N= -1 THEN800
760 IF N= -2 THEN820
770 IF N= -3 THEN830
780 IF N= 0 THEN F=0: PRINT: GOTO730
790 GOTO1370
800 READ A: F=F+1: IF F=1 THEN A=A+0
810 PRINTSPC(A): GOTO730
820 READ B: FOR I=1 TO 3: PRINT "!": NEXT I: GOTO73
0
830 READ C: FOR I=1 TO C: PRINT "$": NEXT I: GOTO
730
840 DATA -1, 15, -3, 1, -2, 1, -3, 1, -1, 6, -3, 1, -2, 1, -
3, 1, 0
850 DATA -1, 14, -3, 1, -2, 1, -3, 1, -1, 3, -3, 1, -2, 0
860 DATA -3, 1, 0
870 DATA -1, 12, -3, 1, -2, 3, -3, 1, -1, 1, -3, 1, -2, 12, -
3, 1, 0
880 DATA -1, 10, -3, 1, -2, 4, -3, 1, -1, 2, -3, 1, -2, 14, -
3, 1, 0
890 DATA -1, 9, -3, 1, -2, 4, -3, 1, -1, 4, -3, 1, -2, 14, -3
, 1, 0
900 DATA -1, 7, -3, 1, -2, 5, -3, 1, -1, 4, -3, 1, -2, 16, -3
, 1, 0
910 DATA -1, 6, -3, 1, -2, 4, -3, 1, -1, 7, -3, 1, -2, 16, -3
, 1, 0
920 DATA -1, 5, -3, 1, -2, 4, -3, 1, -1, 7, -3, 1, -2, 17, -3
, 1, 0
930 DATA -1, 6, -3, 1, -2, 6, -3, 1, -1, 4, -3, 1
940 DATA -2, 17, -3, 1, 0
950 DATA -1, 7, -3, 1, -2, 8, -3, 1, -1, 1, -3, 1, -2, 17, -3
, 1, 0
960 DATA -1, 9, -3, 1, -2, 8, -3, 1, -2, 17, -3, 1, 0
970 DATA -1, 11, -3, 1, -2, 8, -3, 1, -2, 16, -3, 1, 0
980 DATA -1, 12, -3, 1, -2, 9, -3, 1, -2, 14, -3, 1, -1, 5, -
3, 1, 0
990 DATA -1, 14, -3, 1, -2, 8, -3, 1, -2, 14, -3, 6, 0
1000 DATA -1, 15, -3, 1, -2, 8, -3, 1, -2, 17, -3, 1, 0
1010 DATA -1, 16, -3, 1, -2, 7, -3, 1, -2, 17, -3, 1, 0
1020 DATA -1, 15, -3, 2, -2, 8, -3, 1, -2, 15, -3, 1, 0
1030 DATA -1, 14, -3, 1, -2, 2, -3, 1, -2, 8, -3, 1, -2, 12,
-3, 1, 0
1040 DATA -1, 12, -3, 1, -2, 4, -3, 1, -2, 9, -3, 1, -2, 5, -
3, 1, 0
1050 DATA -1, 10, -3, 1, -2, 21, -3, 1, 0
1060 DATA -1, 11, -3, 1, -2, 20, -3, 1, 0
1070 DATA -1, 12, -3, 1, -2, 4, -3, 1, -2, 13, -3, 1, 0
1080 DATA -1, 14, -3, 3, -2, 14, -3, 1, 0
1090 DATA -1, 16, -3, 1, -2, 14, -3, 1, 0
1100 DATA -1, 17, -3, 1, -2, 13, -3, 1, 0
1110 DATA -1, 18, -3, 1, -2, 12, -3, 1, 0
1120 DATA -1, 19, -3, 1, -2, 11, -3, 1, 0
1130 DATA -1, 20, -3, 1, -2, 10, -3, 1, 0
1140 DATA -1, 20, -3, 1, -2, 11, -3, 1, 0
1150 DATA -1, 21, -3, 1, -2, 11, -3, 1, 0
1160 DATA -1, 22, -3, 1, -2, 11, -3, 1, 0
1170 DATA -1, 22, -3, 1, -2, 14, -3, 1, 0
1180 DATA -1, 22, -3, 1, -2, 16, -3, 1, 0
1190 DATA -1, 23, -3, 1, -2, 17, -3, 1, 0
1200 DATA -1, 23, -3, 1, -2, 15, -3, 1, -2, 2, -3, 1, 0
1210 DATA -1, 24, -3, 1, -2, 15, -3, 1, -2, 2, -3, 1, 0
1220 DATA -1, 24, -3, 1, -2, 16, -3, 1, -2, 2, -3, 1, 0
1230 DATA -1, 23, -3, 1, -2, 18, -3, 1, -2, 1, -3, 1, 0
1240 DATA -1, 22, -3, 1, -2, 18, -3, 1, -2, 2, -3, 1, 0
1250 DATA -1, 21, -3, 1, -2, 19, -3, 1, -2, 2, -3, 1, 0
1260 DATA -1, 20, -3, 1, -2, 19, -3, 1, -2, 2, -3, 1, 0
1270 DATA -1, 19, -3, 1, -2, 19, -3, 1, -2, 1, -3, 2, 0
1280 DATA -1, 18, -3, 1, -2, 18, -3, 1, -2, 5, -3, 1, 0
1290 DATA -1, 17, -3, 1, -2, 17, -3, 1, -2, 8, -3, 1, 0
1300 DATA -1, 16, -3, 1, -2, 16, -3, 1, -2, 17, -3, 1, 0
1310 DATA -1, 16, -3, 1, -2, 15, -3, 1, -2, 12, -3, 1, 0
1320 DATA -1, 15, -3, 1, -2, 14, -3, 1, -2, 14, -3, 1, 0
1330 DATA -1, 15, -3, 1, -2, 12, -3, 1, -1, 2, -3, 1, -2, 13
, -3, 1, 0
1340 DATA -1, 15, -3, 1, -2, 11, -3, 1, -1, 4, -3, 1, -2, 12
, -3, 1, 0
1350 DATA -1, 15, -3, 1, -2, 10, -3, 1, -1, 6, -3, 1, -2, 12
, -3, 1, 0
1360 DATA -1, 15, -3, 1, -2, 9, -3, 1, -1, 8, -3, 1, -2, 11,
-3, 1, -99
1370 FOR S=1 TO 8 : PRINT: NEXT S: RETURN

```



```

3380 LET T9= INT (S1+365.25*S3+A(S2)+1E-02*S2-3E-02)
3400 REM * Korrektur da 1900 KEIN SCHALTJAHR
3420 IF S3=0 IF S2<=2 LET T9=T9+1
3440 REM * DER WIEVIELTE WOCHENTAG
3460 LET R9=T9-INT ((T9-1)/7)*7
3480 REM * ANFANGS ADRESSE IM STRING D$
3500 LET S9=R9*10-9
3520 RETURN
3540 REM *****
3560 REM * UNTERPROGRAMM 1
3580 REM * EINGABE 6 stellig ?
3600 IF LEN T9$<6 GOTO 3760
3620 REM * S1->TAG S2->MONAT S3->JAHR
3640 LET S1=T9$(1,2)
3660 LET S2=T9$(3,4)
3680 LET S3=T9$(5,6)
3700 REM * TAG UND MONAT GÜLTIG ?
3720 IF S1=1 IF S1<=31 IF S2=1 IF S2<=12 GOTO 3800
3740 REM * fehlerhaft
3760 RETURN
3780 REM * gültig
3800 RETURN +1
3820 REM *****
3840 REM * PROGRAMM ENDE
3860 REM *****
3880 REM * CLEAR SCREEN UND JUMP TO ENDE
3900 REM *****
3920 PRINT 'CS';
3940 CHAIN "INT" ($TOP) o.ä.

```

Ich plane gerade die Rechnerkopplung (bidirektionaler PIO- Mode 2 mit Interrupt) zwischen NASCOM II und einem ECB- System (Janich & Klass CPU I). Der ECB- Rechner soll als Floppyrechner, Druckerspooler und allgemeiner E/A- Rechner fungieren. Hat vielleicht schon Jemand Ähnliches probiert und gebaut?  
 Horst Die ckhoff, [REDACTED]

*Sonst auf einer Cassette Verpackung  
 von der Redaktion:  
 "Heute Sonne !!!  
 Ist kein Klopapier sondern  
 Handtuch." Sehr bewegend!*

```

2280 REM *****
2300 REM * Laden 1. JANUAR des eingeebenen Jahres
2320 REM *****
2340 LET T9$="0101",T$(5,6)
2360 REM *****
2380 REM * Sprung UP1 Eingabe überprüfen
2400 REM *****
2420 REM * PLATZHALTER, REM bei fehlerhaftem Datum
2440 REM *****
2460 GOSUB 3600
2480 REM *****
2500 GOSUB 3280 /* SPRUNG UP2
2520 REM *****
2540 REM * BERECHNEN DER KALENDERWOCHE
2560 REM *****
2580 LET K=INT ((T-T9+R9-1)/7)+1
2600 IF R9>4 LET K=K-1
2620 REM *****
2640 REM * bis KW 52 inc.
2660 REM *****
2680 IF K'53 GOTO 2080
2700 REM *****
2720 REM * Prüfen ob KW 53 oder KW 1 vom Folgejahr
2740 REM *****
2760 LET T9$(5,6)=S3+1 USING "##"
2780 REM *****
2800 REM * WELCHER TAG IST DER 1. JAN. (XX+1)?
2820 REM *****
2840 REM * PLATZHALTER bei fehlerhaftem Datum
2860 REM *****
2880 GOSUB 3600 /*UP1
2900 REM *****
2920 GOSUB 3380 /*UP2
2940 REM *****
2960 REM * WOCHENTAG <= DONNERSTAG -> KW 1
2980 REM *****
3000 IF R9<=4 LET K=1
3020 REM *****
3040 REM * ENTSCHIEDEN OB KW 52 ODER KW 53
3060 REM *****
3080 IF K=0 IF R9=5 LET K=53
3100 IF K=0 IF R9=6 IF FRA (S3/4)=.25 LET K=53
3120 IF K=0 LET K=52
3140 REM *****
3160 REM * AUSGABE : T$(1,2)=TAG
3180 REM * T$(3,4)=MONAT
3200 REM * T$(5,6)=JAHR
3220 REM * D$(S,S+9)=WOCHENTAG
3240 REM *****
3260 PRINT USING "##"; TAB (0,5);T$(1,2);".":T$(3,4);".":T$(5,6);" = ";D$(S,S+9);
3280 PRINT TAB (20,5); "KW";K;
3300 GOTO 1600
3320 REM *****
3340 REM * UNTERPROGRAMM 2
3360 REM * BERECHNEN ANZAHL TAGE SEIT DEM 1.1.1900

```

```

840 IF ERR 0 GOTO 3920
920 DIM D$(70),T$(6),T9$(6),B$(5)
940 DIM A(12)
1060 LET B$="←376←231←←376←237←"
1140 LET D$="MONTAG     DIENSTAG   MITTWOCH  DONNERSTAGFREITAG  SAMSTAG   SONNTAG   "
1240 DATA 0,31,59,90,120,151,181,212,243,273,304,334
1320 FOR I=1 TO 12
1340   READ A(I)
1360 NEXT I
1440 PRINT 'CF';
1600 PRINT TAB (0,3);"DATUM      : TMMJJ";B$;
1620 INPUT TAB (13,3),T9$
1700 IF T9$="" GOTO 3920
1840 GOSUB 3600
1860 GOTO 1600
1940 GOSUB 3380
2020 LET T$=T9$
2100 LET T=T9
2180 LET R=R9
2260 LET S=S9
2340 LET T9$="0101",T$(5,6)
2460 GOSUB 3600
2470 REM
2500 GOSUB 3380
2580 LET K= INT ((T-T9+R9-1)/7)+1
2600 IF R9>4 LET K=K-1
2680 IF K<53 GOTO 3080
2760 LET T9$(5,6)=S3+1 USING "##"
2880 GOSUB 3600
2890 REM
2920 GOSUB 3380
3000 IF R9<=4 LET K=1
3080 IF K=0 IF R9=5 LET K=53
3100 IF K=0 IF R9=6 IF FRA (S3/4)=.25 LET K=53
3120 IF K=0 LET K=52
3260 PRINT USING "##"; TAB (0,5);T$(1,2);"-";T$(3,4);".19";T$(5,6);" = ";D$(S,S+9);
3280 PRINT TAB (30,5);"KW";K;
3300 GOTO 1600
3380 LET T9= INT (S1+365.25*S3+A(S2)+1E-02*S2-3E-02)
3420 IF S3=0 IF S2=2 LET T9=T9+1
3460 LET R9=T9- INT ((T9-1)/7)*7
3500 LET S9=R9*10-9
3520 RETURN
3600 IF LEN T9$<6 GOTO 3760
3640 LET S1=T9$(1,2)
3660 LET S2=T9$(3,4)
3680 LET S3=T9$(5,6)
3720 IF S1>=1 IF S1<=31 IF S2>=1 IF S2<=12 GOTO 3800
3760 RETURN
3800 RETURN +1
3920 PRINT 'CS';
3940 CHAIN "INT" (S70?)

```

## PASCAL-Drucker<sup>®</sup> von JÜRGEN WEIERMANN

ZEAP Z80 Assembler - Source Listing

	0010 ;Z-80		E162 FE0D	0210	CF	ODH
	0020 ;NASCOM		E164 2B03	0220	JK	Z LNEND
	0030 ;Integrierung der Drucker-		E166 DF	0230	RST	1BH
	0040 ;Routine und der AUTO-INIT-R		E167 6F	0240	DEFB	6FH
	0050 ;in das BLS-PASCAL Betriebes		E168 C9	0250	RET	
	0060 ;Juergen Weiermann 1/83		E169 DF	0260 LNEND	RST	1BH
	0080 ;AUTO-INIT		E16A 6F	0270	DEFB	5FH
	0090 ;		E16B 3E0A	0280	LD	A,DAH
			E16D DF	0290	RST	1BH
			E16E 6F	0300	DEFB	6FH
			E16F 3E0D	0310	LD	A,DAH
E157	0100	ORG #E157	E171 C9	0320	RET	
E157 00	0110	NOP				
E158 DF	0120	INIT RST 1BH		0340 ;Auftrag fuer AUTO-INIT		
E159 71	0130	DEFB 71H		0350 ;		
E15A 13	0140	INC DE	E727	0360	DEB	6FH
E15B 2163E1	0150	LD HL,#E163	E727 CDE8E1	0370	CALL	INIT
E15E 22780C	0160	LD (#0C/8),HL				
E161 C9	0170	RET				
	0180 ;					
	0190 ;DRUCKER-ROUTINE					

ZEAP Z80 Assembler - Symbol Table

E158H 0120 INIT	E169H 0260 LNEND
-----------------	------------------

# PASCAL-Spiel

VON MICHAEL BACH

Das Spiel macht deutlich, wie ein geladenes Teilchen im Kraftfeld von Ionen hin- und hergerissen ist. Von links oben kommt ein Elektron, das zwischen positiven und negativen Ionen hindurch möglichst weit nach rechts unten fliegen soll (das gibt Punkte). Die Physik stimmt nicht ganz, man kann es sich auch als eine Ebene mit Hügeln und Tälern vorstellen. Erweiterung als Spiel mit mehreren Personen machen sicherlich viel Spaß.

```
1 PROGRAM GELADEN (*18.11.83*)
2 (* SIMULATION DER BEWEGUNG EINES ELEKTRONS IN EINER
3 MIT LADUNGEN BESETZTEN FLAECHE. DIE ANFANGSFLUG-
4 RICHTUNG KANN MIT CURSORTASTEN GEWAHLT WERDEN.
5 M.Bach, [REDACTED], [REDACTED] Gundelfingen.
6 Nach BYTE 7/82. *)
7
8
9 CONST XMAX=95; YMAX=47; XMIN=0; YMIN=3;
10 MAXION=6;
11 VI=1.0; (*ANFANGSGESCHWINDIGKEIT*)
12 CG=0.2; (*FELDKONSTANTE*)
13 VAR I: INTEGER;
14 C: STRING(1.);
15 IONX, IONY, IONL: ARRAY(1..MAXION.) OF INTEGER;
16 X0, Y0, PX, PY, VX, VY: REAL;
17
18 PROCEDURE CLS; BEGIN WRITE(CHR(12)) END;
19
20
21 PROCEDURE LINE(X0, Y0, X1, Y1, Z: INTEGER);
22 (*Vektor von (X0, Y0) nach (X1, Y1), Z wie in PLOT(X, Y, Z)*)
23 VAR I, DX, DY, D, AX, AY, BX, BY: INTEGER;
24 BEGIN
25 DX:=X1-X0; DY:=Y1-Y0; BX:=0; AY:=0;
26 IF DX<0 THEN BEGIN AX:=-1; DX:=-DX END ELSE AX:=1;
27 IF DY<0 THEN BEGIN BY:=-1; DY:=-DY END ELSE BY:=1;
28 IF DX<DY THEN BEGIN
29 I:=DX; DX:=DY; DY:=I; BX:=AX; AX:=0; AY:=BY; BY:=0;
30 END;
31 I:=0; D:=DX SHIFT -1 (*=DIV 2*);
32 REPEAT
33 PLOT(X0, Y0, Z);
34 X0:=X0+AX; Y0:=Y0+AY; D:=D+DY; I:=I+1;
35 IF D>DX THEN BEGIN
36 D:=D-DX; X0:=X0+BX; Y0:=Y0+BY;
37 END;
38 UNTIL I=D;
39 END; (*LINE*)
40
41
42 PROCEDURE PLT ION(X, Y, L: INTEGER);
43 VAR I: INTEGER;
44 BEGIN
45 LINE(X-1, Y, X+1, Y, 1);
46 IF L<0 THEN LINE(X, Y-1, X, Y+1, 1);
47 END;
48
```

