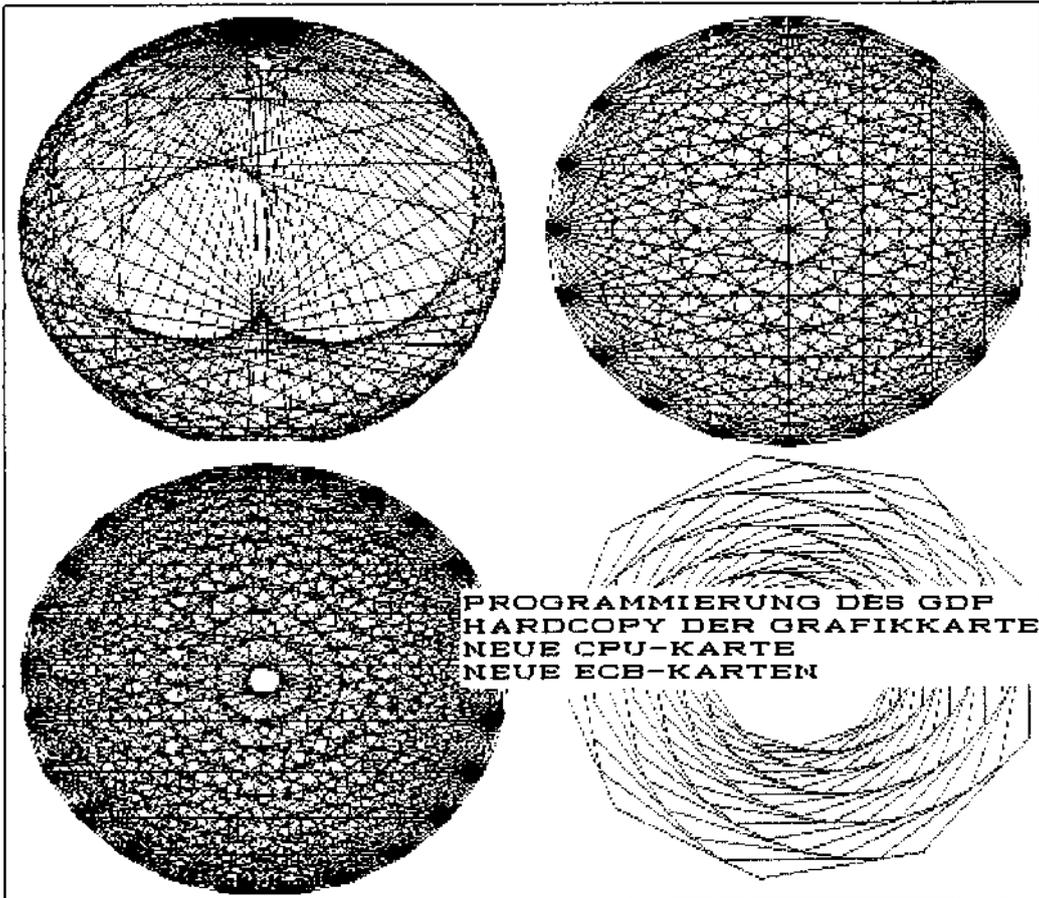


# 80-bus journal

Zeitschrift für NASCOM, GEMINI und andere  
Z80-Anwender

2. JAHRGANG\*APRIL/MAI/JUNI 1984\*AUSGABE 2



PROGRAMMIERUNG DES GDP  
HARDCOPY DER GRAFIKKARTE  
NEUE CPU-KARTE  
NEUE ECB-KARTEN

Ein Abonnement erhalten Sie für DM 60.- im Jahr

# 80-bus journal

## Intern

Liebe Leser,  
gleich zu Beginn das Wichtigste: wir brauchen dringend Beiträge! Ich weiß, daß die Mehrzahl der Leser augenblicklich damit beschäftigt ist, ECB-Karten aufzubauen und zu testen. Deshalb wurden meine bisherigen Hilferufe wohl auch nicht beachtet. Nun ist es aber soweit; ich habe meine Schublade völlig geleert und sogar das "Star Wars" veröffentlicht, das ich bisher als letzte eiserne Reserve ganz unten liegen hatte, da ich solche Spiele nicht allzuerne verbreite. Es ist nun eine ganze Urlaubswoche für die Fertigstellung dieses Heftes draufgegangen, weil ich auch noch eigene Programme aufbereiten mußte, die noch nicht für den Abdruck bestimmt waren, um das Heft noch etwas aufzufüllen.

In letzter Zeit ist es auch Sitte geworden, daß man die Autoren zum größten Teil direkt ansprechen muß, um Artikel zu bekommen. Früher kamen die Artikelfluten von alleine. Nun wird mir angst und bange, wenn ich an die Sommermonate denke, die erfahrungsgemäß nicht gerade zum Artikelschreiben ermutigen.

Zusammengefaßt läuft dies auf den Hilferuf hinaus: Bitte, liebe Leser, schickt Beiträge, sonst müßt Ihr auf die nächste Ausgabe sehr lange warten.

Vor allem die Disk-Benutzer: Schicken Sie doch schnell mal eine Diskette herüber mit Programmen, die auch andere Leute interessieren könnten.

Apropos Disketten: Ich habe mir Gedanken zur Zukunft des Journals gemacht und erbitte dazu Ihre Stellungnahme.

Da der Großteil eines Heftes aus Programmen besteht, die ja naturgemäß in den Rechner und nicht auf das Papier gehören, ist die Weitergabe von Programmen über ein schriftliches Medium ein Umweg, der auch noch Fehler verursacht.

So wäre meine Idee für die Zukunft (vor allem was CP/M angeht), die Programme nur noch auf Diskette zu vertreiben und nur noch Schaltpläne, Schaubilder, Grafik und vielleicht Programmbeschreibungen zu drucken. Wir würden damit die sehr hohen Druckkosten gewaltig reduzieren, die ja bei einer so geringen Auflage (wir haben im Augenblick etwa 160 Abonnenten) gewaltig zu Buche schlagen.

Nun höre ich schon die Cassettenbenutzer jammern. Haben Sie sich aber nicht auch schon geärgert, wenn Sie ein langes Listing eingetippt haben, während ein Diskettenbenutzer in einer Sekunde das Programm fehlerfrei einliest?