```
49
50 PROCEDURE EINGABE;
51 VAR C: STRING(1.); I, ALFA: INTEGER; OK: BOOLEAN;
52 BEGIN
53 ALFA:=45; OK:=FALSE;
54 REPEAT
55 X0:=30*COS(ALFA*PI/180); Y0:=20*SIN(ALFA*PI/180);
56 LINE(XMIN, YMIN, ROUND(X0), ROUND(Y0), 2);
57 READ(C); I:=ORD(C);
58 CASE I OF
59 18,19: IF ALFA<2 THEN ALFA:=ALFA-2;
60 17,20: IF ALFA<88 THEN ALFA:=ALFA+2;
61 OTHERS: OK:=TRUE;
62 END;
63 LINE(XMIN, YMIN, ROUND(X0), ROUND(Y0), 2);
64 UNTIL OK;
65 END;
66
67
68 PROCEDURE FELD;
69 VAR X, Y, D, I, J: INTEGER;
70 OK: BOOLEAN;
71 BEGIN
72 CLS; SCREEN(18,16); WRITE('*** GELADEN *** ');
73 LINE(0, YMIN, XMAX, YMIN, 1); LINE(0, YMAX, XMAX, YMAX, 1);
74 LINE(0, YMIN, 0, YMAX, 1); LINE(XMAX, YMIN, XMAX, YMAX, 1);
75 END;
76
77
78 PROCEDURE IONEN;
79 VAR X, Y, D, I, J: INTEGER;
80 OK: BOOLEAN;
81 BEGIN
82 FOR I:=1 TO MAXION DO BEGIN
83 REPEAT
84 OK:=TRUE;
85 X:=XMIN+10+RANDOM(XMAX-XMIN-10-2);
86 Y:=YMIN+5+RANDOM(YMAX-YMIN-5-2);
87 FOR J:=2 TO I-1 DO BEGIN
88 D:=SQR(X-IONX(.J.))+SQR(Y-IONY(.J.));
89 IF D<SQR(8) THEN OK:=FALSE;
90 END;
91 UNTIL OK;
92 IONX(.I.):=X; IONY(.I.):=Y;
93 IF ODD(I) THEN IONL(.I.):=1 ELSE IONL(.I.):=-1;
94 PLT_ION(IONX(.I.), IONY(.I.), IONL(.I.));
95 END;
96 END;
97
98
99 PROCEDURE BEWEGUNG;
100 VAR DX, DY, R: REAL; PXA, PYA, I: INTEGER;
101 BEGIN
102 R:=SQRT(SQR(X0)+SQR(Y0))/VI; IF R=0 THEN R:=1;
103 VX:=X0/R; VY:=Y0/R;
104 PX:=XMIN+1; PY:=YMIN+1;
105 REPEAT
106 PXA:=ROUND(PX); PYA:=ROUND(PY);
107 FOR I:=1 TO MAXION DO BEGIN
108 DX:=IONX(.I.)-PX; DY:=IONY(.I.)-PY;
109 R:=CG*IONL(.I.)/(SQR(DX)+SQR(DY));
110 VX:=VX+DX*R; VY:=VY+DY*R;
111 END;
112 PX:=PX+VX; PY:=PY+VY;
113 LINE(PXA, PYA, ROUND(PX), ROUND(PY), 1);
114 UNTIL (PX<XMIN) OR (PX>XMAX) OR (PY<YMIN) OR (PY>YMAX);
115 END;
116
117
118 BEGIN (*DAUPT*)
119 REPEAT
120 FELD;
121 IONEN;
122 EINGABE;
123 BEWEGUNG;
124 SCREEN(12,16);
125 I:=ROUND((PX-XMIN)*(PY-YMIN)/100.0);
126 WRITE(I, ' Punkte. Weitermachen? ');
127 READ(C); WRITE(C);
128 UNTIL ((C='N') OR (C='n'));
129 END.
```

# CLD-DOS

Wie Sie aus der folgenden Liste ersehen können, existiert im Kreise der CLD-DOS Anwender eine ganze Menge an Software. Die Beteiligten der Aktion sind bereit, Ihre Programme auch anderweitig zur Verfügung zu stellen, und so wird für die Nachbauer der Floppy-Karte bald die Frage auftauchen, wie Sie denn dieses Riesen-Software-Paket anwenden können.

Zwei Dinge sind zu erreichen:

1. Die Software muß auf unser Floppy-Format umkopiert werden (mc)
2. Das CLD-DOS Betriebssystem muß in unserem Format verfügbar gemacht werden.

Punkt 1 läßt sich gut erledigen, denn es gibt Anwender, die beide Formate (LAMPSON-Controller hardsektoriert und mc-Format) lesen/schreiben können.

Was das Betriebssystem angeht, so hat sich Herr Lampson bereit erklärt, es zu einem akzeptablen Preis in unserem Format anzubieten, soweit genügend Interessenten vorhanden sind.

Melden Sie sich doch, wenn auch Sie Zugriff auf unten stehende Programme bekommen wollen. Wir werden weiterhin mit LAMPSON in Kontakt bleiben und teilen es Ihnen mit, wenn wir zu einem positiven Ergebnis gekommen sind.

(Im Augenblick liegt die Preisvorstellung bei DM 140,-)

## VERKAUFE:

Fernschreiber T 100 S, schallgedämmt  
TTY-Schnittstelle eingebaut  
mit Lochstreifenstanzer- und Leser  
Fernschaltgerät

abzuholen bei:

**DM 130,-**

Bernd Schuhmacher

Tel. [REDACTED]

\*\*\*\*\*  
\* 80-BUS JOURNAL DISKETTEN TAUSCH SERVICE \*  
\*\*\*\*\*

Beteiligung an der Tauschaktion:  
Bitte eine (möglichst mit vielen Programmen bespielte) Diskette einsenden und 5,-DM (Schein oder Briefmarken) fuer Verpackung, Porto usw. beilegen.

Wolfgang Mayer-Guerr

[REDACTED] Recklinghausen.

Tel. [REDACTED]

Der Einsender eines Programms muss nicht immer auch der Urheber sein!

Stand: 16-FEB-83

Nr.	Filename	Sektoren	Kurzbeschreibung	eingesal
.....				
Hilfsprogramme fuer Maschinensprache				
A001	CNVHD	.ACM	8 Konvertiert HEX in DEZ	( GB )
A002	CNVBA	.ACM	3 " Bin in ASCII	( GB )
A003	OUTWRD	.ACM	3 Word an das Terminal	( GB )
A004	OUTBYT	.ACM	2 Gibt Byte aus	( GB )
A005	OUTHEX	.ACM	3 Hex an Terminal	( GB )
A006	CNVAB	.ACM	3 ASCII zu Hex	( GB )
A007	INSTR	.ACM	5 Einlesen eines Strings	( GB )
A008	INHEX	.ACM	5 Einlesen einer HexZahl	( GB )
A009	RDFILE	.ASM	9 Einlesen eines Files	( GB )
A010	WRFILE	.ASM	10 Abspeichern eines Files	( GB )
A011	TD	.ASM	9 Undefinieren von Tasten	( HE )
A012	UHR	.ASM	10 Interrupt-Uhr	( GE )
A013	START	.ACM	2 Obergrenze Workspace	( UK )
A014	COMMAND	.ACM	3 Befehlschleife	( UK )
A015	NOBLANK	.ACM	1 ignoriert Blanks	( UK )
A016	INCHAR	.ACM	2 Zeichen vom Terminal	( UK )
A017	INLINE	.ACM	3 Zeile vom Terminal	( UK )
A018	HEXDUT	.ACM	2 Ausgabe von Hexzahlen	( UK )
A019	STRBUF	.ACM	14 Stringbuffer	( UK )
A020	HEXNUM	.ACM	4 String in Zahl wandeln	( UK )
A021	HEXIN	.ACM	2 Hexzahl lesen	( UK )
A022	DBUG	.ASM	71 DBUG disassembliert	( JL )
A023	SYSCMD	.ASM	59 SYSCMD "	( JL )
A024	TOENE	.ACM	2 Toene fuer Spiele	( WMS )

## Programme zur Diskettenverwaltung

B001	CHVOL	.ABS	10 Aendern der Diskettennummer	( DL )
B002	CLRFLA	.ABS	5 Loescht alle (auch L) Flags	( DL )
B003	DIAGNOSE	.ABS	3 Erweiterter STATUS-Befehl	( MR )
B004	FORMAT	.ABS	6 Rekonstruiert Formatspur	( DL )
B005	CHL	.ABS	2 Diskettenlabel aendern	( DL )

## Hilfsprogramme in Maschinensprache

C001	REVAS	.ABS	13 NAS-DIS (nur zum Laden)	( WMS )
C002	DASM	.ABS	14 8080 Disassembler	( GB )
C003	TP	.ASM	15 Device driver f. Lochstreifenleser	( GE )
C004	FC	.ABS	7 Vergleicht 2 Files	( MR )
C005	DISASM	.ABS	16 Z80 RAM Disassembler	( MR )
C006	SORT	.ABS	4 Sortierprogramm	( MR )
C007	SUBMIT	.ABS	1 File ersetzt fast. Eingabe	( MR )
C008	PICPATCH	.ABS	4 DEBUG fuer PIC Files	( MR )
C009	DISAS	.ABS	15 Z-80 Disassembler	( UK )
C010	MINIDIS	.ABS	15 komp. Z80 Disk-Disassembler	( MR )

Mathematik/Naturwissenschaft CLD-BASIC

F001 TITRATIO.BAS 20 Titration Saecure-Base ( WMG )  
 F002 KURDIS .BAS 13 Kurvendiskussion ( HE )  
 F003 KINETIK .BAS 20 Reaktionskinetik ( WMG )  
 F004 CODON .BAS 18 Aminosaeurensynthese ( WMG )  
 F005 PRINZ .BAS 1 Primzahlen ( GR )  
 F006 FAKUL .BAS 2 berechnt. Fakultaet ( JL )

.....  
 Spiele in CLD-BASIC

G001 WUMFUS .BAS 32 Jagd in 20 Hoehlen ( WMG )  
 G002 WUMFUS2 .BAS 28 Mehrere Hoehleensysteme ( WMG )  
 G003 CSR .BAS 8 Spektroskopie ( WMG )  
 G004 ELIZA .BAS 29 Psychoanalyse ( WMG )  
 G005 OREGON .BAS 72 Treck nach Oregon ( WMG )  
 G006 MONDLAN .BAS 13 Mondlandung ( WMG )  
 G007 RODED .BAS 12 Cowboy faengt Pferd ( WMG )  
 G008 MILLI .BAS 34 Wie werde ich Millionaer ( WMG )  
 G009 MILGEMIN.DAT 1 Bester Spieler von 5008 ( WMG )  
 G010 INTERCEP.BAS 14 Rette dich vor Robotern ( WMG )  
 G011 TIC-TAC .BAS 6 Minimuehle ( WMG )  
 G012 WAHL .BAS 57 Wahl des US-Praesidenten ( WMG )  
 G013 KRAFTWER.BAS 43 Energieversorgungsbetrieb ( WMG )  
 G014 CLEMSO .BAS 28 Detektivspiel ( WMG )  
 G015 SCHIFFE .BAS 43 Schiffe versenken ( WMG )  
 G016 WALD .BAS 63 Such den Schatz ( WMG )  
 G017 WALD .DAT 1 Bester Spieler von 5016 ( WMG )  
 G018 KKW .BAS 52 Kernkraftwerk ( WMG )  
 G019 KKW .DAT 1 Bester Spieler von 5018 ( WMG )  
 G020 ROERSE .BAS 19 Spekulieren mit Aktien ( WMG )  
 G021 HAMURABI.BAS 16 Regiere eine Stadt ( WMG )  
 G022 SUPERMAN.BAS 36 Fang den Boesewicht ( WMG )  
 G023 LIMONADE.BAS 34 Verdienne viel ( UK )  
 G024 STARTREK.BAS 34 Enterprize jagt Klingons ( GR )  
 G025 MONSTER .BAS 26 Monsterjagd (aehn1. G016) ( GR )  
 G026 MATCHB .BAS 6 Nim Spiel ( GR )  
 G027 SENSO .BAS 5 Senso ( GR )  
 G028 ELIZAZ .BAS 37 ELIZA deutsch ( HP )  
 G029 HANOI .BAS 14 Tuermme von Hanoi ( HS )  
 G030 KALAH .BAS 29 Kugelspiel ( HS )  
 G031 FLIP .BAS 8 Wahrscheinlichkeit ( HS )  
 G032 MASTERMI .BAS 16 Mastermind ( HS )  
 G033 MASTERI2 .BAS 62 Enterprize jagt Klingons ( HS )  
 G034 GOLF .BAS 21 Golfspiel ( HS )

.....  
 Spiele in MBASIC

H001 ELIZA .BAS 25 Psychoanalyse ( WMG )  
 H002 MONDLAN .BAS 11 Mondlandung ( WMG )  
 H003 RODED .BAS 8 Cowboy faengt Pferd ( WMG )

C011 FROLOGUE.ABS 3 Systemstart ( MR )  
 C012 SUPERED .ABS 7 Ueberraschung ( MR )  
 C013 BACKUP .ABS 8 Disk-Totalkopie mit Verify ( MR )  
 C014 LP .DVD 6 fuer OKI 80 ( MR )  
 C015 FT .ASM 16 Programmierbare Tasten ( HE )  
 C016 BACKUPL .ABS 10 Umgestelltes Backup ( HE )  
 C017 SEDITN .ABS 14 Umgestellte. Screen Edit ( HE )  
 C018 AUF .ASM 15 Beschriften v. Etiketten ( HP )  
 C019 DCF77 .ASM 11 Uhr ( JL )  
 C020 EDITOR .ABS 21 Zeilenorientierter Editor ( JL )  
 C021 TEDIT .ABS 23 " (aus MC) ( JL )

.....  
 Spiele in Maschinensprache

D001 KEKS .ABS 2 Zweidimensional, NIM aehnlich ( WMG )  
 D002 RATE1 .ABS 2 Zahlenraten ( WMG )  
 D003 RATE2 .ABS 3 Zahlenraten ( WMG )  
 D004 CRAFTS .ABS 4 Wuerfelspiel ( WMG )  
 D005 BLACK .ABS 8 BlackJack Kartenspiel ( WMG )  
 D006 DREH .ABS 2 Zahlendreh ( WMG )  
 D007 HURKEL .ABS 3 Ratespiel ( WMG )  
 D008 UFO .ABS 2 UFOjagd ( GE )  
 D009 STAR .ABS 17 Enterprize jagt Klingons ( WMG )  
 D010 SUPER .ABS 3 Mastermind gegen Computer ( WMG )  
 D011 KLINGON .ABS 9 Klingons fangen ( WMG )  
 D012 KONZENTR .ABS 7 Gedaechnistraining ( WMG )  
 D013 REVERSI .ABS 92 verbess. Othello (Fortran) ( MR )  
 D014 SPACE1 .ABS 13 Invaders m.Graph/Ton/Beweg. ( HE )

.....  
 Hilfsprogramme fuer Microsoftbasic

E001 SHAKER .BAS 3 Sortieralgorithmus ( WMG )  
 E002 HAURUCK .BAS 3 " ( WMG )  
 E003 BUBBLE .BAS 2 " ( WMG )  
 E004 INSERT .BAS 3 " ( WMG )  
 E005 SHELL .BAS 3 " ( WMG )  
 E006 MEAP .BAS 3 " ( WMG )  
 E007 QUICK .BAS 4 " ( WMG )  
 E008 UPSIKENE .BAS 2 Unterprogramm Sirene ( WMG )  
 E009 RELOC .BAS 9 Relociertes Maschinensprache ( HE )  
 (programm mit Tastendefinition (erforderl. E010))

E010 EXT .ABS 1 ( HE )  
 E011 RELOCA .BAS 9 wie E010, laedt automatisch ( HE )  
 E012 DATABANK .BAS 4 Verwalten einer Datenbank ( WMG )  
 E013 RANDAUSG .BAS 4 Druckausgabe m.Randausgleich ( MR )  
 E014 WEEKDAY .BAS 7 berechnet Wochentag ( MR )  
 E015 SCREEN .BAS 2 Adressierbarer Cursor ( WMG )  
 E016 PACKER .BAS 43 Komprimiert Basicprogramme ( MR )  
 E017 STAT .BAS 10 Fehlerrechnung ( SH )



1019 HEDBG .DOC 34 Vorankündigung DBUS ( HE )

.....

Hilfsprogramme fuer CLD-BASIC

J001 SET .BAS 5 Graphik SET,RESET,POINT ( BE )  
 J002 HEXLIST .BAS 4 Hexausdruck eines Files ( UK )  
 J003 DATEI .BAS 15 Anlegen einer Datei ( UK )  
 J004 SINGLE .BAS 2 Einzeleingabe in File ( UK )  
 J005 KBD .BAS 1 Umwandlung in Hex ( UK )  
 J006 UMCODE .BAS 5 Umcodieren ( UK )  
 J007 ADR .BAS 23 Adressenkartei ( GR )  
 J008 KALEND .BAS 9 Datum-Berechnung ( GR )  
 J009 RENUMBER .BAS 9 Renumber v.Zeilennummern ( GR )  
 J019 DIGITAL .BAS 20 Digitalschaltungen ( JL )  
 J020 TIEFF .BAS 6 Tiefpass ( JL )  
 J021 0THKEN .BAS 5 0TH-Kenner ( JL )

.....

Programmiersprachen

K001 FORTH .ASM 43 nach G.Kreidl/Journal ( WMS )

Information ueber PASCAL, 'C', LISP, figFORTH bei ( WMS )

.....

Programme fuer Gemini IVC - Board (80x25 Zeichen)

L001 IVC .ABS 3 Bildschirmroutine ( WMS )  
 L002 H19 .ABS 3 simuliert Heath-Terminal ( WMS )  
 L003 ED .ABS 15 Bildschirmeditor + 2. Tast. ( WMS )  
 L004 SARGON .ABS 27 Schach ( WMS )  
 L005 REVERSI .ABS Reversi m. Graphik (Pascal) ( MR )

.....

Zur Bibliothek haben beigetragen

G.Baier GB  
 H.Emmelmann HE  
 B.Endert GE  
 B.Huber BH  
 U.Kafka UK  
 J.Lachmann JL  
 C.Lampson CL  
 W.Mayer-Guerr WMG  
 H.Poppinga HP  
 M.Reimer MR  
 G.Reinehr GR  
 H.Schlemmer HS

H004 MILLI .BAS 27 Wie werde ich Millionaer ( WMS )  
 H005 MILSEWIN.DAT 1 Bester Spieler von H004 ( WMS )  
 H006 INTERCEP.BAS 12 Entkomme den Robotern ( WMS )  
 H007 TICTAC .BAS 4 Minimehle ( WMS )  
 H008 CSR .BAS 6 Computerspektroskopie ( WMS )  
 H009 VIERER .BAS 10 Vier gewinnt ( WMS )  
 H010 ZONEX .BAS 9 Vierfarbentfeld erraten ( WMS )  
 H011 HIRN .BAS 8 Hirnzwirn ( GB )  
 H012 LANDER .BAS 8 Landung auch auf Planeten ( WMS )  
 H013 WUMPUS .BAS 24 Jag den Wumpus ( WMS )  
 H014 WUMFUS2 .BAS 21 Noch mehr Hoehlen ( WMS )  
 H015 AWARI .BAS 7 Mini-Kalah ( WMS )  
 H016 KRAFTWER .BAS 36 Energiebetrieb ( WMS )  
 H017 SCHIFF .BAS 33 Schiffe versenken ( WMS )  
 H018 KKW .BAS 42 Kernkraftwerk ( WMS )  
 H019 KKW .DAT 1 Bester Spieler von H018 ( WMS )  
 H020 LIMINADE .BAS 26 Verdiane viel ( WMS )  
 H021 TRUCK .BAS 56 Fernfahrer in den USA ( WMS )  
 H022 TRUCK .DAT 1 Bester Spieler von H021 ( WMS )  
 H023 WALD .BAS 51 Such den Schatz ( WMS )  
 H024 WALD .DAT 1 Bester Spieler von H023 ( WMS )  
 H025 BORNES .BAS 43 Kartenspiel Mille Bornes ( WMS )  
 H026 BUNNY .BAS 5 Druckt Hasen ( GB )  
 H027 KURVE .BAS 2 " Kurve ( GB )  
 H028 LOVE .BAS 1 " " ( GB )  
 H029 BANNER .BAS 10 " " ( GB )  
 H030 ZEIT .BAS 2 Zeitzeichen ( GB )  
 H031 WAHL .BAS 47 Wahl des US-Prasidenten ( WMS )  
 H032 VERKEHR .BAS 90 Verkehrsminister ( WMS )  
 H033 FEUER .BAS 10 Waldbrand bekampfen ( WMS )  
 H034 FLUCHT .BAS 5 Weich dem Monster aus ( WMS )  
 H035 INSEL .BAS 9 Schatzsuche auf einer Insel ( WMS )  
 H036 DUEST .BAS 32 Schatzsuche/M.Bach ( WMS )

.....

Anweisungen zu Programmen

I001 STAK .DOC 13 Fuer D009 ( WMS )  
 I002 BORNES .DOC 11 Fuer H025 ( WMS )  
 I003 WAHL .DOC 5 Fuer G012 und H031 ( WMS )  
 I004 SORT .DOC 12 Fuer C006 ( MR )  
 I005 FC .DOC 8 Fuer C004 ( MR )  
 I006 DISASM .DOC 14 Fuer C005 ( MR )  
 I007 P1CPATCH.DOC 8 Fuer C003 ( MR )  
 I008 SUEMIT .DOC 9 Fuer C007 ( MR )  
 I009 DIAGNOSE.DOC 5 Fuer B003 ( MR )  
 I010 RANDAUSE.DOC 3 Fuer E015 ( MR )  
 I011 LEVDV .DOC 10 Fuer C014 ( MR )  
 I012 SUPERED .DOC 5 Fuer C012 ( MR )  
 I013 MINIDIS .DOC 28 Fuer C010 ( MR )  
 I014 PROLOGUE.DOC 16 Fuer C011 ( MR )  
 I016 BACKUP .DOC 6 Fuer C013 ( MR )  
 I017 SUPERDIS.DOC 13 Vorankündigung Disassemb. ( MR )  
 I018 BACKUPL .DOC 5 Fuer C016 ( HE )

PROHILLE.BAS	7	CP	Zahlenspiel	CLD-Basic
AUTOMAT .BAS	7	CP	Satzformulierer	"
STARTREK.BAS	81	CP	Weltraumspiel	"
STARINST.BAS	27	CP	Instruktionen zum Spiel	"
ACEYDUCE.BAS	9	CP	Kartenspiel	"
BAND .BAS	8	CP	Umrechnung Bandzaehler - Zeit	"
BAND .DOC	2	CP	Erlaeuterungen hierzu	
CURSOR .BAS	2	MG	Adressierbarer Cursor	
SPOOLER .ASM	30	MG	Drucken im Hintergrund	
QUITSP .ASM	3	MG	Abbruch des Spoolers	
FLASS .ASM	16	MG	Aendern von Flags (erfordert Disk Nr. 1)	
CPM .BAS	24	SR	Critical Path Method	
QCON .ASM	9	DU	ZEAP in ASEM Files	
ZEIA .ASM	4	DU	Austausch von Hex-Zeichen im RAM	
BUCH1 .BAS	94	UT	Buchhaltung	
BUCH4 .DOC	3	UT	"	
BILD64 .BAS	3	UT	Bildroutine fuer 64x16 Zeichen	
KATALOG .BAS	19	MG	Katalog aller Diskfiles	
KATALOG .ASM	3	MG	Unterprogramm fuer KATALOG.BAS	

\*\*\*\*\*

Ab hier geht's nicht mehr umsonst!

Herr  
 M.Reimer  
 \_\_\_\_\_  
 \_\_\_\_\_ Berlin \_\_\_\_\_ Tel. \_\_\_\_\_

bietet eine Hardware-Uhr mit entsprechender Software an.  
 Die Uhr ist Akku-gepuffert, wird wie ein RAM angesprochen. Anzeige: Wochentag, Datum, Monat, Stunde, Minute, Sekunde, Zehntelsekunde. Schaltjahre werden beruecksichtigt.

*Bei Erklärung, warum eine beschädigte Kassetten-Tasche von mir verbleibt wurde, stand darauf: "Hat der kleine Krigebissen." Einige Zeit später bekam ich die gleiche Tasche wieder zurückgeschickt und der Bauer bemerkt: "Hat es auch geschmeckt. Was schmeckt ihm daran?"*

# PASCAL-Funktionen

von JÜRGEN WEIERMANN

```

FUNCTION ARCSIN(X:REAL):REAL;
(*Arcussinus in Grad*)
(*Function ATAN wird benoetigt*)

BEGIN
  IF X=0.0 THEN ARCSIN:=0.0
  ELSE
    IF X=1.0 THEN ARCSIN :=90.0
    ELSE
      IF X=-1.0 THEN ARCSIN:=-90.0
      ELSE ARCSIN := ATAN(1.0,X/SQRT(1.0-SQR(X)))
END;

BEGIN
  REPEAT;
    WRITE('X:');
    READLN(X);
    WRITELN('Arcsin von ',X:5:2,' ist ',ARCSIN(X):7:4)
  UNTIL X=-1.0
END.
  
```

```

PROGRAM TASIN;

VAR
  X:REAL;

FUNCTION ATAN(X,Y:REAL):REAL;
(*Arcustangens in Grad*)

CONST
  PI180=57.2957795;
VAR
  A:REAL;
BEGIN(*Function ATAN*)
  IF X=0.0 THEN
    IF Y=0.0 THEN ATAN :=0.0
    ELSE ATAN :=90.0
  ELSE (*X<>0*)
    IF Y=0.0 THEN ATAN :=0.0
    ELSE (*X and Y <>0*)
      BEGIN
        A:=ARCTAN(ABS(Y/X))*PI180;
        IF X>0.0 THEN
          IF Y>0.0 THEN ATAN :=A(*X,Y >0*)
          ELSE ATAN :=-A(*X>0,Y<0*)
        ELSE (*X<0*)
          IF Y>0.0 THEN ATAN :=180.0-A(*X<0,Y>0*)
          ELSE ATAN :=180.0+A(*X<0,Y<0*)
        END(*else*)
      END;
END;
  
```

# Overlay

von GERHARD KLEMENT

```

0010 ; == OVERLAY ==
0020 ; 9.10.82
0030 ;
0E00          0040      ORG  EF00
              0050 ;
0E00 0CDB      0060  ARGX  EQU  EDC0B
0E00 0C2B      0070  ARGX  EQU  EDC2B
0E00 1045      0080  PRIFLG EQU  E1045
0E00 105C      0090  CURLIN EQU  E105C
0E00 10C3      0100  CLRTOP EQU  E10C3
0E00 10D6      0110  BEGSIN EQU  E10D6
0E00 10DB      0120  BEGAR  EQU  E10DB
0E00 10DA      0130  BEGFRE EQU  E10DA
0E00 10EA      0140  WRA2  EQU  E10EA
0E00 10FC      0150  FSTLIN EQU  E10FC
0E00 E4DF      0160  RESET  EQU  EE4DF
0E00 EA1B      0170  CG0T0 EQU  EEA1B
0E00 F9AD      0180  HLTASC  EQU  EFA9A
              0190 ;
0E00 2A0B10    0200  SAVE   LD   HL (BEGAR)
0E00 E94B610   0210          LD   BC (BEGSIN)
0E07 E042      0220      SBC   HL BC
0E09 2280E     0230          LD   (SINLEN) HL
0E0C 2A9A10    0240      LD   HL (BEGFRE)
0E0F 23        0250      INC  HL
0E10 E042      0260      SBC   HL BC
0E12 228AE     0270      LD   (GESLEN) HL
0E15 E07392DE  0280      LD   (STACKP) SP
0E19 2A920F    0290      LD   HL (STACKP)
0E1C 11E0FF    0300      LD   DF -20
0E1F 19        0310      ADD  HL DE
0E20 228EDE    0320      LD   (RAMHI) HL
0E23 23        0330      INC  HL
0E24 E04B8A0E  0340      LD   BC (GESLEN)
0E2B 0B        0350      DEC  BC
0E29 E042      0360      SBC   HL BC
0E2B 03        0370      INC  BC
0E2C 22900E    0380      LD   (NWTAB) HL
0E2F 2A9510    0390      LD   HL (BEGSIN)
0E32 E05B900E  0400      LD   DE (NWTAB)
0E35 DF49      0410      SCAL "I
              0420 ; -----
0E38 AF        0430      XOR  A ;
0E39 320B0C    0440      LD   (ARGN) A ; :DAD OFFSLI=0
0E3C 3E52      0450      LD   A "R ;
0E3E 322B0C    0460      LD   (ARGX) A ;
0E41 DF52      0470      SCAL "R ; :LOAD PGM
              0480 ; -----
0E43 2A8EDC    0490  RETR  LD   HL (RAMHI)
0E46 22C310    0500          LD   (CLRTOP) HL
0E49 2AD610    0510      LD   HL (BEGSIN)
0E4C ED4B8C0E  0520      LD   BC (SINLEN)
0E50 ED4A      0530      ADC  HL BC
0E52 22D810    0540      LD   (BEGAR) HL
0E55 2A9610    0550      LD   HL (BEGSIN)
0E58 E04B8A0E  0560      LD   BC (GESLEN)
0E5C 0B        0570      DEC  BC
0E5D ED4A      0580      ADC  HL BC
0E5F 22DA10    0590      LD   (BEGFRE) HL
0E62 2A900E    0600      LD   HL (NWTAB)
0E65 E05BDE10  0610      LD   DE (BEGSIN)
0E69 03        0620      INC  BC
0E6A DF49      0630      SCAL "I
              0760 ; -----
0E6C 21FFFF    0770          LD   HL EFFFF ;SET PROGRAM
0E6F 225C10    0780          LD   (CURLIN) HL; PARAMETERS
0E72 C09FE4    0790      CALL RESET ;
0E75 2AFC10    0800      LD   HL (FSTLIN);POI TO 1st LINE
0E78 3E01      0810      LD   A 1 ;DISABLE OUTPUT
0E7A 324510    0820      LD   (PRIFLG) A ;
0E7D CDAD19    0830      CALL HLTASC ;PRI LINE TO WRA2
0E80 21E410    0840      LD   HL WRA2 ;POINT TO ACC
0E83 AF        0850      XOR  A ;
0E84 324510    0860      LD   (PRIFLG) A ;CLR PRIFLG

```

```

0E87 C316EA    0870      JP   CG0T0 ;HERE WE GO
              0880 ;
0E8A 0000      0890  GESLEN DEFW 0
0E8C 0000      0900  SINLEN DEFW 0
0E8E 0000      0910  RAMHI  DEFW 0
0E90 0000      0920  NWTAB  DEFW 0
0E92 0000      0930  STACKP DEFW 0

```

```

10 REM == OVERLAY DEMO 1 ==
20 :
30 DOK(4100,3584
40 REM -- Definition der Strings --
50 A=4
60 S(5,5)=22
70 A$(2,2)="STRING 1"
80 B$="STRING 2"
90 :
100 REM -- Unspeichern in Stringbereich --
110 A$(2,2)=A$(2,2)+""
120 B$=B$+""
130 PRINTUSR(0)
Ok

```

```

10 REM == OVERLAY DEMO 2 ==
20 :
30 REM -- Definition der Strings --
40 REM A=4
50 REM S(5,5)=22
60 REM A$(2,2)="STRING 1"
70 REM B$="STRING 2"
80 :
90 REM -- Ausdrucken --
100 PRINTA
110 PRINTS(5,5)
120 PRINTAS(2,2)
130 PRI473$
Ok

```

Overlay erlaubt das Nachladen von BASIC Programmen mit Autostart, ohne daß Variable verloren gehen. (Ähnlich wie in FORTRAN).

# Wächter

von K.TRUST/G.ASSMANN

## SPANNUNGSÜBERWACHUNG

ICs mit mehreren Spannungen werden beim Fehlen einer negativen Versorgungsspannung bekanntlich leicht zerstört. Ebenso kann das "Hochlaufen" einer Spannung verheerende Folgen haben. Mit der vorliegenden Schaltung können zwei positive und zwei negative Spannungen ueberwacht werden. Die negativen Spannungen werden durch Operationsverstaerker auf positive Werte gebracht. Als Referenz wird die stabilisierte Spannung des TCA 965 benutzt, die bei ca. 3,6 V liegt. Wenn die Versorgungsspannungen innerhalb des eingestellten Fensters liegen, sind die Ausgaenge der Fensterkomparatoren auf High-Potential (IC 1...4) und somit wird T1 durchgeschaltet; die Relais ziehen an.

Liegt die START-Leitung (Stift 31) auf Low, kippt das nicht nachtriggerbare Monoflop IC 6 fuer einige Millisekunden und schaltet ueber D3 den Transistor T1 durch. Relais 1 und 2 ziehen an und schalten die zu ueberwachenden Spannungen durch.

Liegen alle Spannungen innerhalb des eingestellten Fensters, dann bleibt ueber das UND-Gatter und D2 der T1 durchgeschaltet und die Relais halten sich. Fehlt eine Spannung oder liegt sie nicht innerhalb des Fensters, so ist der zugehoerige Komparatorausgang (Pin 3) auf Low, das UND-Gatter sperrt, die Relais fallen nach dem Zurueckkippen des Monoflops wieder ab; die Schaltung laesst sich nicht starten.

Faelit waehrend des Betriebs eine Spannung aus oder die Spannung weicht vom eingestellten Fenster ab, dann werden alle Spannungen sofort abgeschaltet.

Fuer jede zu ueberwachende Spannung ist eine Bruecke J1...J4 einzulegen.

Um beste Langzeitstabilitaet zu erlangen, wurde auf Potis zum Einstellen von Fenstermitte und -breite verzichtet und Metallfilmfestwiderstaende benutzt (R1...R16).

Die Werte der Parallelwiderstaende muessen individuell mittels eines Potis ausgemessen und sehr genau ausgesucht werden, da durch Bauteiltoleranzen nur ungefaehre Werte angegeben werden koennen (s. Zeichnung).

Es duerfte aber nicht schwer fallen, die einzelnen Groessen wie Fenstermitte und -breite einzustellen, hierzu dient Bild 3.

Fuer T1 kann jeder beliebige npn-Transistor genommen werden, der ein gutes Schaltvermoegen, sowie relativ hohe Belastbarkeit hat (Stroeme bis 200 mA).

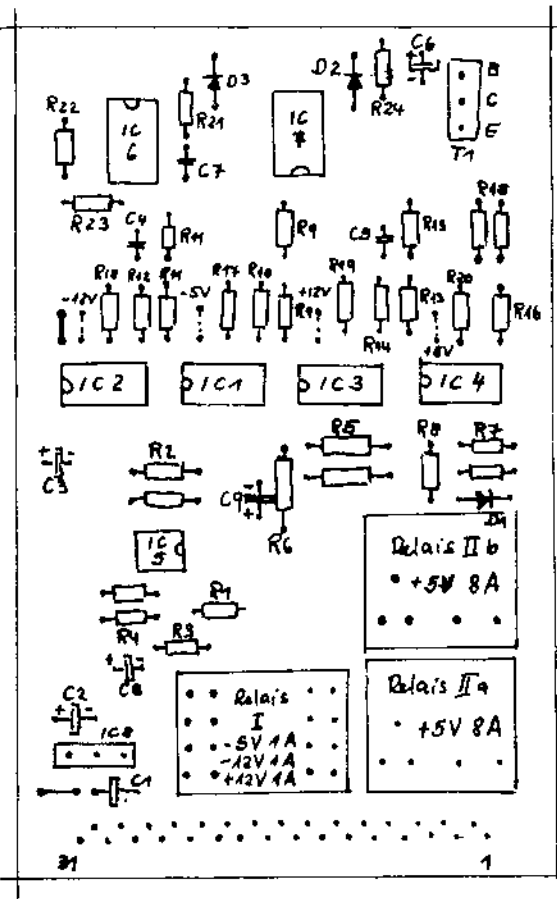
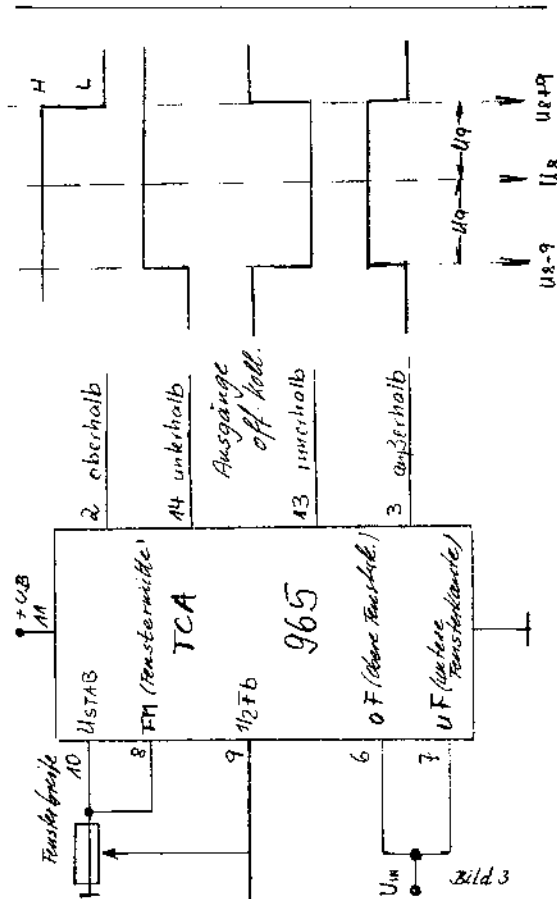
Relais 1 und 2 sind handelsuebliche Relais (Dahms-Elektronik), OMRON Typ G2L-113P, ERNI Rel. Fuer die +5 V-Spannung wurden zwei Relais vorgesehen um auch Stroeme groesser 5A sicher zu schalten.

Damit die Spannungs-Ueberwachungskarte arbeiten kann, muss sie von der +5 V-Versorgung des Systems unabhængig sein. Es ist empfehlenswert hierfuer eine separate 12 V Spannungsversorgung aufzubauen.

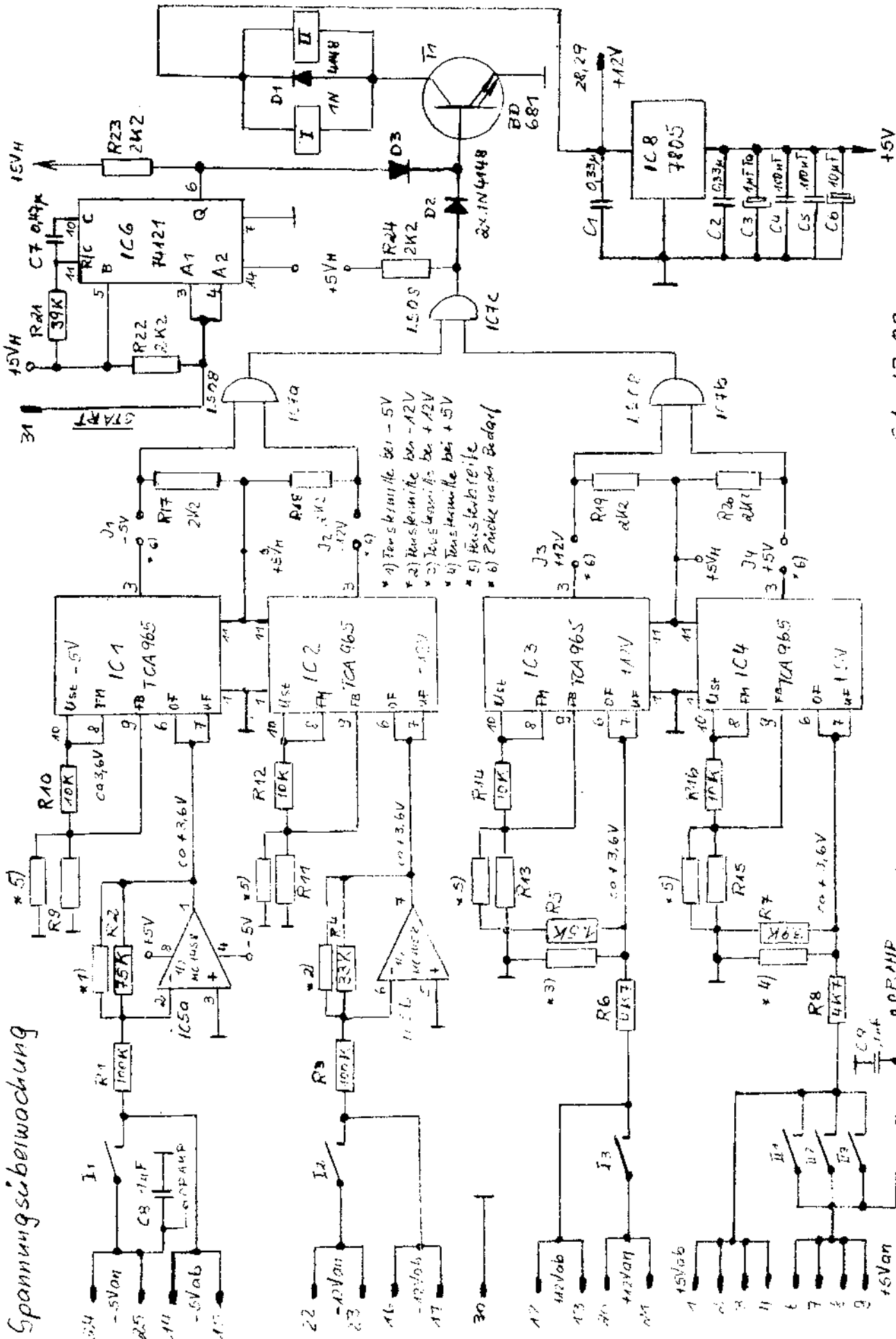
Fertig geaetzte und gebohrte Karten koennen von folgender Adresse zum selbstkostenpreis von DM 15,- bezogen werden:

Georg Assmann

Herzhausen



# Spannungsbewachung



- \* 1) Transistor mit 50V - 5V
- \* 2) Transistor mit 50V - 12V
- \* 3) Transistor mit 50V - 12V
- \* 4) Transistor mit 50V - 5V
- \* 5) Transistor mit 50V
- \* 6) Pinbelegung Bedarf

GA. 12. 82

Ausgabe 1

# Clock

von GERHARD KLEMENT

CLOCK lauert in der Input Routine und zeigt an, wie lange Jemand zur Beantwortung von Fragen braucht. Es entstand für ein Schulungsprogramm.

ZEAP Z80 Assembler - Source Listing