Man könnte natürlich auch einen Cassettendienst einrichten (wenn sich dazu ein Leser bereit fände); aber mit der Austauschbarkeit von Cassetten ist das so eine unsichere Sache, wie wir alle wissen. Da sollte sich der Bequemlichkeit wegen doch der eine oder andere einen Ruck geben und endlich auf Diskette umsteigen. In einem Jahr oder spätestens in zwei macht er es sowieso, wenn der Schullunge von nebenan mit seiner Diskettenstation protzt, und der alte Hase immer noch seine Cassetten durchnudelt. (Da hilft auch das Argument mit den überzüchteten Baudraten nichts, denn nicht die eigentliche Ladezeit ist ausschlaggebend, sondern die Zeit, die mit dem Suchen nach einer Datei verтан wird).

Diese Gedanken betreffen das nächste Jahr, sollten aber beizeiten diskutiert werden. Ich bin auf Ihre Meinung gespannt.

Die holländische NASCOM-Gruppe hat sich aufgelöst. Ich hoffe, daß sich einige Leute an unsere Gruppe anschließen werden. Etwas frischen Wind können wir schon gebrauchen, zumal der holländische Club einige fähige Leute aufzuweisen hat (ich erinnere an die Beiträge von B.H.Klaassen und Eric v.d. Vaart).

Auch in Dänemark existiert eine starke Gruppe von ca 500 Leuten, von denen die Hälfte schon CP/M benutzt. Auch hier bemühen wir uns, Kontakte aufzubauen.

Die Tasten zum Ausbau der NASCOM1 Tastatur sind meines Wissens in Deutschland nicht mehr erhältlich. Sowohl Lamson als auch Gemini erklären, daß sie keine mehr erhalten können. Auf meine Anfrage im letzten Heft

erhielt ich auch keine Antwort. Nun habe ich aber glücklicherweise eine Adresse ausfindig gemacht, bei der man original Licon- Tasten erhalten kann (Preis f 4,- pro Stück).

J. Visschers

Hengelo (Ov)

Holland

Ich möchte nochmals an die CP/M- Aktion erinnern (CP/M für 250,- incl. MWSt). Bisher haben wir ca 15 Interessenten, benötigt werden aber 50. Wir werden noch bis Anfang Juli warten und dann die Aktion abbrechen. Ich bemühe mich, für die bisherigen Interessenten aus anderer Quelle einen günstigen Preis für eine Sammelbestellung auszuhandeln. Sie werden frühzeitig informiert.

Im Augenblick arbeiten wir an einem BIOS für CP/M. Karl Schulmeister Jun. wird sich an die Bildschirmausgabe machen, während ich mir zunächst die Floppy-Routinen vorgenommen habe. Bis wir unser CP/M beziehen, ist das BIOS fertig.

In dieser Ausgabe finden Sie unter "Leserservice" einige neue Karten. Bitte melden Sie sich rechtzeitig, wenn Sie Interesse an einer durchkontaktierten Serie haben.

Einige dieser Karten stellen wir in dieser Ausgabe vor. Der Bestückungsplan zur "Erweiterungskarte" wird noch nachgeliefert.

Da einige neue Leser Interesse am Schaltplan zur 80-Zeichen- Karte haben, drucken wir den Artikel und den Plan aus Heft 5/82 nochmals ab.

Leider gibt es wieder eine neue Version des PHEAS (siehe FDC- Seite). Das Hex- Listing soll Ihnen die neue Version zugänglich machen. Besitzer der Disk0 können Ihre Diskette aber auch von mir kostenlos auf den neuesten Stand bringen lassen.

Peter Brendel hat seinen SPRITE- Editor vorgestellt. Wer würde ihn auf die Grafikkarte anpassen?

Mein Miniprogramm "Sinus" ist sehr langsam. Wer macht es schneller?

Damit sich der Kreis schließt, verabschiede ich mich mit der wiederholten Bitte um Beiträge und wünsche Ihnen viel Spaß mit dieser Ausgabe

Ihr Günter Böhm

P.S. Grüße von Gabi

## Impressum

HERAUSGEBER:

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

VERLAG:

Gabi Böhm Hard- und Software  
Ludwigshafener Str. 21 d  
7500 Karlsruhe

VERTRIEBSWEISE und BEZUGSPREIS:

Jahresabonnement In- und Ausland DM 60,-  
Es erscheinen 4 Doppelhefte pro Jahr und dazwischen aus aktuellem Anlaß Kurzmittellungen  
Es können aus organisatorischen Gründen 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 Postscheckkonto:  
Gabi Böhm PSchA Klrh

Wird das Abonnement nicht bis einen Monat vor Jahresende gekündigt, so verlängert es sich automatisch um ein weiteres Jahr.

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 und dem Journal. 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 werden.

BESONDERER LESERSERVICE:

Jeder Abonnent hat pro Ausgabe (auch in Kurzmittellungen) eine nichtkommerzielle Kleinanzeige frei.

Preise für kommerzielle Anzeigen sind beim Verlag zu erfragen.

## Inhalt

80-Bus Journal Intern	29 CPU- Platine	Karl Schulmeister
3 Impressum/ Inhalt	33 SLOUT zur ac I/O	Günter Böhm
4 Leserservice	34 CENTRONIX- Schnittstelle	Dieter Metzler
6 Neue ECB- Karten	36 Zahlenwandlung	Eberhard Horch
Zustandsanzeige	37 Postscheck	Klaus Nombaur
9 Erweiterungskarte	39 STERNE.PAS	Michael Bach
10 Programmierung der Grafik	40 Kannibalen	Gerhard Klement
16 PRINTGRAF Grafikausdruck	41 AdreAverw. Nachlese	Günter Böhm
18 Taschenrechner	42 Spielothek 2	Sauerbrey/Kastrup
19 Schwingkreisberechnung	44 Hex-Dez Wandlung	Günter Böhm
Leserbrief	45 Hardwaretips	Jörg Mittich
20 Floppy- Seite	46 64- Zeichen	Uwe Tepp
22 Filterberechnung	47 80 Zeichen- Karte	Zippel/Oberle
23 PHEAS 2.5	50 Hisoft Tips	Constantin Olbrich
USR(N)	READFROM	Günter Böhm
24 SPRITE- Editor	51 Nachträge	
26 Buchführung Nachtrag	Mitarbeiter	
27 STAR WARS	W.Mayer-Gurr	52 Lampson und Zerbe

# Leser- Service

Die gemeinsamen Bemühungen, unsere Systeme auf ECB-Bus umzustellen, haben viele Früchte getragen; und so können wir inzwischen auf eine stolze Anzahl von Karten verweisen, die von Journal-Lesern erstellt worden sind. In erster Linie ist da natürlich Kari Schulmeister und Gerhard Abmann zu danken, aber andere Leser ziehen nach.