```

0010 ; -- CLOCK --
0020 ; 10.10.82
0030 ;*****
0040 ;* VON BASIC MIT USR AUFRUFEN *
0050 ;* RESCLK ... RESET *
0060 ;* INPLOP ... INPUT LOOP MIT COUNT *
0070 ;*****
OC80 0080 ORG EC80
0090 ; C80-D08
0100 ;
OC80 0003 0110 COUNTS EQU 3
OC80 0006 0120 DCOUNT EQU 6
OC80 0038 0130 RDEL EQU E0038
OC80 0062 0140 IN EQU E0062
OC80 0065 0150 CRT EQU E0065
OC80 0068 0160 B2HEX EQU E0068
OC80 08D9 0170 SCR1 EQU E08D9
OC80 DC29 0180 CURSR EQU E0C29
OC80 F0F7 0190 USHRE1 EQU EF0F7
0200 ;
OC80 21F40C 0210 RESCLK LD HL TEXT
OC83 11D408 0220 LD DE SCR1-5
OC86 011100 0230 LD BC 17
OC89 EDB0 0240 LDIR
OC8B AF 0250 XOR A
OC8C 0606 0260 LD B,DCOUNT
OC8E 210500 0270 LD HL,NUM
OC91 77 0280 CLR D (HL),A
OC92 23 0290 INC HL
OC93 10FC 0300 DJNZ CLR
OC95 C9 0310 RET
0320 ; == COUNTER ==
OC96 2A290C 0330 INPLOP LD HL (CURSR)
OC99 22F20C 0340 LD (CUSTD) HL
OC9C 365F 0350 LD (HL) ""
OC9E 210500 0360 LOOP LD HL,NUM
OCA1 110800 0370 LD DE,SUM
OCA4 7E 0380 LD A (HL)
OCA5 CED1 0390 ADC A,1
OCA7 27 0400 DAA
OCAB 12 0410 LD (DE) A
OCA9 0602 0420 LD B,COUNTS-1
OCAB 23 0430 LAB1 INC HL
OCAC 13 0440 INC DE
OCAD 7E 0450 LD A (HL)
OCAE CED0 0460 ADC A,0
OCB0 27 0470 DAA
OCB1 12 0480 LD (DE) A
OCB2 10F7 0490 DJNZ LAB1
0500 ; -- DISPLAY --
OCB4 21D900 0510 LD HL,SCR1
OCB7 110700 0520 LD DE,NUM+COUNTS-1
OCBA 0603 0530 LD B,COUNTS
OCBC 22290C 0540 LAB2 LD (CURSR),HL
OCB1 1A 0550 LD A,(DE)
OCC0 0F68 0560 SCAL B2HEX
OCC2 23 0570 INC HL
OCC3 23 0580 INC HL
OCC4 1B 0590 DEC DE
OCC5 10F5 0600 DJNZ LAB2
OCC7 DF62 0610 SCAL IN
OCC9 47 0620 LD B,A
OCCA 3812 0630 JR C,EXIT
OCCC 210800 0640 LD HL,SUM
OCCF 110500 0650 LD DE,NUM
OCD2 010300 0660 LD BC,COUNTS

```

```

OCDS EDB0 0670 LDIR
OC07 0603 0680 LD B,3
OC09 FF 0690 DELY RST RDEL
OCDA 10F0 0700 DJNZ DELY
OCDC 18C0 0710 JR LOOP
0720 ;
OCDE 2AF20C 0730 EXIT LD HL (CUSTD)
OC0E 22290C 0740 LD (CURSR) HL
OC04 F800 0750 CP 13
OC06 2004 0760 JR NZ,G01
OC08 3620 0770 LD (HL) ""
OC0A 1802 0780 JR G02
0790 ;
OC0C DF65 0800 G01 SCAL CRT
OC0E AF 0810 G02 XOR A
OC0F C3F2F0 0820 JP USRRET
0830 ;
OCF2 0002 0840 CUSTD DEFS 2
OCF4 54B96065 0850 TEXT DEFM /Time units/
20202020
20202020
758C6974
73
0005 0003 0860 NUM DEFS COUNTS
0008 0003 0870 SUM DEFS COUNTS

```

CLOCK DEMO

```

10 REM -- RESET CLOCK --
20 CLS
30 CLEAR 1000
40 D0KE 410,3200:U=USR(0)
50 REM -- RUN CLOCK --
60 D0KE 4100,322:A$=""
70 PRINT"INGEBEN : ";
8 U=USR(0)
90 IF U=13 THEN PRINT:GOTO110
100 A$=A$+CHR$(U):GOTO80
110 PRINT "INGEBEN = "A$
120 FOR I=1 TO 1000:NEXT:PRINT:GOTO40
NK

```

## KLEINANZEIGE

Suche FDC +Floppy für CP/M mit NC Video-  
karte 80x24 für NC

Ebenso ASCII- Selectric Interface für  
IBM Kugelkopfschreibmaschine

Rene Claus ,

## KLEINANZEIGE

Verkaufe billig: 3 Stück 2716 mit NAS-  
DIS/DEBUG, Toolkit

4 Stück 2532 mit NASDIS/DEBUG, ZEAP 2.0,  
BASIC

14 Stück 2708; orig. N1 Zeichengenerat.  
8K-BASIC (RAM-Version) auf orig. Casset-  
te mit Handbuch

Uwe Fricke ,

Tel. ,

# CONMOD II

von JÜRGEN WEIERMANN

## CONMOD Verbesserung!

Wenn ich gewusst hätte welche Bedeutung mein Programm CONMOD, bezogen auf die Entwicklung um NAS-SYS 3 bekommen würde, so hätte ich direkt etwas vernünftiger geschrieben und nicht ein halbfertiges Programm veröffentlicht lassen. Die Tatsache dass ein eingegebenes Backspace den Abbruch des Programms erfordert und auch die Fehlerkorrektur durch die INLINE-Routine erhebliche Schwierigkeiten aufweist, haben mich dazu veranlasst das ganze Programm noch einmal zu überarbeiten.

Hier nun das Ergebnis: INLINE wird ersetzt durch eine Schleife um DF 7DH aus der man nur durch Eingabe von 0DH herauskommt (Achtung: nur die aktu. Zeile kann editiert werden). Damit wäre das erste Problem beseitigt. Die Beseitigung des Backspace-Problems ist noch einfacher in den jeweiligen Eingabe-Routinen wird BS auch erkannt und ignoriert. Ich hoffe, dass diese Änderungen dazu beitragen können, dass die Eingabe von Hex Listings einfacher geworden ist.

Hier noch eine verblüffend einfache Änderung, gefunden von Herrn oligaweller. Durch die Änderung von 3 Bytes im Assembler-Listing mit einem "\*" gekennzeichnete auf die entsprechenden, im Listing aufgeführten Werte lässt das Programm auch eine Eingabe ohne Prüfsumme zu.

## ZEAP Z80 Assembler - Source Listing

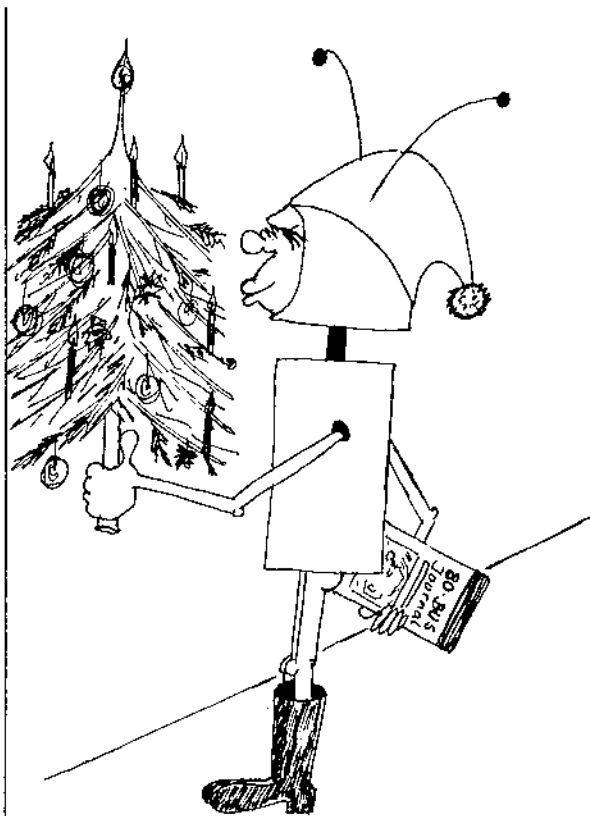
```

0010 ;Z-80 NASCOM
0020 ;CONMOD V1.1
0030 ;PROG. ZUR EINGABE EINES HEX-LIST
0040 ;MIT PRUEFSUMME
0050 ;RELOKATIBEL
0060 ;JUERGEN WEIERMANN 3/82

1000      0070      ORG #1000
1000      0080      ENT
1000 DF    0090      RST 18H
1001 60    0100      DEFB 60H
1002 D5    0110      PUSH DE
1003 E1    0120 START POP HL
1004 E5    0130      PUSH HL
1005 DF    0140      RST 18H
1006 66    0150      DEFB 66H
1007 0609  0160      LD B,09 ;* 08H
          0170 ;EINGABE I
1009 DF    0180 IN1 RST 18H
100A 7B    0190      DEFB 7BH
100B FE11  0200      CP 11H
100D 281F  0210      JR Z,EDIT1
100F FE12  0220      CP 12H
1011 281F  0230      JR Z,EDIT2
1013 FE08  0240      CP 08H
1015 28F2  0250      JR Z,IN1
1017 F7    0260      RST 30H
          0270 ;EINGABE II
1018 DF    0280 IN2 RST 18H
1019 7B    0290      DEFB 7BH
101A FE11  0300      CP 11H
101C 281B  0310      JR Z,EDIT4
101E FE12  0320      CP 12H
1020 2813  0330      JR Z,EDIT3
1022 FE08  0340      CP 08H
1024 28F2  0350      JR Z,IN2

1026 F7    0360      RST 30H
1027 DF    0370      RST 18H
1028 69    0380      DEFB 69H
1029 05    0390      DEC B
102A 20DD  0400      JR NZ IN1
102C 1810  0410      JR NEW
          0420 ;EDIT FUNKTIONEN
102E F7    0430 EDIT1 RST 30H
102F 04    0440      INC B
1030 3E11  0450      LD A,11H
1032 F7    0460 EDIT2 RST 30H
1033 18E3  0470      JR IN2
1035 F7    0480 EDIT3 RST 30H
1036 05    0490      DEC B
1037 3E12  0500      LD A,12H
1039 F7    0510 EDIT4 RST 30H
103A 18CD  0520      JR IN1
          0530 ;
103C 18C5  0540 RELO1 JR START
          0550 ;
          0560 NEW LD HL,(0C29H)
1041 DF    0570      RST 18H
1042 7C    0580      DEFB 7CH
1043 E5    0590      PUSH HL
1044 D1    0600      POP DE
1045 DF    0610      RST 18H
1046 64    0620      DEFB 64H
1047 2A210C 0630      LD HL,(0C21H)
104A 3E00  0640      LD A,00
104C 0609  0650      LD B,09H ;* 08H
104E 85    0660      ADD A,L
104F 84    0670      ADD A,H
1050 F5    0680 NEXT PUSH AF
1051 DF    0690      RST 18H
1052 64    0700      DEFB 64H
1053 2A210C 0710      LD HL,(0C21H)
1056 F1    0720      POP AF
1057 05    0730      DEC B
1058 2803  0740      JR Z,TEST
105A 85    0750      ADD A,L
105B 18F3  0760      JR NEXT
105D BD    0770 TEST CP L
105E 2817  0780      JR Z,OK ;* JR,OK
1060 EF    0790      RST 28H
1061 45    0800      DEFB 45H
1062 00    0810      DEFB 00H
1063 2A290C 0820      LD HL,(0C29H)
1066 E5    0830      PUSH HL
1067 DF    0840 GO RST 18H
1068 7B    0850      DEFB 7BH
1069 FE0D  0860      CP 0DH
106B 2803  0870      JR Z,CBACK
106D F7    0880      RST 30H
106E 18F7  0890      JR GO
1070 E1    0900 CRACK POP HL
1071 2B    0910      DEC HL
1072 22290C 0920      LD HL,(0C29H),HL
1075 18C7  0930      JR NEW
1077 2A290C 0940 OK LD HL,(0C29H)
107A DF    0950      RST 18H
107B 7C    0960      DEFB 7CH
107C E5    0970      PUSH HL
107D D1    0980      POP DE
107E 13    0990      INC DE
107F 13    1000      INC DE
1080 13    1010      INC DE
1081 13    1020      INC DE
1082 0608  1030      LD B,08
1084 DF    1040 MORE RST 18H
1085 64    1050      DEFB 64H
1086 2A210C 1060      LD HL,(0C21H)
1089 7D    1070      LD A,L
108A E1    1080      POP HL
108B 77    1090      LD HL,(HL),L
108C 23    1100      INC HL
108D E5    1110      PUSH HL
108E 05    1120      DEC B
108F 20F3  1130      JR NZ,MORE
1091 DF    1140      RST 18H
1092 64    1150      DEFB 64H
1093 18A7  1160      JR NEXT

```



Hallo liebe Leser,  
Es ist mal wieder so weit (diesen Satz kennen Sie schon): Weihnachten steht vor der Tür, und Sie haben noch nicht alle Geschenke eingekauft.

Hat man Computer- Freaks in der Verwandt- oder Bekanntschaft, so ist das kein Problem: man bestellt schnell noch ein Abo 83 des 80-Bus Journals bei der Gabi Böhm oder gar einen Geschenkgutschein für das brandneue Abo 84. Was aber, wenn der Beschenkte in spe nichts mit NASCOM oder ECB im Sinne hat? NASCOMPL gibt Rat: auch hier läßt sich ein Abonnement dieses vielseitigen Journals als ausgezeichnete Geschenkidee verwenden.

Ich will Ihnen nur ein paar einfache Beispiele nennen, wie jedermann unser Journal zu seiner und seiner Umwelt Freude einsetzen kann.

1. Malbuch für Kinder (die kleinen freuen sich besonders über den NASCOMPL)
2. Als Maßeinheit besonders vielseitig: Ein Jahresabo ist exakt 20 mm dick, die Länge beträgt 29,4 cm. Wenn Sie drei Hefte aneinanderlegen... Eine genaue Gebrauchsanleitung zum Messen und Wiegen kann mitgeliefert werden.

3. Beim Verlegen von Teppichböden eignet sich das Journal hervorragend zum Ausgleichen von Unebenheiten. Die Juli/August-Ausgabe sollte bei besonders groben Löchern mehrfach bestellt werden.

4. Als Schlafmittel eignet sich das mehrfache Lesen des Editorials. Das Vorhandensein von 10 verschiedenen Heften kann dabei einer Abhängigkeit von einer bestimmten Ausgabe entgegenwirken.

Wenn Ihnen an weiteren interessanten Anwendungen des Journals gelegen ist, bestellen Sie doch einfach unser Sonderheft 13 mit vielen Beschreibungen, Fotos, Plänen und Einbauanleitungen. Dieses Nachschlagewerk ist auch in Englisch, Dänisch und Suaheli erhältlich. Ich freue mich auf Ihre Bestellung.

In diesem Sinne Ihr NASCOMPL

## 2 BASIC Spiele

von EBERHARD HORCH

```

10 REM * -----
20 REM * Umgeschrieben für NASCOM nach einem
30 REM * Vorschlag von R.Baumann in CHIP Wissen
40 REM * "Computer spiele und Knobelien"
50 REM * E.Horch, Hannover
60 REM * -----
100 CLS:PRINT"          *** Eidechsen spiel ***"
110 PRINT"          -----"
120 PRINT:PRINT"          - Spielregeln - "
130 PRINT" Die Eidechse versucht die Fliege zu"
140 PRINT"erhaschen. Zum Auslösen der Zunge bei der"
150 PRINT"Eidechse nehmen Sie bitte die Space-Taste."
160 PRINT:PRINT"          --- Viel Spaß ---"
170 FOR N=1 TO 4000:NEXT
180 PRINT:PRINT:PRINT"achtung":FOR I=1 TO 300:NEXT
190 PRINT"      Fertig machen":FOR I=1 TO 300:NEXT
200 PRINT"      Start":FOR I=1 TO 300:NEXT
210 REM ----- < Fliege plazieren > -----
220 Z=0:U=0
230 U=U+1:IF U=11 THEN 650
240 R=INT(10*RAND(1))
250 X=2065+(R*64)+INT(40*RAND(1))
260 CLS:POKE X,42
270 REM ----- < Eidechse laufen > -----
280 FOR K=2690 TO 2947
290 POKEK-1,32:POKEK,30:POKEK+65,32:POKEK+64,32
300 POKEK,178:POKEK-1,35:POKEK-2,35:POKEK-3,35
310 POKEK-4,35:POKEK-5,61:POKEK-6,61:POKEK-7,61
320 POKEK-8,45:POKEK+1,210:POKEK-9,32
330 POKEK+63,60:POKEK+62,32:POKEK+60,61
340 POKEK+59,32
350 REM ----- < Gef Zeichen > -----
360 DOKE 3200,25311:DOKE 3202,-24526
370 DOKE 3204,-14068:A=3232
380 DOKE 4100,3200
390 Y=USR(1)
400 IF PEEK(A)+30 THEN 420
410 NEXT K

```



```

410 REM ----- < Fangversuch > -----
430 FOR T=K-64 TO K-896 STEP-64
440 POKE T,148
450 IF PEEK(T-64)<>42 THEN 570
460 FOR H=T TO K STEP 64
470 POKE H-64,32:POKE H,42
480 FOR I=1 TO 25:NEXT I
490 NEXT H
500 REM ----- < BEUTE VERSCHLINGEN > -----
510 POKE K-63,32:POKE K-40,32:POKE K,210
520 POKE K+1,210
530 FOR I=1 TO 50:NEXT I
540 POKE K,210:POKE K+1,46
550 GOSUB 580:GOTO 230
560 POKE T,148:POKE T,32
570 NEXT T:GOTO 230
580 REM ----- < Sub Schlucken > -----
590 PRINT:PRINT" Schmatz !!"
600 Z=Z+1
610 PRINT:PRINT Z;" . Fliege geschnappt"
620 FOR I=1 TO 300:NEXT I
630 RETURN

640 REM ----- < Spielende > -----
650 CLS:PRINT"Sie hatten 10 Versuche und haben"
660 PRINT" ";Z;"Fliegen gefangen."
670 PRINT:PRINT:PRINT
680 INPUT"Wollen Sie noch ein Spiel (J/N)";A#
690 IF A#="J" THEN 180
700 IF A#<>"J" THEN PRINT:PRINT"Danke für das Spiel"
710 END
Ok

```

```

10 CLEAR 100
20 DOKE 3200,25311:DOKE 3202,-24526
30 DOKE 3204,-14068:DOKE 3206,13007
40 DOKE 3208,3232:DOKE 3210,201
50 A=3232
60 CLS
70 SCREEN 10,1:PRINT"*** REAKTIONSTEST ***"
80 SCREEN 10,2:PRINT"-----":PRINT
90 PRINT:PRINT:PRINT"Wollen Sie einen leichten oder
100 PRINT"schwierigen Test machen? Bei leicht"
110 PRINT"druicken Sie bitte die Taste 'L' und"
120 PRINT"bei schwierig die Taste 'S'."
130 DOKE 4100,3206
140 M=USR(0)
150 IF PEEK(A)=83 THEN280
160 IF PEEK(A)=76 THEN180
170 IF PEEK(A)<>83 OR PEEK(A)<>76 THEN140
180 PRINT:PRINT"Der Computer zeichnet ein Quadrat
190 PRINT"den Schirm,dass 1-7 sec sichtbar ist."
200 PRINT"Sie sollen die LeerTaste druicken,wenn"
210 PRINT"das Quadrat verschwindet.Sie haben nun"
220 PRINT"10 Versuche.Wenn Sie bereit sind bitte"
230 PRINT"den Buchstaben 'L' eintippen."
240 DOKE 4100,3206
250 M=USR(0)
260 IF PEEK(A)=76 THEN390
270 IF PEEK(A)<>76 THEN250
280 PRINT"Bei dem schwierigen Test wird ebenfalls"
290 PRINT"ein Quadrat auf den Schirm gezeichnet."
300 PRINT"Nach dem Verschwinden erscheint auf dem"
310 PRINT"Schirm an irgendeiner Stelle jedoch ein"
320 PRINT"Buchstabe,den Sie auf der Tastatur fix"
330 PRINT"druicken muessen.Zum Start Druicken Sie"
340 PRINT"bitte die Taste 'S'."
350 DOKE 4100,3206
360 M=USR(0)
370 IF PEEK(A)=83 THEN860
380 IF PEEK(A)<>83 THEN360
390 FOR L=1 TO 15:PRINT:NEXT
400 DOKE3232,211
410 FOR Y=5 TO 9
420 FOR X=17 TO 25
430 SCREEN X,Y:PRINT"*"

```

```

440 NEXT X
450 NEXT Y
460 W=INT(RND(1)*250)
470 FOR T=1 TO W
480 DOKE4100,3200
490 D=USR(1)
500 IF PEEK(A)=32 THEN830
510 NEXT T
520 CLS
530 DOKE4100,3200
540 FOR B=1 TO 3000
550 D=USR(1)
560 IF PEEK(A)=32 THEN580
570 NEXT B
580 B=(B*15.4)-15.4:REM B=Zeit in Millisec.
590 C=C+1:E(C)=B :REM C=Nr. des Versuches
600 REM E(C)=Ergebnis des Vers.
610 FOR Z=1 TO C
620 PRINT Z;:PRINTTAB(2)" Versuch:.....";E(Z);
630 PRINTTAB(2B)"Millisec."
640 NEXT Z
650 IF Z=11 THEN710:REM Nach 10 Vers. Ende
660 PRINT"Neuer Versuch (J/N)"
670 DOKE 4100,3206
680 M=USR(0)
690 IF PEEK(A)=74 THEN390
700 D=INT(D)
710 FOR Z=1 TO C:D=D+E(Z):NEXT:REM Durchschnitt
720 D=D/(Z-1)
730 PRINT:PRINT"Ihr Durchschnitt war";D;"Millisec. !!"
740 PRINT:PRINT:PRINT"Wollen Sie es nochmal versuchen?"
750 PRINT "Bitte tippen Sie 'J' oder 'N' ein."
760 DOKE 4100,3206
770 M=USR(0)
780 IF PEEK(A)=74 THEN10
790 IF PEEK(A)<>74 THEN800
800 PRINT:PRINT"
810 PRINT"
820 END
830 PRINT:PRINT:PRINTTAB(11)"Bitte nicht mogeln !"
840 FOR D=1 TO 500:NEXT D
850 GOTO390
860 FOR L=1 TO 15:PRINT:NEXT L
870 FOR Y=3 TO 11
880 FOR X=13 TO 29
890 SCREEN X,Y:PRINT"RND"
900 NEXT X
910 NEXT Y
920 W=INT(RND(1)*250)
930 FOR T=1 TO W:NEXT T
940 CLS
950 DOKE4100,3200
960 X=1+INT(48*RND(1)):Y=1+INT(15*RND(1))
970 V=1+INT(64+26*RND(1)):POKE 3232,V
980 SCREEN X,Y:PRINT CHR$(V)
990 FOR B=1 TO 5000
1000 D=USR(1)
1010 IF PEEK(A)= V THEN1030
1020 NEXT B
1030 B=(B*15.4)-15.4:REM B=Zeit in Millisec.
1040 C=C+1:E(C)=B :REM C=Nr. des Versuches
1050 REM E(C)=Ergebnis des Vers.
1060 FOR Z=1 TO C
1070 PRINT Z;:PRINTTAB(2)" Versuch:.....";E(Z);
1080 PRINTTAB(2B)"Millisec."
1090 NEXT Z
1100 IF Z=11 THEN1170:REM Nach 10 Vers. Ende
1110 PRINT"Neuer Versuch (J/N)"
1120 DOKE 4100,3206
1130 M=USR(0)
1140 IF PEEK(A)=74 THEN860
1150 IF PEEK(A)<>74 THEN1160
1160 D=INT(D)
1170 FOR Z=1 TO C:D=D+E(Z):NEXT:REM Durchschnitt
1180 D=D/(Z-1)
1190 PRINT:PRINT"Ihr Durchschnitt war";D;"Millisec. !!"
1200 PRINT:PRINT:PRINT"Wollen Sie es nochmal versuchen ?"
1210 PRINT "Bitte tippen Sie 'J' oder 'N' ein."
1220 DOKE 4100,3206
1230 M=USR(0)
1240 IF PEEK(A)=74 THEN10

```

Vielen Dank !!

# Roulette

von KLAUS MOMBAUR

```

2 REM      *** Roulette ***
4 REM
5 REM aus "BASIC COMPUTER GAMES" v. H.Ahl
8 REM
9 REM Mit "rollender Kugel" versehen,
10 REM modifiziert und an NASCOM 2 angepasst
11 REM Klaus Mombaur Nuernberg
12 REM
13 CLS
14 PRINTTAB(14)*R O U L E T T E *
18 PRINTTAB(14)*-----*
20 INPUT"Bitte das Datum (z.B.24.4.81): ";D$
25 PRINT
30 PRINT"Bitte hinterlegen Sie nun 1000 DM Kauti-
  on
35 INPUT"Erledigt (J/N) ";E$
37 IF E$<>"J" THEN 25
40 PRINT"Ich begruesse Sie am Roulette - Tisch!
50 PRINT:INPUT"Wuenschen Sie die Spielregeln? (
  J/N)";F$
60 IF F$="N" THEN 1550
70 GOSUB100:GOTO200
100 CLS
102 PRINT"          So sieht das Spielfeld aus:
105 PRINT" |-----|-----|-----|
;
;
106 PRINT"-----|-----|
110 PRINT" | 1 4 7 10 | 13 14 19 22 | 25 28 31 *
;
;
115 PRINT" 34 | |
120 PRINT" | " " " | " " " | " " " *
;
;
125 PRINT" " " | 0 |
130 PRINT" | 2 5 8 11 | 14 17 20 23 | 26 29 32 *
;
;
135 PRINT" 35 | |
140 PRINT" | " " " | " " " | " " " *
;
;
145 PRINT" " | 00 |
150 PRINT" | 3 6 9 12 | 15 18 21 24 | 27 30 33 *
;
;
155 PRINT" 36 | |
160 PRINT" |-----|-----|-----|
;
;
165 PRINT"-----|-----|
180 RETURN
200 PRINT"Die 35:1 Wetten sind:
210 PRINT" Die Zahlen 1 bis 36 * 1-36
*
220 PRINT"Die 2:1 Wetten sind:
230 PRINT" | bis 12 * 37 * 1.Kolonne
* 40 *
240 PRINT" |3 bis 24 * 38 * 2.Kolonne
* 41 *
250 PRINT" |25 bis 36 * 39 * 3.Kolonne
* 42 *
260 INPUT" weiter";A$
270 CLS
280 GOSUB 100
300 PRINT"Die 1:1 Wetten sind:
310 PRINT" | bis 18 * 43 * Ungerade * 46
*
320 PRINT" |19 bis 36 * 44 * Rot (--) * 47
*
330 PRINT" Gerade * 45 * Schwarz * 48
*
335 FORP$=1TO47:PRINTCHR$(45);:NEXT
340 PRINT" Bei --0- * 49 * und -00- * 50
*
350 INPUT" gewinnen nur diese mit 35:1! weit-
  er";A$
360 GOSUB100

```

```

370 PRINT"Wenn ich Sie nach einer Wette frage,
372 PRINT"dann tippen Sie die Nummer und den Ei-
  nsatz
374 PRINT"getrennt durch ein Komma:
376 PRINT"Beispiel: 45,75 bedeutet, dass Si-
  e 75";
378 PRINT" DM
380 PRINT"auf + G e r a d e + = Spielnr. 45 se-
  tzen
385 PRINT"Einsatz: Minimum 1 DM Maximum 500 0
  M
390 PRINT"Anzahl der Wetten: maximal 6
395 INPUT"- - - - - Wollen wir spielen (J/N)";A$
400 IF A$="N" THEN 90
410 CLS
1550 REM Programmbeginn
1580 DIMA(50),B(100),C(100),T(100),X(30)
1600 FORI=1TO30:X(I)=0:NEXT:REM X loeschen
1610 P=1000:D=100000
1620 CLS:GOSUB 100
1621 GOSUBS200
1622 SCREEN1,1/3
1623 PRINT" 1/3 Kolonne 1/2 Ge Ug Rt
  Sw 0 00
1630 SCREEN1,9:INPUT"Wieviele Wetten ";Y
1640 IF Y>6 THEN 1620
1650 IF Y<1 OR Y<>INT(Y) THEN 1620
1660 FORI=1TO 50:A(I)=0:NEXT:REM A loeschen
1670 FORC=1TOY
1680 PRINT"Nummer";C;
1690 INPUTX,Z
1700 B(C)=Z
1710 T(C)=X
1720 IF X<1 OR X>50 OR X<>INT(X) THEN 1680
1730 IF Z<1 OR Z>500 THEN 1680
1740 IF Z<1 OR Z>500 THEN 1680
1750 IF A(X)=0 THEN 1780
1760 PRINT"Sie haben diese Wette schonmal!
1770 GOTO1680
1780 A(X)=1
1790 NEXT
1800 CLS
1810 P4=1:R=1:GOSUB4000
1820 R=2:GOSUB4000
1830 S=INT(100/2.6*RND(1))
1840 IF S=0 OR S>30 THEN 1830
1850 X(S)=X(S)+1
1860 IF S<37 THEN 1920
1870 IF S=37 THEN 1900
1880 PRINTTAB(15);"*****"
1892 PRINTTAB(15);" * 00 * "
1895 PRINTTAB(15);"*****"
1890 GOTO2020
1900 PRINTTAB(15);"*****"
1902 PRINTTAB(15);" * 0 * "
1904 PRINTTAB(15);"*****"
1910 GOTO2020
1920 RESTORE
1930 FORI=1TO18
1940 READ R
1950 IFR=S THEN 2000
1960 NEXT
1970 A$="Schwarz"
1980 PRINTTAB(15);"*****"
1982 PRINTTAB(15);" * ;S;A$;TAB(30) * "
1985 PRINTTAB(15);"*****"
1990 GOTO2020
2000 A$="Rot"
2010 GOTO1980
2020 PRINT
2030 FORC=1TOY
2040 IFT(C)<37 THEN 2710
2050 QNT(C)-36GOTO2090,2190,2220,2250,2300,2350
  ,2400,2470,2500
2060 QNT(C)-45GOTO2530,2560,2630
2070 GOTO2710
2080 STOP
2090 REM I=-1(37) 2:1
2100 IF S<12 THEN 2150
2110 PRINT"Sie haben";B(C);"DM in Wette";C;"ver-
  tonen
2120 D=D+B(C)
2130 P=P-B(C)
2140 GOTO2180
2150 PRINT"Sie haben";B(C)*2;"DM in Wette";C;
2151 PRINT"gewonnen!
2160 D=D-B(C)*2

```

```

2170 P=P+B(C)*2
2180 GOTO2810
2190 REM 13-24 (30) 2:1
2200 IF S<12 AND S<25 THEN 2150
2210 GOTO2110
2220 REM 25-36 (39) 2:1
2230 IF S<24 AND S<37 THEN 2150
2240 GOTO2110
2250 REM 1.Kolonne (40) 2:1
2260 FORI=1TO34 STEP3
2270 IF S=I THEN 2150
2280 NEXT
2290 GOTO2110
2300 REM 2.Kolonne (41) 2:1
2310 FORI=2TO35 STEP3
2320 IF S=I THEN 2150
2330 NEXT
2340 GOTO2110
2350 REM 3.Kolonne (42) 2:1
2360 FORI=3TO36 STEP3
2370 IF S=I THEN 2150
2380 NEXT
2390 GOTO2110
2400 REM 1-18 (43) 1:1
2410 IF S<19 THEN 2430
2420 GOTO2110
2430 PRINT"Sie haben";B(C);"DM in Wette";C;"gew-
  onnen!
2440 D=D-B(C)
2450 P=P+B(C)
2460 GOTO2810
2470 REM 19-36 (44) 1:1
2480 IF S<36 AND S<18 THEN 2430
2490 GOTO2110
2500 REM Gerade (45) 1:1
2510 IF S/2=INT(S/2) AND S<37 THEN 2430
2520 GOTO2110
2530 REM Ungerade (46) 1:1
2540 IF S/2<>INT(S/2) AND S<37 THEN 2430
2550 GOTO2110
2560 REM Rot (47) 1:1
2570 RESTORE
2580 FORI=1TO18
2590 READ R
2600 IF S=R THEN 2430
2610 NEXT
2620 GOTO2110
2630 REM Schwarz (%C) 1:1
2640 RESTORE
2650 FORI=1TO18
2660 READ R
2670 IF S=R THEN 2110
2680 NEXT
2690 IF S>36 THEN 2110
2700 GOTO2430
2710 REM 1-36,0,00 (1-36,49,50) 35:1
2720 IFT(C)<49 THEN 2760
2730 IFT(C)=49 AND S=37 THEN 2780
2740 IFT(C)=50 AND S=38 THEN 2780
2750 GOTO2110
2760 IFT(C)=S THEN 2780
2770 GOTO2110
2780 PRINT"Sie haben";B(C)*35;"DM in Wette";C;
2781 PRINT"gewonnen
2790 D=D-B(C)*35
2800 P=P+B(C)*35
2810 NEXT C
2820 PRINT
2830 PRINT"Spiegelgeld: ",Bank DM", " Sie DM"
2840 PRINT" ",D,P
2850 IF P=0 THEN 2890
2860 PRINT"Obacht! Sie spendeten Ihre letzte Ma-
  rk!
2870 GOTO3190
2880 IF D=0 THEN 2920
2890 PRINT"Sie haben die Bank gesprengt!
2900 P=101000.
2910 GOTO2960
2920 PRINT"Wollen Sie nochmal spielen?";
2930 INPUTS$
2940 IF S$="J" THEN CLS:GOTO1620
2950 DATA1,3,5,7,9,11,13,15,17,19,21,23,25,27
2951 DATA30,32,34,36
2960 IFF(1) THEN 3190
2970 PRINT"Du wen soll ich den Scheck ausstell-
  en?
2980 INPUTB$

```

```

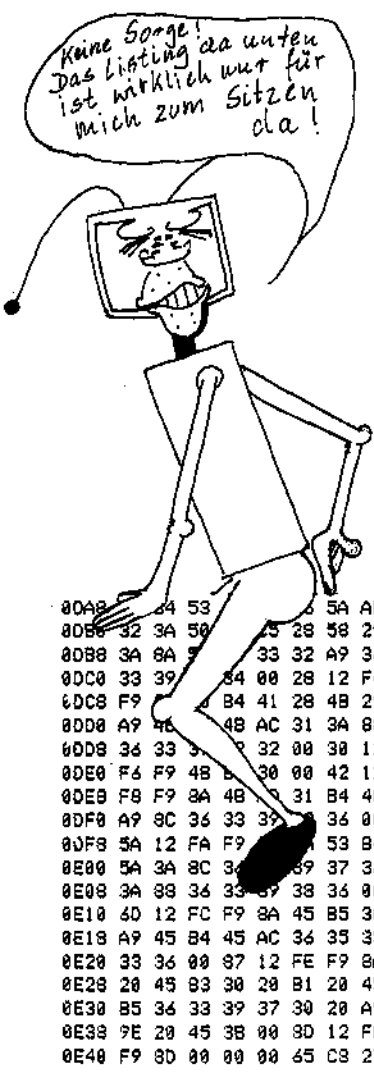
2990 CLS
3000 FOR I=1 TO 47:PRINT"-";NEXT
3010 PRINT"Check Nr. ";INT(100000+RND(1))
3030 GOSUB3230
3060 PRINT"Zahlen Sie auf diesen Scheck an:
3070 PRINT
3080 PRINTB$;" -- DM";P;"---
3090 PRINT
3110 PRINT"Bayer. Staatsbank Kto.Nr 165700-855
3120 PRINT
3130 PRINTTAB(35)"Der Computer"
3140 PRINT"Nuernberg ";J$;TAB(35)"_____";
3150 PRINT
3160 FOR I=1 TO 49:PRINT"-";NEXT
3170 PRINT"Kommen Sie bald mal wieder !!!"
3180 GOTO3210
3190 PRINT"ich danke fuer Ihr Geld!"
3200 PRINT"ich baue davon ein goldenes Roulette
-Rad!
3210 PRINT
3220 FOR A=1 TO 28000:NEXT:CLS:END
3230 REM
3310 M=0%
3330 RETURN
3400 END
4000 FOR I=1 TO 5
4010 POKE 2145,42
4020 GOSUB5000
4030 POKE 2145,32
4040 POKE 2148,42
4050 GOSUB5000
4060 POKE 2148,32
4070 POKE 2216,42
4080 GOSUB5000
4090 POKE 2216,32
4100 POKE 2282,42
4110 GOSUB5000
4120 POKE 2282,32
4130 POKE 2348,42
4140 GOSUB5000
4150 POKE 2348,32
4160 POKE 2413,42
4170 GOSUB5000
4180 POKE 2413,32
4190 POKE 2542,42
4200 GOSUB5000
4210 POKE 2542,32
4220 POKE 2669,42
4230 GOSUB5000
4240 POKE 2669,32
4250 POKE 2732,42
4260 GOSUB5000
4270 POKE 2732,32
4280 POKE 2794,42
4290 GOSUB5000
4300 POKE 2794,32
4310 POKE 2856,42
4320 GOSUB5000
4330 POKE 2856,32
4340 POKE 2916,42
4350 GOSUB5000
4360 POKE 2916,32
4370 POKE 2913,42
4380 GOSUB5000
4390 POKE 2913,32
4400 POKE 2910,42
4410 GOSUB5000
4420 POKE 2918,32
4430 POKE 2842,42
4440 GOSUB5000
4450 POKE 2842,32
4460 POKE 2776,42
4470 GOSUB5000
4480 POKE 2776,32
4490 POKE 2710,42
4500 GOSUB5000
4510 POKE 2710,32
4520 POKE 2645,42
4530 GOSUB5000
4540 POKE 2645,32
4550 POKE 2516,42
4560 GOSUB5000
4570 POKE 2516,32
4580 POKE 2389,42
4590 GOSUB5000
4600 POKE 2389,32
4610 POKE 2326,42
4620 GOSUB5000

```