Hätten wir den ECB-Boom unter unseren Lesern vorausgesehen, wäre eine Serienfertigung aller Schulmeister-Karten lohnend gewesen und hätte viel Mühe beim Bohren und manuellen Durchkontaktieren erspart. Ob jetzt noch genügend Interessenten übrig sind, um alte Karten - wie z.B. die attraktive 256K RAM-Karte - als Serie aufzulegen, bleibt abzuwarten, nachdem Herr. Abmann schon von ca. 30 Bestellern der ungebohrten Karte berichtet hat.

Hier kann auch die Leserfrage beantwortet werden, wieviel Bestellungen notwendig sind. Grundsätzlich lassen wir ab zehn Bestellungen eine Serie fertigen. Da jeder Besteller die Grundkosten mittragen muß, wird solch eine Karte relativ teuer (65.-), bei mehr Bestellungen geht der Preis natürlich runter. Die FDC- und Grafikkarte bleiben durch die teuren Proms leider an der oberen Preisgrenze.

Ich persönlich bin immer an einer durchkontaktierten Karte interessiert, da ich die Fummelei mit stumpfen Bohrern hasse, und die vervielfachte Lötarbeit bei komplexen Karten, das häufige Herausfallen der Durchkontaktierungen beim Löten der Rückseite und die ständige Angst, eine fehlerhafte Durchkontaktierung nur schwer wieder zu finden mir ein Greuel sind.

Da wir in unserer Gemeinschaft über die Möglichkeit haben, professionell gefertigte Karten machen zu lassen, was für eine Einzelperson mit begrenztem Taschengeld absolut unmöglich wäre, sollten wir diese Chance ausnutzen. Alleine dieser Vorteil scheint mir Grund genug, unsere Interessengemeinschaft auch in Zukunft zu erhalten.

Man könnte die Vorteile auch auf Sammeleinkäufe für relativ teure Bauteile ausdehnen; aber der Arbeitsaufwand ist zu groß, um das

alles aus einer Hand zu bewältigen. Ich habe diese Erfahrung wieder bei der Bestellung der Buchsenleisten und Schalter für die Busverlängerung und die Adapterkarte gemacht.

Vielleicht findet sich hier ein Leser, der Interesse daran hätte, günstige Bausätze für unsere Karten zusammenzustellen, sodaß wir auch hier einen Service anbieten könnten. Was meine Frau und mich betrifft, so sind wir mit den Platinen (und leider auch Proms) Jedenfalls ausgelastet.

Übrigens: für die 80-Bus Verlängerungskarte haben sich nur drei Interessenten gemeldet, für eine Serie weitaus zu wenig. Klaus Peter Schmidt hat sich bereit erklärt, für die Mitinteressenten die Platine zu ätzen. Falls Sie da noch schnell mit einsteigen wollen, bitte an folgende Adresse wenden (ich meine aber das 80-Bus oder "Riesenformat", um Mißverständnissen vorzubeugen):

Klaus Peter Schmidt

Edingen

Tel. [Redacted]

Aber nun zu dem, was als Leserservice schon vorliegt, oder durch Ihre Vorbestellung ins Leben gerufen wird:

Folgende Karten sind augenblicklich lieferbar (fertig durchkontaktiert und glanzverzinnt):

80x24 Zeichen Karte	DM 65.-
Floppy Controllert incl. 2 Proms	DM 70.-
Grafik 256x512 incl. 1 Prom	DM 65.-

(Preise incl. 14% MWSt, Porto und Verpackung. Sollte man bei Preisvergleichen beachten; denn oft werden hier tolle Bearbeitungskosten herbeigezaubert).

Folgende Karten werden augenblicklich in Kleinserie hergestellt (Qualität wie oben):

CPU mit PIO und CTC, Interruptfähig,  
2K EPROM u. 2K stat. RAM (auch als EMUF zu verwenden

Adapterkarte NASCOM (80-Bus) auf 2x ECB (macht den Anschluß von ECB-Karten ohne löten möglich)

Busverlängerung ECB-Bus (zum Testen von ECB-Karten außerhalb des 19"-Einschubs)

Die Preise der drei obigen Karten stehen noch nicht fest (ist von der Bestelleranzahl

abhängig, wird aber jedenfalls preiswerter als marktübliche Angebote).

Für folgende Karten liegt ein Layout vor. Bei entsprechendem Interesse werden wir eine durchkontaktierte Serie fertigen lassen:

Centronics-Druckeranschluß  
Bustermiierung (etwas ausgefeilter als mc)  
Buszustandsanzeige (mit 40 LEDs zur Kontrolle der Busignale)

5V/10A Regelnetzteil  
Netzteilkarte -5/-12/+12/+28V Je 1A  
Spannungswächter zur Sicherung der ECB-Karten

Erweiterungskarte zur Erzeugung von NMI und Mi-getriggertem Reset, WAIT-Erzeugung durch beliebige Signale. Hardwaremäßige Einzelschrittsteuerung zur Buskontrolle.

EPROM-Programmierer für alle EPROMs von 2708 bis 2764

24K RAM/EPROM-Karte

256K-RAM-Karte

48x16 Video-Karte

Kansas-City-Interface mit Tastaturanschluß

I/O-Karte mit 2 x PIO u. 1 x SIO

Sollten für obige Karten zu wenige Bestellungen eingehen, sodaß eine Serienfertigung nicht gerechtfertigt wäre, könnten die Karten geätzt aber ungebohrt zum Preis von DM 20,- (Selbstkosten) geliefert werden. (Eine durchkontaktierte Fertigung erspart aber viel Mühe und Fehlerquellen, so daß man den Mehrpreis gerne in Kauf nehmen sollte).