```

4630 POKE 2326,32
4640 POKE 2264,42
4650 GOSUB5000
4660 POKE 2264,32
4670 POKE 2202,42
4680 GOSUB5000
4690 POKE 2202,32
4700 POKE 2142,42
4710 GOSUB5000
4720 POKE 2142,32
4800 NEXT
4810 RETURN
5000 FOR P3=1 TO 200 STEPP4:NEXT
5010 ONRGUT05050,5100
5050 P4=P4+1.1
5060 RETURN
5100 P4=P4/1.1
5110 RETURN
5200 DOKE3019,14131
5210 POKE3021,45
5220 DOKE3022,14443
5230 DOKE3027,12340
5240 POKE3029,45
5250 DOKE3030,12852
5260 DOKE3035,13108
5270 POKE3037,45
5280 DOKE3039,13344
5290 DOKE3042,13620
5300 DOKE3046,13976
5310 DOKE3050,14132
5320 DOKE3054,14388
5330 DOKE3059,14644
5340 DOKE3063,12341
5350 RETURN
ok

```



# Mitarbeiter

Wenn Sie Fragen zu Artikeln, Programmen oder Schaltungen haben, sollten Sie sich zunächst direkt an die Autoren wenden. Deshalb hier die Anschriften der Mitarbeiter der letzten beiden Ausgaben.

- |   |  |
|---|--|
| Georg Assmann<br>[redacted]<br>Herzhausen     | Tom D. Rudebusch<br>[redacted]<br>Karlsruhe                        |
| Peter Brendel<br>[redacted]<br>Mannheim       | Michael Bach<br>[redacted]<br>Gundelfingen                         |
| Helmut Emmelmann<br>[redacted]<br>Karlsruhe   | Horst Dieckhoff<br>[redacted]<br>Berlin 19                         |
| Peter Huntemann<br>[redacted]<br>Lohfelden    | Eberhard Horch<br>[redacted]<br>Hannover 72                        |
| Henrik Maier<br>[redacted]<br>Baden-Baden     | Gerhard Klement<br>[redacted]<br>A- [redacted] Wien<br>Osterreich  |
| Klaus Numbaur<br>[redacted]<br>Wendelstein    | Karl Mark<br>[redacted]<br>Essen                                   |
| Dieter Oberle<br>[redacted]<br>Vollmersweiler | Wolfgang Mayer-Gurr<br>[redacted]<br>Recklinghausen                |
| Manfred Stoll<br>[redacted]<br>Hamburg        | Constantin Olbrich<br>[redacted]<br>Berlin                         |
| Jürgen Weiermann<br>[redacted]<br>Nieddigen   | H. Schulte<br>[redacted]<br>A- [redacted] Klagenfurt<br>Osterreich |
|   | Carl Inust<br>[redacted]<br>Roenthal                               |
|   | Jörg Wittlich<br>[redacted]<br>[redacted]<br>Karlsruhe             |

# HULL-FORTH

VON CONSTANTIN OLBRICH

Review : Hull-Forth und NAS-FORTH V 1.1

In meiner kleinen Reihe der Reviews neuer Nascom Software stelle ich heute zwei Forth Implementationen vor, die es seit einiger Zeit gibt:

1. Hull - Forth von A.F.T. Winfield ( # 25 )
2. NAS-FORTH V 1.1 von D. Husband ( # 25 )

Voraussetzend moechte ich klarstellen, dass ich kein ausgesprochener FORTH - Fan bin. Wahrscheinlich schon deshalb nicht, weil ich mir die ca. 150 verschiedenen FORTH Woerter nicht merken kann.

Beiden Forth Implementationen ist eine ausgesprochen schlechte Cassettenqualitaet gemeinsam. Fuer ca. 110 DM kann man eigentlich ein vernuenftiges Cassettenmaterial erwarten. Die Cassette von D. Husband war so schlecht, dass die nur einmal auf dem Band vorhandene 300 Baud Kopie ueberhaupt nicht lesbar war, und ich gezwungen war, mir die fehlenden Bytes aus den Fingern zu saugen. Die Reklamation ist bis heute nicht beantwortet worden (!), zum Glueck konnte ich nach einigem Studium des internen Forth Aufbaus ( durch die hohe Regularitaet des Dictionarys ) die fehlenden Bytes rekonstruieren.

Die Handbuecher sind vom Seitenumfang ungewoehnlich stark. Jedoch fehlen besonders beim NAS-FORTH die elementarsten Informationen, um mit dem System zurande zu kommen. Diskret wird da auf die Kompatibilitaet mit Forth 79 hingewiesen und daB es darueber genugend Buecher gaebe. Merkwueerdigerweise wird das Buch des A.F.T. Winfield ( der Autor des Hull-Forth ) besonders empfohlen. Mit dem 35 DM teuren Taschenbuch "The complete FORTH" kann man dann endlich seine ersten Versueche starten. Im Hull-Forth Manual gibt's immerhin ein Kapitel fuer den Forth-Neuling, den 'self teaching guide'.

Doch nun zum Forth selbst: Vergessen muss man zunaechst das gewoehnte Bildschirmeditieren. Wird noch beim NAS-Forth die Funktion der Cursor- Tasten einfach gesperrt, so kann man im Hull Forth zwar auf dem Bildschirm hin und her gehen, erzeugt damit aber nur

Fehlermeldungen oder Abstuerze. Nach guter alter Teletype- Manier geht's unerschrocken weiter. In der Reklame des D. Husband ist zwar von einem Screen Editor die Rede, worauf ich natuerlich reinfiel. Vielmehr handelt es sich bei diesen Forth Screens um 1 KB Speicherbereiche fuer 16 Zeilen a 64 Zeichen, die stets "optimal" mit Leerzeichen aufgefuellt werden, um so wesentlich mehr an Speicherplatz einzunehmen als eigentlich fuer den Text noetig.

Der im NAS-Forth ebenfalls integrierte Z 80 Assembler ist in Wirklichkeit ein 8080 ASM mit aufgepfropften Z 80 Befehlen, ein Graus fuer jeden Z 80 Programmierer. Dann auch noch die UPN (Umgekehrt Polnische Notation) im Assemblertext! Kostprobe :

```
W INX H PUSH I STAX fuer
INC DE, PUSH HL, LD (BC),A
Ganz klar, nicht ?
```

Im NAS-Forth gibt es zudem noch eine fuer den Benutzer voellig undurchsichtige Hierarchie der Forth Woerter. Staendig muss man eine bestimmte Eingabereihenfolge beachten, oder soll ich sagen erfinden; denn in den 75 Seiten Dokumentation steht nichts dergleichen. Loescht man z.B. die Screen 1 mit 1 CLEAR und hat vorher nicht das Forth Wort EDITOR gegeben, (welches natuerlich ueberhaupt nicht einmal erwaeht wird ), stuerzt das System erst einmal ab. Das Konzept heisst hier wohl "Trial And Error".

Um ganz ehrlich zu sein: nach etwa 3-4 Tagen des Herumprobierens mit NAS-FORTH und dem Forth Buch konnte immer noch nicht vernuenftig mit dem System umgehen, zumal staendige Widersprueche in Dokumentation und Handbuch verbunden mit Systemzusammenstuerzen mir die Lust nahmen.

Im Hull Forth geht es da etwas einfacher und uebersichtlicher zu, und man kann nach kurzer Zeit die ersten Screens ( ebenfalls in diesem bescheuerten Format ) abspeichern. Es gibt sogar eine (fast funktionierende) Moeglichkeit, sich durch eine Katalogfunktion ueber den Inhalt einer Cassette zu informieren.

Weiterhin ist im Manual der Sourcetext der schon in FORTH geschriebenen Standard Forth Woerter abgedruckt. Die Editiermoeglichkeiten innerhalb eines Screen sind aber auch

spartanisch bis vorsintflutlich. Gut finde ich die Moeglichkeit des Decompilierens eines Forth Wortes aus dem Dictionary, womit man bei unbekannter Funktion auf die Source des Wortes schliessen kann. Der NAS-Forth Autor verlangt dafuer noch mal # 5 extra nur fuer ein Listing dieser Funktion. Der Hull Forth Benutzer erfahret ebenfalls, wie man eigene Strukturen wie REPEAT WHILE usw. selbst integriert, anhand von Beispielen, die aber mit Fehlern abgedruckt worden sind. Insgesamt hat man aber mehr Informationen ueber den internen Aufbau von FORTH als bei der NAS-Forth Version. Beim Lernen von Forth ist aber die automatische Stack Anzeige beim NAS-Forth nuetzlich, mit der man sofort die Wirkungsweise der elementaren Stackmanipulationsbefehle testen kann.

An dieser Stelle sei mir ein kritisches Wort zum aktuellen Forth Fieber gestattet: Fuer mich ist eine Programmiersprache nur Mittel zum Zweck und kein Selbstzweck. In diesen Forth Interpretern/ Compilern sehe ich weder den Vorzug einer erhoehnten Ausfuehrungsgeschwindigkeit (in meinen Beispielen war PASCAL um Groessenordnungen schneller), noch den Vorzug einer kuerzeren Programmierzeit. Da es praktisch nicht moeglich ist, die Forth Screen vernuenftig zu editieren, steigt der Aufwand an Tipparbeit. Die in der Literatur stets erwahnte Maschinennaehelike und die damit verbundene Transparenz fuer den Benutzer kann ich nur teilweise nachvollziehen. Vielmehr stellen moderne PASCAL Compiler wie das Hisoft Pascal oder BLS Pascal eine Fuelle von hardwareorientierten Funktionen und Assembler-schnittstellen zur Verfuegung, dass ich auf die fuer mich umstaendlich wirkende Programmierarbeit mit Forth verzichten werde.

Mein Eindruck ist, dass in beiden Faellen der 'FORTH Entwickler' sich ein 8080 Forth besorgt und recht schlampig an den NASCOM angepasst hat. So ist der gesamte Opcode Befehlsvorrat des Hull Forth Compilers nur: JP, JP Z, JP NZ, LD HL,nnnn, CALL adr, RET. Etwas duerftig, finde ich!

Wer sich dennoch unerschrocken in das Forth Abenteuer stuerzen will, sollte sich erst einmal mit dem bereits veroeffentlichten T-Forth des Herrn Klement beschaeftigen, um zu pruefen, ob diese Art des Programmierens

ueberhaupt gewuenscht wird (und man spart Geld).

Zu Hoffen bleibt dann nur noch, dass diese beiden Forth Versionen ( beide V 1,1 !! ) fuer den NASCOM nicht den zukuenftigen Software Standard bestimmen werden.

## Buchfuehrung

VON KLAUS MOMBAUR

Hier nun endlich wieder etwas fuer unsere BASIC- Fans. Vielleicht ist die Anwendung auch fuer einen "Loeter" oder "Maschinensprachler" interessant, der dazu einmal wieder zum BASIC greift, zumal es ja mit Maschinencode vermischt ist.

Zwei "Kaeferchen" seien gleich zuvor genannt:

```
Zeile 50: CLEAR (1000): CLS
Zeile 8336: IFT=32THENGOSUB200:RETURN
```

Folgendes Programm wurde auf NASCOM2 mit NASSYS1 erstellt. Es belegt C80- ca. 4A00 mit Maschinenprogrammen und BASIC. 4F00- 4FFF ist Kontenspeicherbereich, und 5000- 8FFF ist RAM fuer Daten. Dies kann in den Zeilen 200 bis 245 geaendert werden. Das Programm ist so angelegt, daB man nur die Daten allein unter NASSYS abspeichern kann. Es berechnet selbst die Adressen fuer den W-Befehl. Dabei werden die Daten sehr wirtschaftlich im RAM abgelegt, d.h. wenige Buchungen ergeben ein schnelleres Abspeichern. Das Programm sollte evt. bei Vereinen, kleinen Geschaeften, 80-Bus-Journal-Verwaltungen, Haushalt etc. einsetzbar sein.

Ein kleiner Auszug eines gedruckten Beleges:

DM	Vereinskasse - Eingaenge:	Mai 83
48.00	Böhm Günter	
48.00	Kreidl Günter	
999.99	Schulmeister Karl	

DM	Vereinskasse - Ausgaenge:	Mai 83
54.30	Porto 10.S.	
29.46	Papier	

Saldovortrag:	DM	5.00
+ Summe Eingaenge:	DM	1095.99
- Summe Ausgaenge:	DM	83.76
		=====
neuer Saldo:	DM	1007.23
		=====

Beim allerersten Initiieren des Programmes muß zweimal das "n"-Kommando gegeben werden, um den Saldospeicher auf Null zu setzen. Wenn das Menu nichts anderes verlangt, wird zur Fortsetzung des Programms eine beliebige Taste gedruekt. Das BASIC ist auf 20000 zu begrenzen.

```

10 REM -----BUCHFUEHRUNG 4/83-----
11 REM
12 REM C. by K.Nombaur,
13 REM Nuernberg
20 REM
30 REM
40 CLS
50 CLEAR(100)
100 REM TASTATUR - ABFRAGE -----
105 RESTORE 110
110 DATA 25311 :REM DF 62 IN
120 DATA 312 :REM 38 01 JR C
130 DATA 18351 :REM AF XOR A
140 DATA 16927 :REM 47 LD B, A
150 DATA -8179 :REM 2A 0D E0 LD HL (E800)
160 DATA 233 :REM E9 JP (HL)
100 FORJ=3200TO3218STEP2:READD:DOKEJ,D:NEXT
190 GOSUB 200:GOTO 500
200 REM U - DATENSPEICHER initialisieren -----
205 KS=20200 :REM 4F00H KontoSpeicherbeginn
210 RB=20400 :REM 5000H RamBeginn
220 SB=RB :REM 5000H Suchbeginn
230 SR=16303 :REM 3FFFH SuchRegion
235 RE=-28473:REM 0FFFH RamEnde
240 KE=KS+191:REM 4FFFH KontospeicherEnde
245 ZW=KS+128:REM 4FC0H Zwischenspeicher
250 DOKE 3255, RB :REM 0CB7/8
255 DOKE 3250, SR :REM 0CBA/8
260 RETURN
300 REM DATENSPEICHER loeschen -----
301 CLS
302 POKE3111,1
303 SCREEN10,5:INPUT"Bitte den Monatsnamen ";M#
307 SCREEN10,7:PRINT"Bitte warten"
308 POKE3111,0
309 RESTORE310
310 DATA8623,31743 :REM XOR A,LD HL RAMEND
315 DATA17,4672 :REM LD DE KONTOSP
320 DATA-6893,21229 :REM LD (DE) A
330 DATA8417 :REM INC DE PUSH HL
340 DATA-13832 :REM SBC HL DE
350 DATA 0 :REM POP HL
360 DATA 0 :REM JR NZ ^
370 DATA 0 :REM RET
300 FORJ=3212TO3228STEP2:READD:DOKEJ,D:NEXT
384 DOKE 3214,RE-10 :REM RAMEND-10 (0CBE/F)
386 DOKE 3217,RB :REM KONTOSP (0C91/2)
387 DOKE4100,3212:A=USR(0)
388 POKE RB,ASC("0"):REM (RAMBEG)= 0
390 REM Konten/Mo.speicher loeschen (dez 8)
391 FOR A= KS-16 TO KE:POKEA,40:NEXT
392 B=KS-16:REM Monatspeicher laden
393 FORA=ITOLEN(M#):POKEB,ASC(MID$(M#,A,1))
394 B=B+1:NEXT
395 GOTO 900
400 REM SaldoSpei in Vortragsspeicher -----
410 B=KS+88
420 FOR A=KS-64 TO A+40
430 POKE A,PEEK(B):B=B+1
440 NEXT
450 GOTO 300
500 REM Speicherende markieren -----
510 FOR A=1 TO 10
520 POKE RE-A,A:NEXT
600 REM Rechenpro - Addition/Subtraction -----
610 RESTORE 610
621 DATA -29747 :REM CALL E90BH :: HL auf
622 DATA 16185 :REM LD A "*" :: Pfg Pos
623 DATA -5334 :REM EX HL DE
624 DATA -24083 :REM CPI
625 DATA -992 :REM JR NZ FCH
626 DATA 11838 :REM LD A "."
627 DATA -24083 :REM CPI
628 DATA -992 :REM JR NZ FCH ::(3294)
629 DATA -4829 :REM INC HL ::(3295/6)
630 DATA -12685 :REM LD (CCCEH) SP:: Addition
631 DATA 14892 :REM SCF :: Progr
632 DATA 4597 :REM PUSH AF
633 DATA 16199 :REM LD DE KTOEND :: (3304)
634 DATA 2310 :REM LD C 0?
635 DATA -306 :REM SEG LD A (HL)
636 DATA 14382 :REM CP ZEN
637 DATA 8229 :REM JR C END1
638 DATA 11018 :REM JR NZ BEG1
639 DATA -10626 :REM DEC HL
640 DATA 10224 :REM LD A (HL)
641 DATA -10726 :REM BEG1 SUB 30H
642 DATA -32720 :REM LD B A
643 DATA 18215 :REM BKTO LD A (DE)
644 DATA 30961 :REM SUB 30H
645 DATA 824 :REM ADD A B / SUB B
646 DATA 454 :REM DAA
647 DATA -473 :REM LD B A
648 DATA 14346 :REM POP AF
649 DATA -10750 :REM LD A B
650 DATA -2800 :REM JR C BK1
651 DATA 12486 :REM ADD A 81/ SUB 01

```

```

652 DATA 3346 :REM DAA
653 DATA 2344 :REM BK1 CP 80H
654 DATA 11835 :REM JR C BK2
655 DATA -18728 :REM
656 DATA -4063 :REM SUB 10H / SUB 90H
657 DATA 6207 :REM BK2 PUSH AF
658 DATA -4655 :REM ADD A 30H
659 DATA -12677 :REM LD (DE) A
660 DATA -14868 :REM DEC C
661 DATA 0 :REM JR Z END2
662 DATA 0 :REM DEC DE DEC HL
663 DATA 0 :REM JR BEG
664 DATA 0 :REM END1 LD HL ADRNUL
665 DATA 0 :REM JR BEG
666 DATA 0 :REM END2 LD SP (CCCEH)
667 DATA 0 :REM RET
668 DATA 0 :REM
600 FORJ=3200 TO3360STEP2:READD:DOKEJ,D:NEXT
681 REM CDB - 8020
685 DOKE 3351, KS+176 :REM ADRNUL (0D17)
700 REM RAM nach Daten absuchen -----
710 RESTORE 710
720 DATA -29747, 31721 :REM CALL E90B, LD A E
725 DATA 33, 320 :REM LD HL RAMBEGIN
730 DATA15104,-19967 :REM LD BC SUCHREG
740 DATA 17780,-3379 :REM CPIR
750 DATA -13840, 8,0 :REM LD A H, LD B L
760 REM :REM CALL F0F2
770 REM :REM RET
700 FORJ=3250 TO3270STEP2:READD:DOKEJ,D:NEXT
790 DOKE 3255, RB :REM RAMBEGIN (0CB7/8)
795 DOKE 3250, SR :REM SUCHREG (0CBA/8)
800 REM Monat laden -----
805 M#=""
810 FOR A=KS-16 TO KS-6
820 A=CHR$(PEEK(A))
830 IF A#="0" THEN M#M#A
840 M#M#A# :REM
900 REM Druckprogramm -----
910 RESTORE 910
920 DATA3982, 1747,-12482
921 DATA2083,-2754,2083
922 DATA-194, 1491,-24629
923 DATA1491,-8245,1491
924 DATA22561,-8435,8561
925 DATA3419,-38754,-14068
926 DATA30053,-2816,25037
927 DATA-3027,-11319,-9460
928 DATA-13563,-11327,-13563
929 DATA-11313,-9467,-13563
930 DATA 8295,-13038,-8440
931 DATA -14082, 0,0
950 FORJ=3300TO3450STEP2:READD:DOKEJ,D:NEXT
951 REM 0D34 - 0D70
960 DOKE4100,3300:A=USR(0):REM Dru pro ini
961 DOKE4100,3445:A=USR(0):REM Dru aus
980 CLS:DR=0:TP=""
1000 REM MENUE -----
1005 TP="" B u c h f u e h r u n g
1010 GOSUB 8200
1011 PRINT:SCREEN12,1
1012 FORA=ITOLEN(TP)*3:PRINTCHR$(126);
1013 NEXT
1014 PRINT:PRINT
1020 PRINTTAB(5)"Postcheckkonto - P"
1030 PRINTTAB(5)"Bankkonto - B"
1040 PRINTTAB(5)"Sparkonto - S"
1050 PRINTTAB(5)"Vereinskasse - V"
1060 PRINT
1070 PRINTTAB(5)"Daten speichern - A"
1072 PRINTTAB(5)"Daten einlesen - L"
1074 PRINTTAB(5)"Neuer Monat - n"
1080 PRINT:SCREEN25,12:PRINTCHR$(63)
1090 GOSUB200:GOSUB 8100
1100 IFE$="P" THEN 2000
1102 IFE$="B" THEN 3000
1104 IFE$="S" THEN 4000
1106 IFE$="V" THEN 5000
1110 IFE$="A" THEN 7000
1112 IFE$="L" THEN 7500
1114 IFE$="n" THEN 400
1120 CLS:SCREEN7,5
1130 PRINT"Diese Abkuerzung ist nicht vorhanden"
1140 GOSUB 8000
1150 CLS:GOTO1000
1200 REM MENUE 2 -----
1210 CLS
1220 SCREEN 6,6
1230 PRINT"Eingaenge - E"
1240 PRINT
1250 PRINTTAB(5)"Ausgaenge - A"
1255 PRINT
1260 PRINTTAB(5)"Konto-Stand - K"
1265 PRINT:PRINT
1270 PRINTTAB(5)"Drucken - D"
1280 GOSUB 8100

```