Folgende Schaltungen laufen bereits als Prototypen und werden im Augenblick für ECB-Format entflochten. Auch hier entscheiden Bestellungen über die mögliche Serienfertigung:

EPROM-Port, (schnelles Laden des Speichers mit Festprogrammen aus EPROMs)

Hardware-Uhr mit Akkupufferung

Spracherkennung

Interface für Video-Kamera

Für alle oben angebotenen Karten ist natürlich auch die Software vorhanden.

Inzwischen läuft die Floppy-Karte mit dem Floppy-Verwaltungsprogramm EMDOS. Es können damit Disketten im mc-Format gelesen und (CP/M-kompatibel) beschrieben werden.

Wir sind gerade dabei, ein BIOS für CP/M zu schreiben. Die Veröffentlichung soll aber nicht nur ein fertiges Konzept liefern, sondern auch detaillierte Anweisungen, wie man das BIOS an seine eigenen Bedürfnisse anpassen kann. Unserer Meinung nach sind die bisher darüber veröffentlichten Hinweise entweder zu kompliziert oder nicht ausreichend.

Dazu könnten wir CP/M 2.2 zum Traumpreis von DM 250,- incl. MWSt liefern; allerdings nur, wenn wir 50 Interessenten zusammenbringen. Ihre Bestellung entscheidet also mit, ob wir CP/M zu solch einem günstigen Preis anbieten können.

Es lohnt sich übrigens, ein lizenziertes CP/M zu besitzen, denn manche Software erhält man nur mit Vorlage der Lizenz.

Die Programme eines jeden Heftes (und noch einiges mehr, was wegen der Länge nicht gedruckt werden kann) sind übrigens jeweils auf Diskette zu beziehen (DM 15,- incl. MWSt, Porto und Verpackung).

Dazu ist immer noch unsere Diskette fuer DM 20,- lieferbar, die sämtliche Software zur Implementation von EMDOS und unserer FDC-Karte liefert.

Verstehen Sie die Kalkulation unserer Floppy-Preise richtig: Was am Verkauf einer Floppy eingenommen wird (abzüglich Steuer) ist ein notwendiges Polster fuer unsere hohen Druck- und Portokosten. Allerdings ein sehr duennes; denn viele Floppies verkaufen wir nicht.

Inzwischen sind 3 Floppys als Leserservice (Format in Heft 12/83 beschrieben) lieferbar, Sie enthalten folgende Files:

DISK0 (Systemdiskette)

EMDOS.ASSEMBLER PHEAS.ASSEMBLER

WBOOT.ASSEMBLER CBOOT.ASSEMBLER

NASGEN.ASSEMBLER FORMAT.ASSEMBLER

READTRK.ASSEMBLER STAT.ASSEMBLER

NASGEN.COMMAND READTRK.COMMAND

FORMAT.COMMAND STAT.COMMAND

SRLOUT.ASS BACKUP.ASS

BACKUP.COM

Jeweils die neuesten Versionen. Die Systemspuren enthalten WBOOT und EMDOS, sodaß mithilfe des abgedruckten CBOOT (Heft 12/83) direkt gebootet werden kann. Das Eintippen der übrigen Programme können Sie sich dann

sparen.

Die Diskette ist für DM 20,- (incl MWSt, Porto und Verpackung) erhältlich. Bestellung durch Überweisung auf unten stehendes Konto.

DISK1 Programme aus Heft 1/84 und ASSEMBLER-Programme, die wegen der Länge nicht abgedruckt werden konnten:

GRAFIK.BAS	GRAFCOPY.ASS
READIN.ASS	MONITOM.ASS
BREAKOUT.ASS	PACMAN.NAS
PACMAN.ASS	PACGRAF.ASS
PACMAN.DOC	VIERECK.ASS
MAEDCHEN.NAS	T4DRIVER.ASS
VIERECK.DOC	READIN.NAS
TEXTCASS.ASS	TOOLMOM.NAS

ca.120 KBytes, deren Abtippen Sie sich sparen können. DISK1 können Sie durch Überweisung von DM 15,- auf untenstehendes Konto erhalten.

Disk2

Inhalt dieser Ausgabe.

(+ vollständiges PHEAS V 2.5 und Adreßverwaltung V 5.3)

BEZIER.BAS  
PRNTGRAF.ASS  
SINUS.BAS  
RECHNER.BAS  
SCHWING.BAS  
PHEAS25.ASS  
FILTER.BAS  
SPRITE.NAS  
SPRITE.BAS  
ZAHLEN.BAS  
STERNE.PAS  
KANNIBAL.BAS  
SPIELE2.BAS  
ADRVERW.ASS  
HEXDEZ.BAS  
USR(N).ASS



Ob die Programme "Star Wars" (Gürr) und "Postscheck" (Mombaur) noch rechtzeitig übernommen werden können, ist zur Zeit der Drucklegung noch nicht sicher.

Auch Disk2 ist mit einer Überweisung von DM 15,- auf folgendes Konto zu bestellen:

██████████ Gabi Böhm PSchA Kirh

Kleinanzeige

Wegen Systemaufgabe zu verkaufen  
NASCOM2 im original Pult-Metallgehäuse  
NASSYS1, BASIC, Assembler, Disassembler,  
Toolkit  
PIO-Karte, RAM-Karte (32K), Netzteil  
komplett aufgebaut; alles original NAS Teil-  
lei vorh. Software gratis dazu.  
VB nur 1500 .-

Klaus Bott, ██████████

Tel. ██████████

(abends ██████████)

## Neue Karten

VON GERHARD ASSMANN

Die Buszustandsanzeige wurde bereits in unserem Leserservice erwähnt. Es handelt sich hierbei um eine Europakarte (ECB-BUS) mit 40 LEDs auf der Stirnseite, die den aktiven Zustand für folgende Signale anzeigen: D0-D7, A0-A19, Wait, BUSRQ, NMI, M1, RD, WR, MREQ, IORQ, BUSAK, RFSH, HALT. Die Anzeige ist abschaltbar.