```

1290 RETURN
1300 REM MENUE 3 -----
1310 CLS
1320 SCREEN 6,6
1330 PRINT "Neue Buchungen - B"
1340 PRINT
1350 PRINTTAB(5)"Buchungen auflisten"
1355 PRINT
1360 GOSUB 8100
1365 RETURN
2000 REM Postscheck -----
2010 S=1:GOSUB 1200:REM Code PS-Eing
2020 IF E$="K" THEN 2000
2030 IF E$="A" THEN 2500
2040 IF E$="D" THEN DR=1
2050 IFDR=1 THEN GOSUB9000
2100 REM PS-Eingaenge xxxxxxxxxxxxxxxxxxxxxxxx
2110 KN=KS+7:REM Kontospeicherende
2115 TP=""DM Postscheck - Eingaenge:
2118 IFDR=1 THEN PRINTTAB(5)TP#M#:GOTO2140
2120 GOSUB1300:IFE$="B" THEN GOSUB8804:GOTO2200
2125 CLS
2130 GOSUB 8200
2135 PRINT
2140 GOSUB 8300
2150 IFDR=1 THEN S=13:GOSUB9900:GOTO2500
2160 GOSUB 8000
2200 TB=UI:S=1
2210 DOKE 3304,KN :REM (0CEB/9)
2220 DOKE4100,3200:X=USR(TB):REM Betrag add
2230 GOTO2160
2500 REM PS-Ausgaenge xxxxxxxxxxxxxxxxxxxxxxxx
2510 S=2:KN=KS+55
2515 TP=""DM Postscheck - Ausgaenge:
2518 IFDR=1 THEN PRINTTAB(5)TP#M#:GOTO2540
2520 GOSUB1300:IFE$="B" THEN GOSUB8804:GOTO2600
2525 CLS
2530 GOSUB 8200
2535 PRINT
2540 GOSUB 8300
2550 IFDR=1 THEN 2000
2560 GOSUB 8800
2600 S=2:TB=UI
2610 DOKE 3304,KN
2620 DOKE4100,3200:X=USR(TB):REM Betrag add
2630 GOTO2560
2900 REM PS-Kto Stand xxxxxxxxxxxxxxxxxxxxxxxx
2902 TP="" Postscheckkonto - M#
2904 IFDR=1 THEN 2010
2905 CLS
2906 GOSUB 8200
2910 B#=""WA=KS-64:REM Speianfg Vortrag
2920 VA=VA:UC=VA
2930 GOSUB 9400
2950 B#=""WA=KS:REM Speianfg Eing Su
2955 UC=VA
2960 GOSUB 9450
2970 B#=""WA=KS+40:REM Speianfg Ausg Su
2972 UC=VA
2975 GOSUB 9470
2980 B#=""WA=ZW:REM ZWSpeicher hat Saldo
2982 UC=VA
2984 US=KS+00:REM Speianfg Saldo
2985 GOSUB 9500
2910 X=13:GOSUB9900:GOSUB9900
2915 DOKE4100,3445:A=USR(0):REM Dru aus(0D75)
2920 GOTO 900
3000 REM Bank -----
3010 S=3:GOSUB 1200
3020 IF E$="K" THEN 3000
3030 IF E$="A" THEN 3500
3040 IF E$="D" THEN DR=1
3050 IFDR=1 THEN GOSUB9000
3100 REM BK-Eing
3110 KN=KS+15
3115 TP=""DM Bank - Eingaenge:
3118 IFDR=1 THEN PRINTTAB(5)TP#M#:GOTO3140
3120 GOSUB1300:IFE$="B" THEN GOSUB8804:GOTO3200
3125 CLS
3130 GOSUB 8200
3135 PRINT
3140 GOSUB 8300
3150 IFDR=1 THEN S=13:GOSUB9900:GOTO3500
3160 GOSUB 8300
3200 TB=UI:S=3
3210 DOKE 3304,KN :REM (0CEB/9)
3220 DOKE4100,3200:X=USR(TB):REM Betrag add
3230 GOTO3160
3500 REM PS-Ausg
3510 S=4:KN=KS+63
3515 TP=""DM Bank - Ausgaenge:
3518 IFDR=1 THEN PRINTTAB(5)TP#M#:GOTO3540
3520 GOSUB1300:IFE$="B" THEN GOSUB8804:GOTO3600
3525 CLS
3530 GOSUB 8200
3535 PRINT
3540 GOSUB 8300

```

```

3550 IFDR=1THEN3800
3560 GOSUB 8900
3600 S=4: TB=UI
3610 DOKE 3304,KN
3620 DOKE4100,3200: X=USR(TB): REM Betrag add
3630 GOTO3560
3680 REM BK-Kto Stand
3682 TP=" - Bankkonto - "+M$
3684 IFDR=1THEN3810
3685 CLS
3686 GOSUB 8200
3610 B="":VA=KS-56
3620 UB=VA: UC=VA
3630 GOSUB 9400
3650 B="":VA=KS+8
3655 UC=VA
3660 GOSUB 9450
3690 B="":VA=KS+56
3692 UC=VA
3695 GOSUB 9470
3698 B="":VA=ZU
3692 UC=VA
3694 US=KS+96
3695 GOSUB 9500
3610 X=13:GOSUB9900:GOSUB9900
3615 DOKE4100,3445:VA=USR(0):REM Dru aus(0D75)
3620 GOTO 900
4000 REM Sparbuch
4010 S=5: GOSUB 1200
4020 IF E="K"THEN 4000
4030 IF E="A"THEN 4500
4040 IF E="D"THEN DR=1
4050 IFDR=1THENGOSUB9800
4100 REM Spar-Eingaenge
4110 KN=KS+23
4115 TP="DM Sparkonto - Eingaenge: "
4118 IFDR=1THENPRINTTAB(5)TP+M$:GOTO4140
4120 GOSUB1300:IFE="B"THENGOSUB8840:GOTO4200
4125 CLS
4130 GOSUB 8200
4135 PRINT
4140 GOSUB 8300
4150 IFDR=1THENX=13:GOSUB9900:GOTO4500
4160 GOSUB 8800
4200 TB=UI: S=5
4210 DOKE 3304,KN:REM (0CE8/9)
4220 DOKE4100,3200: X=USR(TB)
4230 GOTO4160
4500 REM Spar-Ausgaenge
4510 S=6: KN=KS+71
4515 TP="DM Sparkonto - Ausgaenge: "
4518 IFDR=1THENPRINTTAB(5)TP+M$:GOTO4540
4520 GOSUB1300:IFE="B"THENGOSUB8840:GOTO4600
4525 CLS
4530 GOSUB 8200
4535 PRINT
4540 GOSUB 8300
4550 IFDR=1THEN4000
4560 GOSUB 8800
4600 S=6: TB=UI
4610 DOKE 3304,KN
4620 DOKE4100,3200: X=USR(TB)
4630 GOTO4560
4800 REM Spar-Kto Stand
4802 TP=" - Sparkonto - "+M$
4804 IFDR=1THEN4610
4805 CLS
4806 GOSUB 8200
4810 B="":VA=KS-40
4820 UB=VA: UC=VA
4830 GOSUB 9400
4850 B="":VA=KS+16
4855 UC=VA
4860 GOSUB 9450
4890 B="":VA=KS+64
4892 UC=VA
4895 GOSUB 9470
4900 B="":VA=ZU
4902 UC=VA
4904 US=KS+104
4905 GOSUB 9500
4910 X=13:GOSUB9900:GOSUB9900
4915 DOKE4100,3445:VA=USR(0):REM Dru aus(0D75)
4920 GOTO 900
5000 REM Vereinskasse
5010 S=7: GOSUB 1200
5020 IF E="K"THEN 5000
5030 IF E="A"THEN 5500
5040 IF E="D"THEN DR=1
5050 IFDR=1THENGOSUB9800
5100 REM U.-Eingaenge
5110 KN=KS+31
5115 TP="DM Vereinskasse - Eingaenge: "
5118 IFDR=1THENPRINTTAB(5)TP+M$:GOTO5140
5120 GOSUB1300:IFE="B"THENGOSUB8840:GOTO5200
5125 CLS
5130 GOSUB 8200

```

```

5135 PRINT
5140 GOSUB 8300
5150 IFDR=1THENX=13:GOSUB9900:GOTO5500
5160 GOSUB 8800
5200 TB=UI: S=7
5210 DOKE 3304,KN
5220 DOKE4100,3200: X=USR(TB)
5230 GOTO5160
5500 REM U.-Ausgaenge
5510 S=8: KN=KS+79
5515 TP="DM Vereinskasse - Ausgaenge: "
5518 IFDR=1THENPRINTTAB(5)TP+M$:GOTO5540
5520 GOSUB1300:IFE="B"THENGOSUB8840:GOTO5600
5525 CLS
5530 GOSUB 8200
5535 PRINT
5540 GOSUB 8300
5550 IFDR=1THEN5000
5560 GOSUB 8800
5600 S=8: TB=UI
5610 DOKE 3304,KN
5620 DOKE4100,3200: X=USR(TB)
5630 GOTO5540
5800 REM U.-Kto Stand
5802 TP=" - Vereinskasse - "+M$
5804 IFDR=1THENS810
5805 CLS
5806 GOSUB 8200
5810 B="":VA=KS-40
5820 UB=VA: UC=VA
5830 GOSUB 9400
5850 B="":VA=KS+24
5855 UC=VA
5860 GOSUB 9450
5890 B="":VA=KS+72
5892 UC=VA
5895 GOSUB 9470
5900 B="":VA=ZU
5902 UC=VA
5904 US=KS+112
5905 GOSUB 9500
5910 X=13:GOSUB9900:GOSUB9900
5915 DOKE4100,3445:VA=USR(0)
5920 GOTO 900
6999 REM Unterprogramme =====
7000 REM DATEN SPEICHERN
7005 RESTORE7010: CLS: S=ASC("2")
7010 DATA -29747, 31721:REM CALL E90B, LD A E
7015 DATA 33, 320 :REM LD HL RAMBEGIN
7020 DATA 15104, -19987:REM LD BC SUCHREGION
7025 DATA 20753, -4856 :REM CPJR
7030 DATA 10579, -8436 :REM LD DE CURSORPOS
7035 DATA -13978, 0 :REM LD (0C29) DE
7040 DATA 0 :REM TBCD3
7045 DATA 0 :REM RET
7050 FORJ=3250TO3274STEP2:READB:DOKEJ,B:JNEXT
7054 DOKE3255, RB
7056 DOKE3258, RB
7060 SCREENS,5
7070 PRINT"Band aufnahmebereit?"
7080 GOSUB 8100
7090 SCREENI,2:PRINT"W 400":REM KS-64 in HEX
7100 SCREENI,5:PRINT CHR$(27)
7110 DOKE4100,3250
7120 X=USR(S)
7130 SCREENI,1:MONITOR
7500 REM DATEN EINLESEN
7510 CLS
7520 SCREENS,5
7530 PRINT"Stehet das Band bereit?"
7540 GOSUB 8100
7550 SCREENI,2
7560 PRINT"R"
7570 SCREENS,5:PRINT"Nachher -2- und -RUN-"
7580 SCREENI,1:MONITOR
7999 END
8000 REM U - Zeitschleife
8010 FOR T=1 TO 2000: NEXT
8020 RETURN
8100 REM U - Tastaturabfrage
8105 DOKE4100,3200
8110 E=USR(0):IFE=0THENR8110
8120 E=CHR$(E)
8130 RETURN
8200 REM U - TOP-Zeile
8210 A=3026
8220 FORT=ITOLEN(TP):
8230 A=MI0*(TP,T,1)
8240 POKEA,ASC(A#)
8250 A=A+1
8260 NEXT
8270 RETURN
8300 REM Text und Betrag auslesen
8302 L=0
8305 DOKE4100,3250: TB=USR(S):REM S=Suchzchn
8310 REM TB=RAM-PDS Beginn Text/Betrag
8320 T$="": B$=""

```

```

8322 IFTB (RE+10)THEN GOSUB200:RETURN
8325 REM Auslesen
8328 UI=TB
8330 T=PEEK(TB)
8335 IFT=42)THENB=TB+2:GOTO8345
8340 T=T+CHR$(T): TB=TB+1: GOTO8330
8345 B=PEEK(TB)
8350 IF B(32) THEN 8345
8355 IF B=64 THENGOSUB8600:GOSUB200:RETURN
8360 B=B+CHR$(B): TB=TB+1: GOTO 8345
8365 S=TB
8370 DOKE3255,SN :REM nae. SUCHBEGINN
8380 GOSUB8600:GOTO 8365
8400 REM Text und Betrag - Eingabe
8402 CLS:GOSUB8200
8404 E1$="": POKE3111,1
8410 SCREENI,8: INPUT Text: %E1$
8420 IFLEN(E1$)>31THENPRINT"Zu lang!":GOTO8410
8430 E2=0:PRINT: POKE3111,0
8432 INPUT Betrag: %E2
8433 IFE2>9999,99THENPRINT"Betrag zu hoch!":GOTO
8438
8434 PRINT:PRINT:PRINTTAB(15)"K- wenn Korrigie
ren"
8435 PRINTTAB(15)"S- wenn speichern"
8436 GOSUB 8100
8437 IFE="K"THEN 8400
8438 IFE="S"THEN 8436
8440 E2=INT(E2/100+.5)/100: E2=STR$(E2)
8450 IF ASC(RIGHT$(E2,3))=46 THEN RETURN
8460 IF ASC(RIGHT$(E2,2))=46 THEN 8400
8470 E2=E2+.00": RETURN
8480 E2=STR$(E2): RETURN
8500 REM Text und Betrag speichern
8510 S=ASC("2"): REM RAMEND (?) suchen
8520 DOKE4100,3250: TB=USR(S):REM S=Suchzchn
8525 REM TB=1.FRERAM
8528 IFTB(RE+40)THENPRINT"Speicher ist voll!":G
OSUB8100:GOTO900
8530 TB=TB-1: POKE TB,C2: REM Codexchn statt 2
8532 UE=TB
8535 TB=TB+1
8540 FORA=ITOLEN(E1$)
8545 POKE TB,ASC(MID$(E1$,A,1)): TB=TB+1: NEXT
8550 POKE TB,ASC(" "): TB=TB+1:
8560 FORA=ITOLEN(E2$)
8565 IFLEN(E2$)=ATHENE2$="0"+RIGHT$(E2$,3)
8570 POKE TB,ASC(MID$(E2$,A,1)): TB=TB+1: NEXT
8580 POKE TB,ASC("2")
8590 RETURN
8600 REM Text und Betrag on Screen
8610 L=L+1:IFL(1)12THEN 8640
8615 IFDR=1THEN8640
8620 PRINT:PRINT" weiter?"
8630 GOSUB8100: L=0: CLS: GOSUB8200:PRINT
8640 P=10-LEN(B$)
8650 PRINTTAB(P)B$: PRINTTAB(12)T$
8660 RETURN
8700 REM Kontenspeicher auslesen
8705 M=0
8710 FOR A=1 TO 10
8720 B$=CHR$(PEEK(VA))
8725 IF B$="0"THEN M=M+1
8730 IF B$="0"AND A(7) AND M=0 THENB$=" "
8740 IFA=7)THENB$="":A=A+1:VA=VA-1
8750 B$=B$+B$
8760 VA=VA+1: NEXT
8770 RETURN
8800 REM Gleiche Prognerteile Eing/Ausg
8810 SCREENI,14:PRINT"Neue Eingabe? (J)"
8820 GOSUB8100
8830 IFE("J")THENKNM/2+.5GOTO2000,3000,4000,50
00
8840 GOSUB8400
8850 C2=S: REM Codexzchn
8860 GOSUB8500
8870 TB=UE+1
8900 CLS:GOSUB8200:PRINT
8905 GOSUB8320: RETURN
8908 REM Speicher umladen
8910 J=ZU
8920 FOR A=UB TO UB+8
8930 POKE J,PEEK(A): J=J+1
8940 NEXT
8950 RETURN
9000 REM Addieren: Vortrag + Su Eing = ZWsu
9010 REM Vortragsspei in ZW kopieren
9020 GOSUB 8900
9030 REM 1. Rechenpro aendern
9040 DOKE 3295,UC+7: REM Ende Su EingSpei
9045 DOKE 3304, ZU+7: REM Ende ZW-Speicher
9050 REM 2. ZW + Eing Su in ZW
9055 DOKE 4100,3294: X=USR(0)
9060 RETURN
9100 REM Subtrahieren: ZWsu - AusgSu =Saldo
9110 REM 1. Rechenpro f. subtr. aendern

```

```

9120 POKE 3323,144:POKE3338,214:POKE3338,144
9130 REM 2. Rechenpro f. Su AusgSpei
9140 DOKE 3295,VC+7
9150 REM 3. ZW - Ausg Su in ZW
9160 DOKE 4100,3294: X=USR(0)
9170 REM 4. Rechenpro f. add restaurieren
9180 POKE 3323,128:POKE3338,198:POKE3338,16
9190 RETURN
9200 REM Negative Ausgabe
9210 IF LEN(B#) < 9 THEN RETURN
9220 IF LEFT$(B#,1) < "9" THEN RETURN
9240 GOSUB 9300
9250 REM ZW - jew Kto in ZW (diff zu 1E6)
9252 GOSUB 9100
9253 REM PRINTDEEK(3384),DEEK(3295):INPUT
9254 REM ZW + 1(2,ZW) in ZW
9255 GOSUB 9350:DOKE3295,ZW+15:X=USR(0)
9260 REM ZW auslesen
9265 VA=ZW: B#="": GOSUB 8700
9270 B#=" " + " "
9280 RETURN
9300 REM ZWSpei max
9310 FOR J=ZW TO ZW+8
9320 POKE J,57: NEXT
9330 RETURN
9350 REM 2.ZWSpei 01
9360 FOR J=ZW+8 TO ZW+15
9370 POKE J,49: NEXT
9380 POKE ZW+15,49
9390 RETURN
9400 REM Gleiche Prognteile Ktostand -----

```

```

9410 GOSUB 9600
9420 GOSUB 8700: GOSUB 9200
9430 PRINT:IFOR=1THEN9435
9430 SCREEN 8
9435 PRINTTAB(5)*SaldoVortrag: DM ";B#
9440 RETURN
9450 GOSUB 8700
9455 PRINT* Summe Eingaenge: DM ";B#
9460 GOSUB 9000
9465 RETURN
9470 GOSUB 8700
9475 PRINT* Summe Ausgaenge: DM ";B#
9480 PRINTTAB(21)*-----"
9485 DOKE 3384,ZW+7: REM Speiend(Vortrag)/ZW
9490 GOSUB 9100
9495 RETURN
9500 REM Saldo in Saldospeicher
9505 VB=ZW: ZW=VS
9510 GOSUB 8900: ZW=VB
9515 GOSUB 9700: GOSUB 8700: GOSUB 9200
9520 PRINT* neuer Saldo: DM ";B#

```

```

9525 PRINTTAB(21)*-----"
9530 GOSUB 8100
9550 REM Rechenpro restaurieren
9555 POKE 3294,32: DOKE 3295,9212
9560 RETURN
9600 REM Rechenpro aendern
9610 POKE 3294,33: REM 0CDE LD HL xx xx
9620 DOKE 3295,VC+7: REM Speiend Subtrahend
9630 DOKE 3384,ZW+7: REM Speiend Minuend
9640 RETURN
9700 REM ZW in 2.ZW Kopieren
9710 FOR J=KS+136 TO J+7
9720 POKE J,PEEK(J-8): NEXT
9730 VC=VC+8
9740 GOSUB 9600: RETURN
9800 REM Drucken
9805 A=INP(5):IFA=40THEN9840
9810 PRINTTAB(5)*Bitte Drucker einschalten";
9820 A=INP(5)
9825 IFA=40THENCLS:GOTO 9840
9830 GOTO 9820
9840 DOKE 4100,3484:A=USR(0):REM Dru ein(0DAC)
9845 RESTORE 9860
9845 FORA=1TO7:READX:GOSUB 9900:NEXT
9850 RETURN
9860 DATA 27,78,27,76,48,49,53
9861 REM E,1815
9900 REM U-Druck
9910 OUT4,X:OUT5,253:OUT5,255:WAIT5,16,16
9920 RETURN

```

# RAM/EPROM

von KARL SCHULMEISTER

## ECB-RAM-EPROM-Karte

Die ECB-RAM-EPROM-Karte kann mit insgesamt 24Kb-Bausteinen bestückt werden, wobei folgende Einschränkungen gelten:

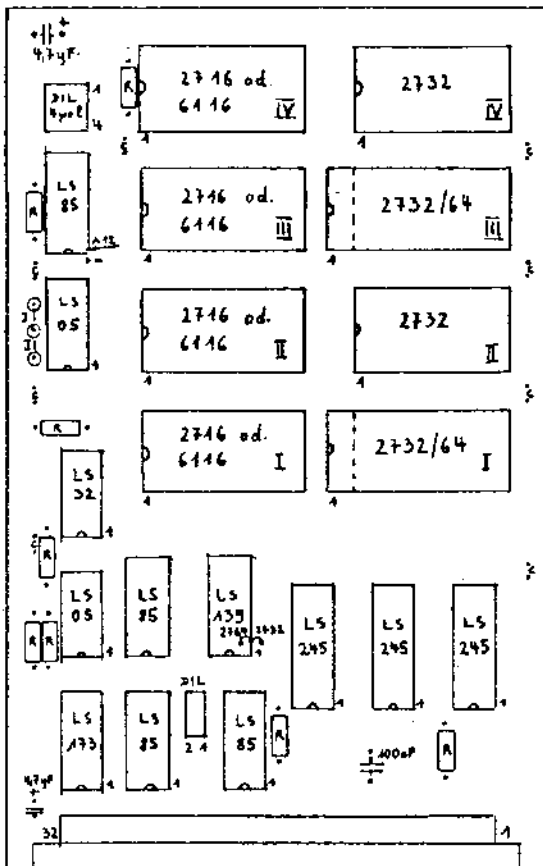
1. 16Kb EPROM, dekodiert in jeweils 4 Blöcke (0000-3FFF; 4000-7FFF; 8000-BFFF; C000-FFFF) mittels des 2-poligen DIL-Schalters, bestehend aus vier Stück 2732 oder zwei Stück 2764. Die Speicherart ist unter dem LS 139 mittels Drahtbrücke einzustellen, wobei der jeweils entsprechende Lötunkt neben Pin 2 mit letzterem zu verbinden ist.

2. 8Kb statisches RAM (6116) oder EPROM (2716), Adressbereich mittels vierpoligem DIL-Schalter einstellbar. Normalerweise würde ein 3-poliger Schalter genügen, es wurde jedoch zusätzlich das Adressbit 12 für die Dekodierung herangezogen und damit der 8Kb Bereich nochmals in 2x4 Kb geteilt. Dies ist z.B. notwendig, um den RAM-Speicher ab Adresse 1000 einsetzen zu können, da ja der Bereich von 0000-0FFF durch den Nascom verwaltet wird. Werden die gesamten 8 Kb benötigt, so ist der Schalter 1 in die Stellung "ON" zu bringen, das Adressbit 12 ist knapp vor dem Pin 1 des LS 85 zu unterbrechen (Bestückungsseite) und Pin 1 mit Masse (Pin 2) zu verbinden.

Die Karte weist noch eine Besonderheit für einen eventuell möglichen Einsatz mit CP/M auf: der gesamte auf ihr befindliche Speicherbereich kann durch den als Ausgabeport 3 eingesetzten LS 173 mit Bit 0 ausgeblendet werden (HIGH), LOW bzw "Reset" blendet wieder alles ein. Mit der Ausblendung verbunden ist natürlich auch eine Freigabe für einen RAM-Speicher im eingestellten Adressbereich, z.B. auf einer 64 Kb RAM-Karte. Der Jumper "RAM-EPROM" verhindert in der Stellung "RAM" ein Ausblenden des 8 Kb - Speicherbereiches und muß daher je nach Einsatz von 2716 oder 6116 in die jeweils benötigte Stellung gesetzt werden.

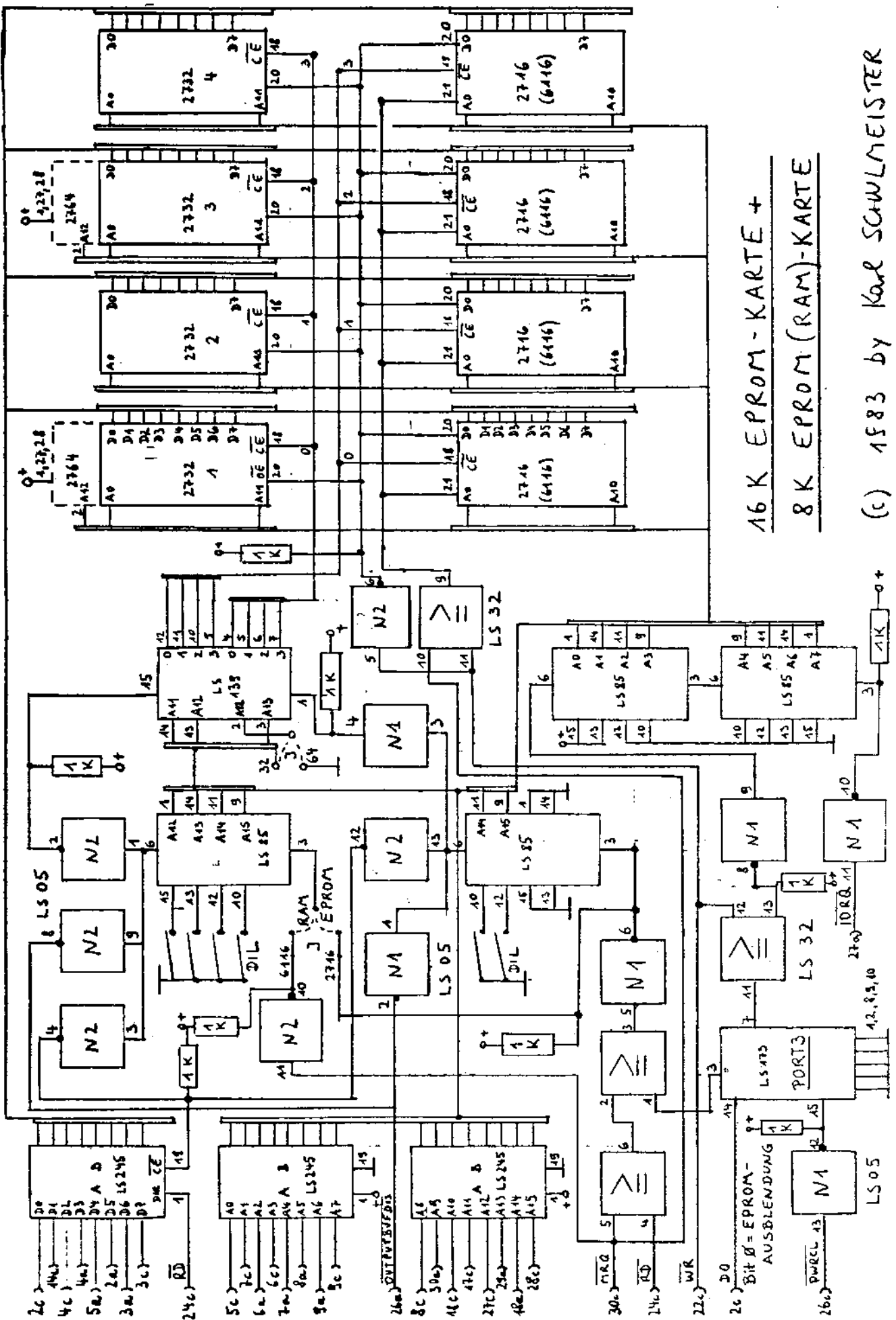
Sollte Port 3 nicht benötigt werden, entfällt der LS 173, Pin 1 des LS 32 muß dann mit Masse verbunden werden.

Mit der vorliegenden Karte ist es möglich, die gesamte Firmware des Nascom ab Adresse A000 zu verwenden und bei Bedarf auszublenden.



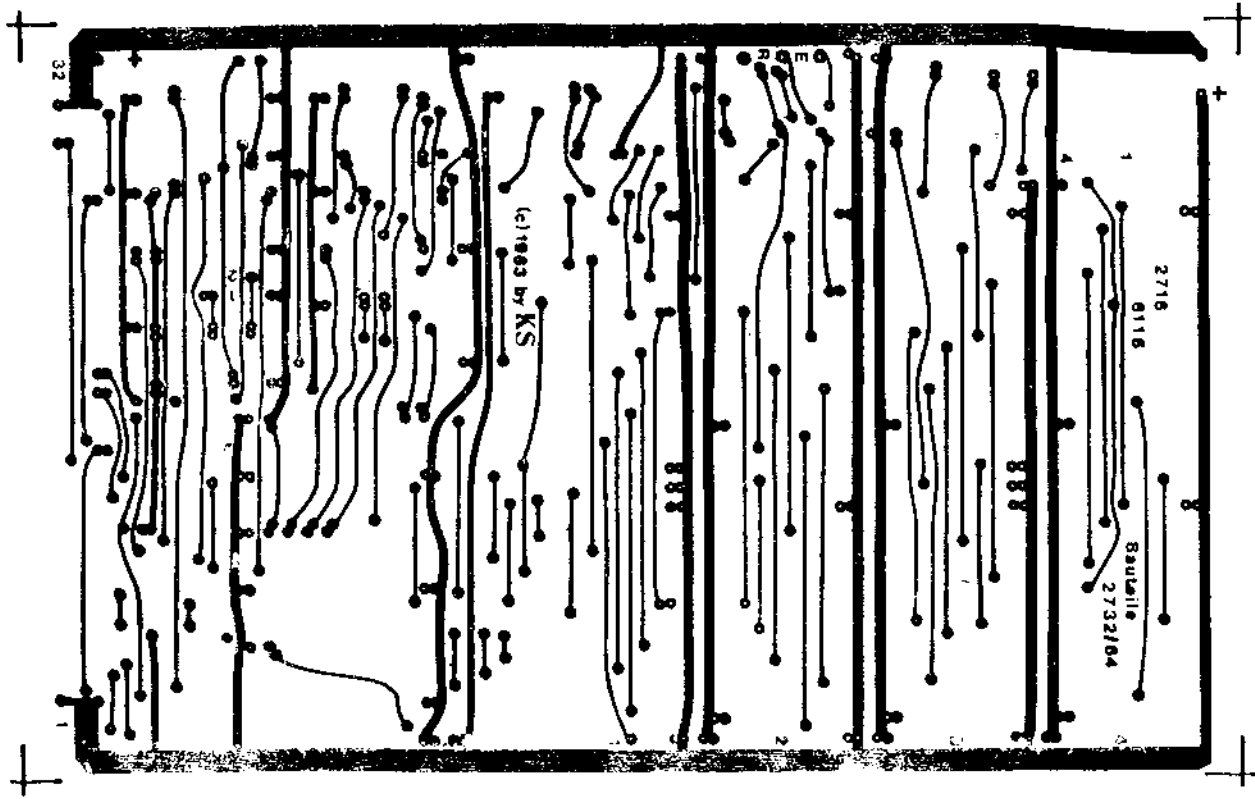
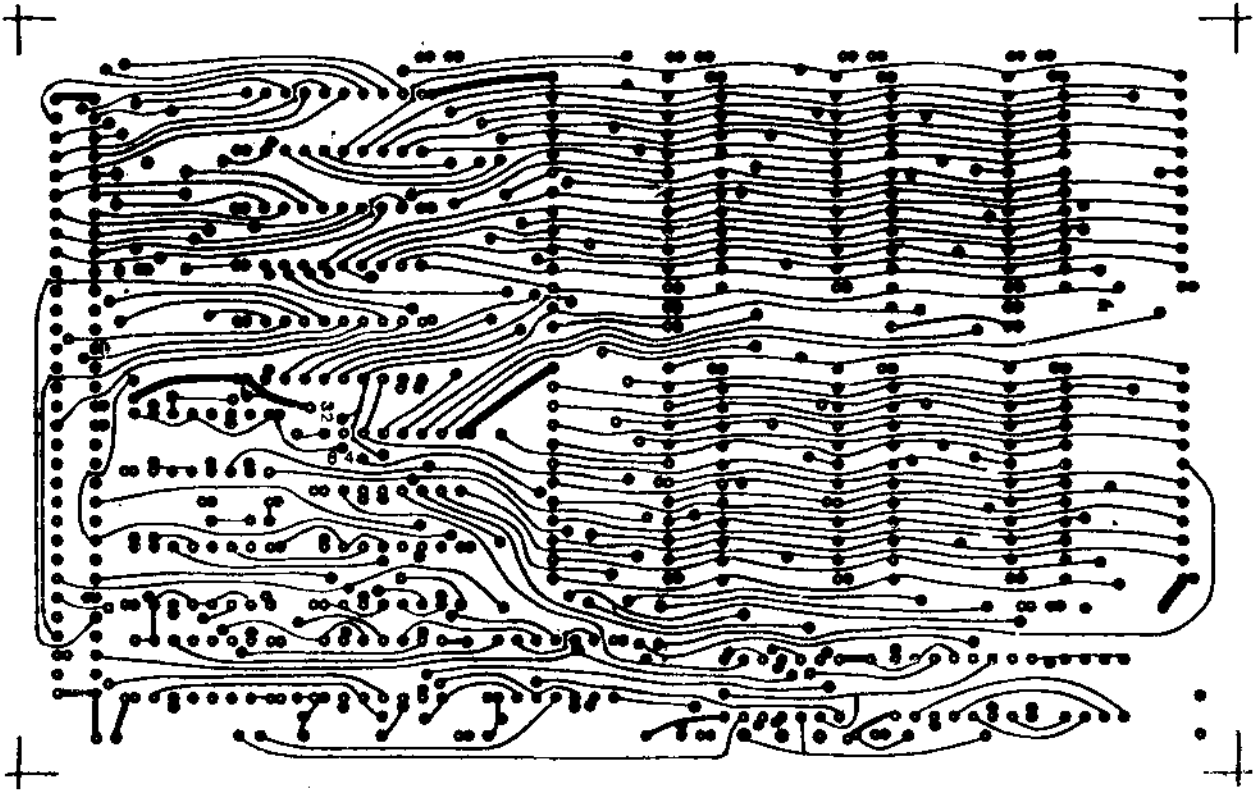
C = 10 nF  
R = 1KΩ





16 K EPROM-KARTE +  
8 K EPROM (RAM)-KARTE

(c) 1983 by Karl SCHULMEISTER



# Gemini Microcomputer

Vertriebs - GmbH

## SONDERANGEBOTE

solange der Vorrat reicht

RAM 'C' - Platine mit 64 KBytes, Bausatz	DM 450,--
EPROM 'B' - Platine, Bausatz	DM 330,--
SUPERMUM Erweiterung für NASCOM 1, Bausatz ohne Netzt.	DM 299,--
Paketpreis für NASCOM 1 Erweiterung, bestehend aus RAM 'C', EPROM 'B' und SUPERMUM	DM 998,--
EPROM - Programmiergerät für NASCOM oder GEMINI, programmiert 2708 und 2716 (5V), Bausatz einschl. Software (Betriebssystem angeben !)	DM 149,--
RTC Real Time Clock Bausatz, stellt über die PIO Urzeit und Datum zur Verfügung, Quarzgesteuert mit Akku einschl. Software	DM 149,--
BASIC ROM V 4.7 für NASCOM	DM 99,--
GRAFIK ROM für NASCOM	DM 47,--
Ersatzteile für NASCOM und GEMINI MULTIBOARD	
Z80 CPU	DM 9,--
Z80 A CPU	DM 9,50
Z80 PIO	DM 6,50
Z80 A PIO	DM 9,--
Z80 A CTC	DM 12,--
UART 6402	DM 25,--
DIL Plattform 16-pol.	DM 2,--
Tasten für Nascom 1/2 - Tastatur 10 Stück,	DM 60,--
Centronics - Stecker (Weibchen)	DM 19,--

Achtung! Jetzt besonders günstig!

Original Gemini Floppydiskstation mit 2 Laufwerken  
(jeweils 350 KBytes form.), anschlussfertig im Gehäuse,  
mit Netzteil und Kabel einschl. Original Gemini FDC,  
fertig aufgebaut und getestet

DM 2800,--

Vero- Frame Einschubrahmen für 80-Bus Platinen

DM 189,--

Alle Preise einschl. ges. Mehrwertsteuer, zuzüglich Porto/Verpackung,  
Lieferung nur gegen Nachnahme.

Bitte fordern Sie unser neuestes 80 - Bus Info an !

=====

Schluderstr.10 • 8000 München 19

Tel. 089/168595



# NASCOM

## NASCOM - Sonderangebote

NASCOM-C, der neue Maßstab für CP/M -und 80-BUS-Systeme !  
NASSYS-kompatibel und derzeit stärkstes CP/M-System !

- \* NASCOM-C mit Z80A CPU, 64KB RAM, MMU, 2xV24 und eine CENTRONICS-Schnittstelle, Videoteil, NUCLEOSYS.....DM 1.298,-
- \* NASCOM-C wie oben als Bausatz.....DM 998,-
- \* NASCOM-C Leerplatine mit Firmware & Dokumentation.DM 298,-
- \* NASCOM-C DMA-Floppy-Controller Option 5" oder 8"..DM 298,-
- \* Floppycontroller-Option für 5"+8" gemischt.....DM 348,-
- \* NASCOM-C 64KB-Erweiterung und Paritylogik.....DM 198,-
- \* Deutsche Tastatur für NASCOM-C (Cherry).....DM 198,-
- \* NASCOM-AVC Farbgrafik mit BASIC,-ASSEMBLER-und CP/M Softwareschnittstellen.....DM 798,-
- \* Hi-Res Farbmonitor für AVC (Zenith).....DM 1.998,-
- \* NASCOM-2a, NASCOM-2 mit 8KB CMOS-RAM ,ZEAP-Editor/Assembler und Microsoft-Basic in ROM als Bausatz.....DM 1.098,-
- \* 80-BUS, 4 Steckplätze, Busrahmen, Führungsleisten.DM 148,-
- \* CLD-Hardcontroller für hardsekt. Minidisketten..DM 498,-
- \* CLD-Softcontroller mit DMA, Echtzeituhr und Interface für Festplattenlaufwerke.....DM 998,-
- \* Softcontroller ohne DMA und CTC, Bausatz.....DM 698,-
- \* Softcontroller als Leerplatine + Firmware.....DM 198,-
- \* Minidiskettenlaufwerk BASF-6106, 200KB.....DM 498,-
- \* CLD-BANKED-Epromkarte für 16 Stück 2708 /16 /32, 2532 sowie 8KB ROMs in vier Banks, Bausatz.....DM 248,-
- \* Leerplatine Epromkarte mit Dok.....DM 148,-
- \* CLD-256KB-Ramkarte, Bausatz ohne DMA und Paritylogik mit 64KB RAM.....DM 698,-
- \* CLD-256KB-Ramkarte als Leerplatine mit Dok.....DM 148,-
- \* BLS-Pascal auf EPROM.....DM 298,-
- \* Page-Mode-Kit für LUCAS RAM-B Karte.....DM 79,-
- \* Grafik-Erweiterung für NASCOM-1 (ohne Grafgen.)...DM 98,-
- \* Grafik-Erweiterung für NASCOM-1 (mit Grafgen.)...DM 119,-
- \* CP/M 2.2 Betriebssystem mit ADM-31 Terminalemulator fuer AVC-Board, Screen-Editing auf CP/M Kommandoebene, Interface für Centronics-Drucker..DM 498,-
- \* BIOS-Anpassung bei Zusendung eines liz. CP/M.....DM 99,-

Alle Platinen mit Lötstopplack, vergoldeten Kontakten und Bestückungsdruck gefertigt und für alle NASCOM und GEMINI-Systeme verwendbar, alle Bausätze und neuen CLD-Platinen mit (gedrehten) Präzisionssockeln. FORTH -und BASIC EPROMS für NASCOM-C in Vorbereitung, desgleichen GSX für AVC & CP/M+

Preise inklusiv MwSt., exklusiv Versandkosten

LAMPSON Digitaltechnik  
Odenwaldstrasse 21-23 Tel.: 06152/56730  
6087 Büttelborn

10/11-83 SEITE 52