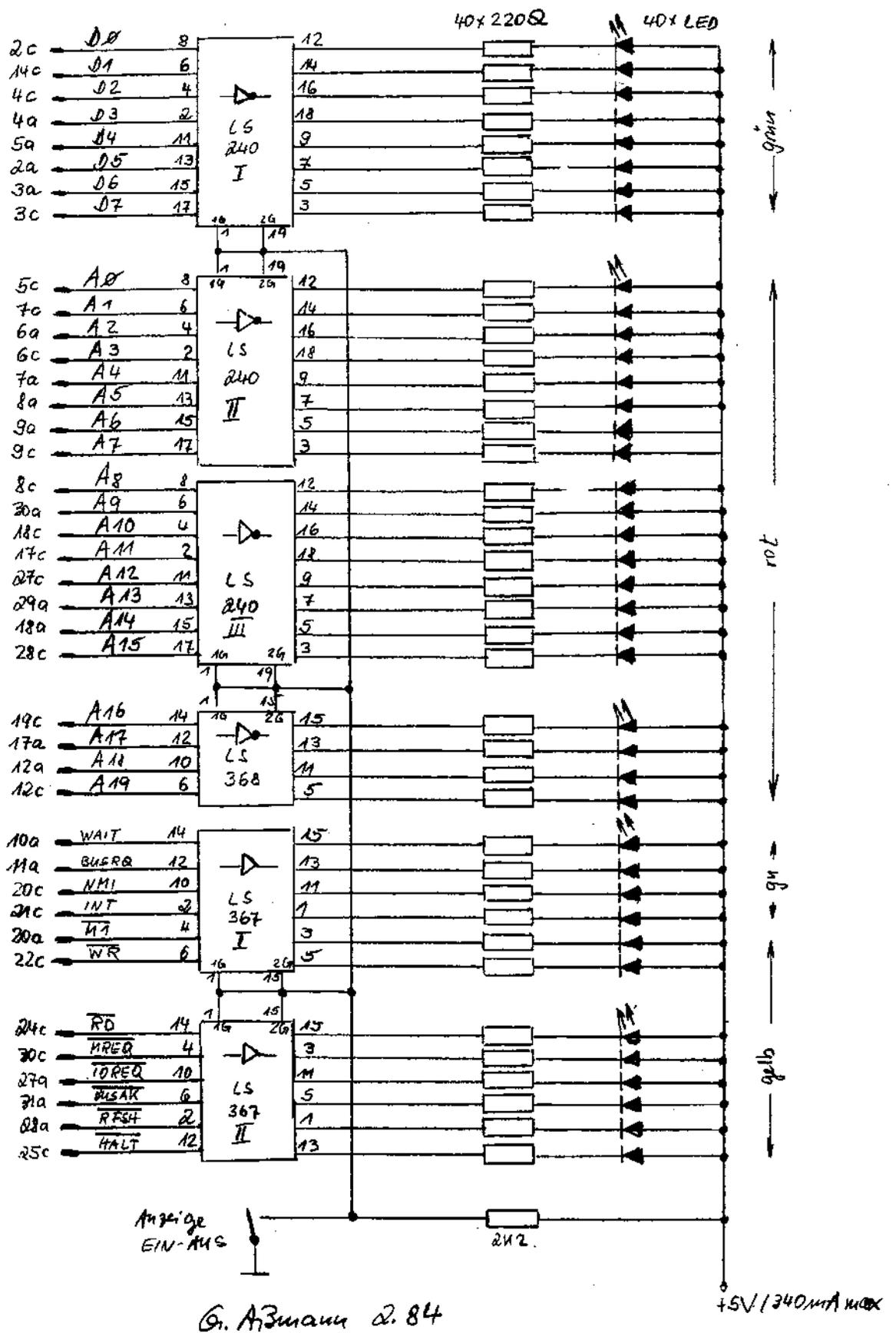
Der darauf folgende Schaltplan bezieht sich auf die Erweiterungskarte. Der Bestückungsplan wird im nächsten Heft nachgeliefert.

Es handelt sich dabei um eine Schaltung, die mittels Taste einen hardwaremäßigen NMI und einen M1-getriggerten RESET wie NASCOM2 erzeugt. Ein WAIT-Signal kann durch die Signale RD, WR, M1, MREQ, IORQ, BUSAK, RFSH oder INT erzeugt werden. Ebenso kann eine beliebige Kombination dieser Steuersignale dazu benutzt werden (durch 8 pol. DIL-Schalter einstellbar).

WAIT funktioniert wie im NASCOM-Journal 10/81 Seite 5 und 6 beschrieben.

Ferner besitzt die Karte einen Taktgenerator, mit DIL-Schalter auf 1, 2, 4, 8 MHz einstellbar mit einem Fan-Out von 50 zur zentralen Taktversorgung. Risetime bei Belastung mit 16 Steckplätzen ca 10 ns. Der Takt ist abschaltbar, und es können dann einzelne T-Cycles zur CPU gegeben werden mit einer STEP-Taste.

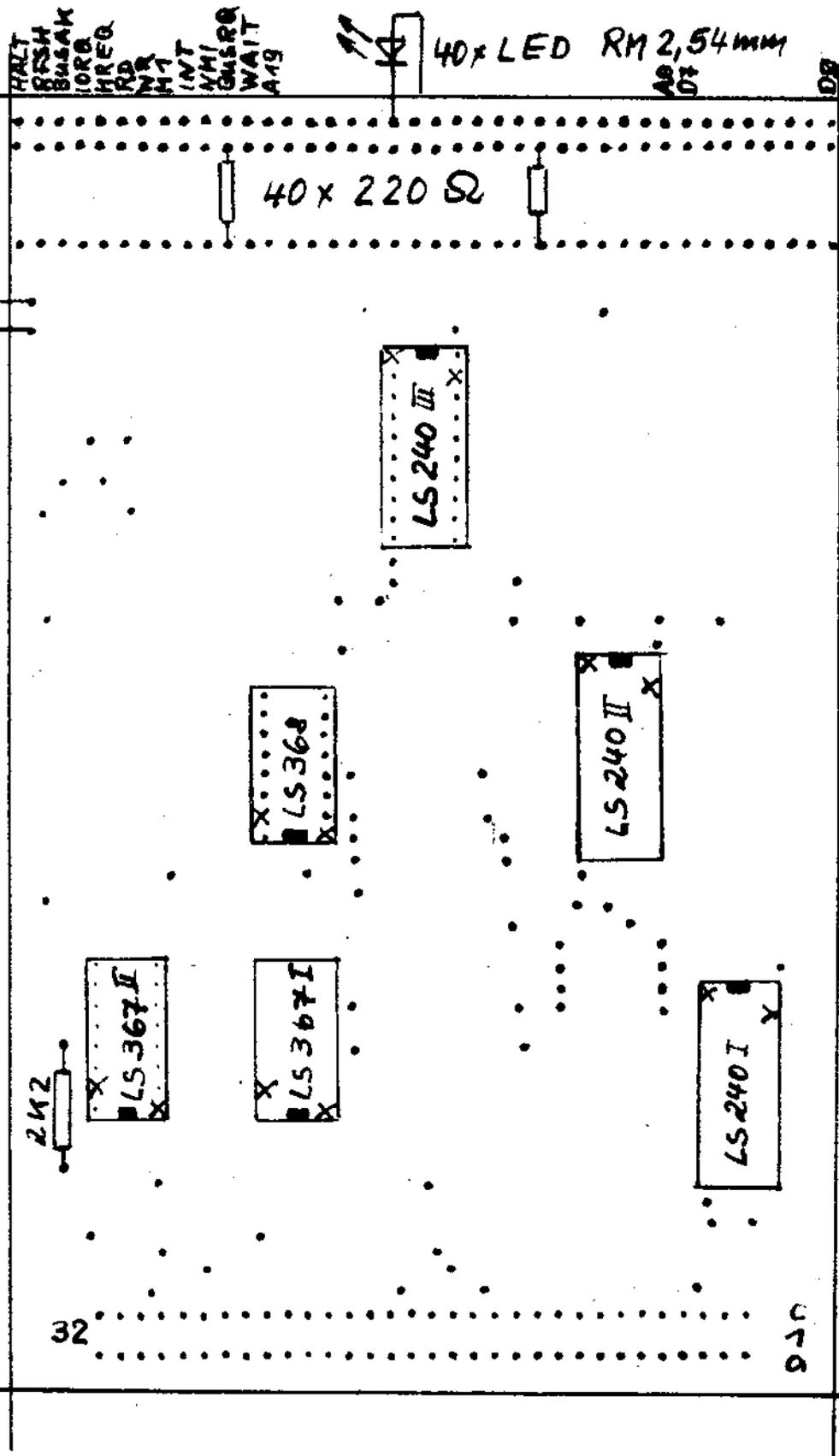
Beide Karten könnten bei entsprechender Nachfrage als durchkontaktierte Serie hergestellt werden.



Buszustandsanzeige

G. Abmann 7.84

Anzeige  
EIN-AUS



x Durchkontaktierung am IC-Stecker.  
Die LED's zeigen nicht die logischen Pegel des Buses an,  
sondern aktive Bussignale der CPU.



# Grafik mit GDP

von GÜNTER BÖHM

Meine Grafikkarte ist aufgebaut und funktioniert! Hinter diesem einfachen Satz steckt wieder eine schweißtreibende Arbeit, die eigentlich nicht nötig gewesen wäre; denn das Layout der Karte ist hervorragend. Zunächst wurden in der Stückliste wieder mal einige Teile übersehen. So fehlen folgende ICs mit Sockeln:

- 1 LS 122
- 1 LS 153
- 1 LS 154
- 1 LS 173

Das Warten auf die nachbestellten ICs war sehr unangenehm, nachdem die Karte ansonsten fertig bestückt vor mir lag.

Nach Eintreffen der ersehnten Chios gab es für mich deshalb kein Zögern mehr, und die Karte wurde sofort in freudiger Erwartung in den Bus geschoben. Die Umschaltung der Software auf Grafik mit

0 9F 8Ø funktionierte auch augenblicklich, aber der Bildschirm blieb dunkel. Dieser Zustand hielt auch einige Tage an. Diese Zeit möchte ich Ihnen nicht schildern, sie stand der Fehlersuche bei der FDC-Karte in nichts nach, nur waren meine Ergebnisse hier sehr überraschend: fünf unterbrochene Verbindungen an IC-Sockeln auf der Karte. Ich habe nun einige Karten aus der neuen Serie nachgemessen, konnte aber keine Fehler finden. Sollte ich hier nach so langer Lötterfahrung tatsächlich gleich fünfmal "Mist gebaut" haben? Ich kann es nicht glauben. Sicherheits halber gebe ich Ihnen die fehlerhaften Verbindungen an; möglicherweise findet man auf der einen oder anderen Karte den gleichen Fehler wieder:

- D5 Pin7 - D6 Pin7
- D5 Pin25 - D6 Pin14
- D3 Pin2 - D4 Pin3
- D9 Pin13 - Widerstand
- D24 Pin3 - D2Ø Pin13

Dann stellte ich noch eine Unverträglichkeit des Videoausgangs mit meinem Monitor fest: kurz nach dem ersehnten Erscheinen eines Grafik-Bildes brach die Helligkeit zusammen. Das Anlöten eines 39Ø Widerstandes zwischen Videoausgang und Masse brachte Abhilfe. Seither läuft die Grafik und reizt mich, die

tollsten Grafikprogramme auszuprobieren oder zu programmieren; hoffentlich setzen sich bald einige Leser daran, denn nach jahrelanger "Klötzchengrafik" ist der Bildschirm nun schon ein Erlebnis.

Die Demonstrationsprogramme von Herrn Klaassen sind sehr schön, Sie finden einige Beispiele in diesem Heft. Der Pacman läuft einwandfrei, und auch meine alten Videobilder habe ich nun schon auf dem Bildschirm betrachtet.

In c't 5/6-84 Seite 46 ff sind einige Vorschläge, die man verwenden könnte. Das "Moiree" davon habe ich schon getestet. mc 5/84 brachte ein Beispiel für gekrümmte Flächen (Seite 75). Dieses reizvolle Programm lief aber nach einer Modifikation nicht wie erwartet (es stellte seltsame Linien dar). Ich drucke meine veränderte Version hier ab, vielleicht findet jemand den Fehler (der ja auch im Programm selbst liegen kann). Leider bin ich kein Mathematiker; die Formeln sind für mich absolut nicht nachprüfbar.

```
Ø REM BEZIER FLÄCHE
1 REM VERS.Ø.Ø mc 5/84
5 DIM X(4,4),Y(4,4),Z(4,4)
6 DIMCX(33),CY(33)
9 PRINT"BEZIER SURFACE"
1Ø GOSUB1ØØ1:REM READ
13 PI=3.1416
15 INPUT"ROTATION X: ";AN:XA=AN*PI/18Ø
17 INPUT"ROTATION Y: ";AN:YA=AN*PI/18Ø
3Ø REM SCALING & TRANSLATION
33 KO=8Ø
35 KX=512/2;KY=256/2
4Ø REM INCREMENTS
44 INPUT"STEPS HORI: ";S
48 INPUT"STEPS VERT: ";R
6Ø REM INIT GRAPHICS
61 OUT 159,128: REM GRAPHIC ON
62 OUT 144,7 : REM RESET ALL REG.
63 OUT 145,3 : REM PEN DOWN, WRITE
7Ø REM BEZIER SURFACE
1Ø9 REM ZERO CONSTANT
11Ø CO=.ØØØØØØØØØØ
111 FORT=ØTOS:U=T/S
113 IF U=ØTHENU=CO
114 IFU=1THENU=U-CO
115 FORQ=ØTOR:F=Q/R
117 IFF=ØTHENF=CO
118 IF F=1THEN F=F-CO
122 PX=Ø:PY=PX:PZ=PX
```

```

133 FORI=0TON:FORL=0TOM
155 V=1:W=V
166 IFI=0 OR I=NTHEN211
177 FORJ=I+1TON:V=V*J:NEXTJ
199 FORJ=1TON-1:V=V/J:NEXTJ
211 V=V*U↑I*(1-U)↑(N-I)
222 IFL=0 OR L=MTHEN 233
224 FORJ=L+1TOM:W=W*J:NEXTJ
226 FOR J=1TOM-1:W=W/J:NEXTJ
233 W=W*F↑L*(1-F)↑(M-L)
244 PX=PX+X(I,L)*V*W
246 PY=PY+Y(I,L)*V*W
248 PZ=PZ+Z(I,L)*V*W
255 NEXTL:NEXTI
300 REM TRANSFORM ROT X & Y
322 TY=PY
333 Y1=TY*COS(XA)-PZ*SIN(XA)
344 Z1=TY*SIN(XA)+PZ*COS(XA)
366 X1=PX*COS(YA)+Z1*SIN(YA)
370 REM SCALE&TRANSLATE
388 X1=X1*KO+KX:Y1=Y1*KO+KY
400 REM DRAW LINES -----
411 IFQ=0THEN433
415 XS=XA:XE=X1
416 YS=YA:YE=Y1
417 GOSUB 2000
433 XA=X1:YA=Y1
440 REM DRAW NET -----
444 IFT=0THEN466
450 XS=CX(Q):XE=X1
451 YS=CY(Q):YE=Y1
452 GOSUB2000
466 CX(Q)=X1:CY(Q)=Y1
477 NEXTQ:NEXTT
555 INPUTA:OUT159,0:END:REM GRAPHICS OFF
1000 REM DATA
1001 N=2
1002 M=2
1111 DATA-1,-1,-1,-1,0,0,-1,1,0
1122 DATA0,-1,0,0,0,1,0,1,0
1133 DATA1,-1,0,1,0,0,1,1,0
1199 FORI=0TON:FORJ=0TOM
1211 READX(I,J),Y(I,J),Z(I,J)
1213 PRINTX(I,J),Y(I,J),Z(I,J)
1222 NEXTJ:NEXTI
1233 PRINT"KONTROLLPUNKTE
1333 RETURN
2000 OUT152,0:REM MSBX
2005 DX=XE-XS:X=0:IFDX<0THENX=2:REM DX VORZEICH
.
2010 DX=ABS(DX)
2015 IFXS=255THENOUT152,1:XS=XS-255
2020 OUT153,XS
2025 OUT149,DX

```

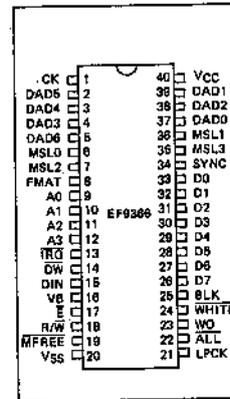
```

2030 DY=YE-YS:Y=0:IFDY<0THENY=4:REM DY VORZEICH
2035 OUT 155,YS
2040 DY=ABS(DY):OUT 151,DY
2045 BEFEHL=17+X+Y
2050 OUT144,BEFEHL
2055 RETURN

```

Zum Verständnis der Theorie verweise ich auf den Artikel in oben genannter mc.

Jedenfalls wird es in nächster Zeit eine Menge an Softwarearbeit für die Grafikkarte geben müssen. Um Ihnen bei der Entwicklung zu helfen, möchte ich im folgenden einiges an notwendiger Information über den GDP abdrucken, wie ja schon Karl Schulmeister in der letzten Ausgabe vorgeschlagen hat.



VSS Ground  
VCC +5V  
CK Master Clock für eine Bildfrequenz von 50Hz 1.7472 MHz  
FMAT bei 9366 unbenutzt  
WO Write Only high;kein Refresh, kein Display (Bild)  
SYNC Bild- und Zeilensynchronisation  
BLK Blanking (high bei Write oder Refresh)  
VB Vertical blanking (high)  
DAD 7 Adressen für den Refresh  
MSL Memory select für Pixel schreiben  
ALL wenn LOW, Zugriff auf alle Speicher (z.B.gesamt Löschen)  
DIN Display in  
DW Display memory write  
MFREE Memory free low, wenn auf Speicher zugegriffen werden kann (nach #F-Befehl)  
WHITE schreibt weiß zur Benutzung des Lichtgriffels  
LPCK Eingang ,steigende Flanke speichert

Bildschirmadresse in XLP und YLP  
 D0-D7 Datenbus  
 A0-A3 Adreßbus  
 R/W Read/Write- Signal  
 E Enable (Chip Select)  
 IRQ Interrupt-request (durch CTRL1 programmierbar)

## REGISTER

Die Grafikkarte ist für die Basisadresse 90H ausgelegt. Der GDP enthält nun eine Anzahl von Registern, die von dieser Basisadresse aus zu errechnen sind. (Register 0=90, Reg. 1=91 etc). Die Registeradressen finden Sie in der Tabelle.

ADDRESS REGISTER					REGISTER FUNCTIONS		Number of bits
Binary					Read R/W = 1	Write R/W = 0	
A3	A2	A1	A0	Hex			
0	0	0	0	0	STATUS	CMD	8
0	0	0	1	1	CTRL 1 (Write control and interrupt control)		7
0	0	1	0	2	CTRL 2 (Vector and symbol type control)		4
0	0	1	1	3	CSIZE (Character size)		8
0	1	0	0	4	Reserved		—
0	1	0	1	5	DELTA X		8
0	1	1	0	6	Reserved		—
0	1	1	1	7	DELTA Y		8
1	0	0	0	8	X MSBs		4
1	0	0	1	9	X LSBs		8
1	0	1	0	A	Y MSBs		4
1	0	1	1	B	Y LSBs		8
1	1	0	0	C	XLP (Light pen)	Reserved	7
1	1	0	1	D	YLP (Light pen)	Reserved	8
1	1	1	0	E	Reserved		—
1	1	1	1	F	Reserved		—

Hier möchte ich nun die Funktion der Register beschreiben:

### X und Y Register

Beide sind 12-Bit-Register, die die Position des nächsten Punktes angeben, der in den Bildschirmspeicher geschrieben wird. Beim Y-Register wird nur die untere Hälfte benutzt (8 Bit=256 Punkte), beim X-Register wird nur dann die obere Hälfte benutzt, wenn die untere (LSBX) 255 überschreitet. Dann wird MSBX auf 1 gesetzt.

### DELTA X und DELTA Y

Beide 8 Bit- Register geben den Abstand auf X- und Y-Achse beim Linienzeichnen an. Sie enthalten vorzeichenlose Ganzzahlen.

### CSIZE Register

Vergrößerungsfaktor beim Schreiben von Buch-

staben und Symbolen. Das MSB des 8-Bit-Registers gibt den Faktor für X, das LSB den für Y an (also beide jeweils von 0 bis F).

### CTRL1 Register

Alle 7 Bits dieses Registers haben eine gesonderte Bedeutung:

Bit0

Wenn Low ist kein Schreiben möglich

Bit1

LOW = löschen HIGH = schreiben

Bit2

LOW = normales Schreiben

HIGH = schnelles Schreiben ohne Bildschirm

Bit3

LOW = alle 12 Bits von X und Y sind signifikant. Es wird ein Bereich von 4096x 4096 Punkten verwendet (aber natürlich nicht abgebildet)

Bit4

LOW = unterdrückt den Interrupt, der durch den Lichtgriffel ausgelöst wird

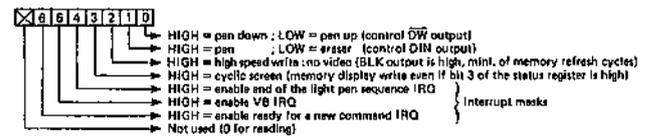
Bit5

LOW = unterdrückt den Interrupt beim vertikalen blanking

Bit6

LOW = unterdrückt den Interrupt, der anzeigt, daß das System für einen neuen Befehl bereit ist.

CONTROL REGISTER 1 (Read/Write)



### CTRL2 Register

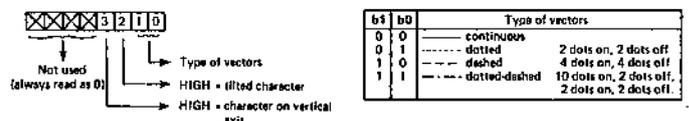
Durch dieses 4-Bit- Register werden die Parameter beim Schreiben von Vektoren und Buchstaben festgelegt:

Bit0,1 4 Linientypen (siehe Tabelle)

Bit2 LOW=gerade HIGH=schräg

Bit3 LOW=waagrecht HIGH=senkrecht

CONTROL REGISTER 2 (Read/Write)



## CMD COMMAND Register

Ins Kommandoregister kann nur geschrieben werden. Jeder Befehl (siehe Tabelle) löst eine Aktion des GDP aus. Durch einige Befehle können auch Bits in anderen Registern geändert werden.

COMMAND REGISTER									
67	66	65	64	63	62	61	60	59	58
0	0	0	0	0	0	0	0	0	0
57	56	55	54	53	52	51	50	49	48
1	1	1	1	1	1	1	1	1	1
47	46	45	44	43	42	41	40	39	38
0	0	0	0	0	0	0	0	0	0
37	36	35	34	33	32	31	30	29	28
1	1	1	1	1	1	1	1	1	1
27	26	25	24	23	22	21	20	19	18
0	0	0	0	0	0	0	0	0	0
17	16	15	14	13	12	11	10	9	8
1	1	1	1	1	1	1	1	1	1
7	6	5	4	3	2	1	0	0	0
0	0	0	0	0	0	0	0	0	0
0000	0	Set bit 1 of CTRL1 : Pen selection							
0001	1	Clear bit 1 of CTRL1 : Eraser selection							
0010	2	Set bit 0 of CTRL1 : Pen/Eraser down selection							
0011	3	Clear bit 0 of CTRL1 : Pen/Eraser up selection							
0100	4	Clear screen							
0101	5	X and Y registers reset to 0							
0110	6	X and Y reset to 0 and clear screen							
0111	7	Clear screen, set CSIZE to code "minimize" All other registers reset to 0 (except XLP, YLP)							
1000	8	Light-pen initialization (WRITE forced low)							
1001	9	Light-pen initialization							
1010	A	8 x 8 block drawing (size according to CSIZE)							
1011	B	4 x 4 block drawing (size according to CSIZE)							
1100	C	Screen scanning : Pen or Eraser as defined by CTRL1							
1101	D	X register reset to 0							
1110	E	Y register reset to 0							
1111	F	Direct image memory access (reset) for the next free cycle.							

SPACE	0	R	P	-	P
1	1	A	O	E	Q
2	2	B	R	b	t
3	3	C	R	c	t
4	4	D	Y	d	t
5	5	E	U	e	u
6	6	F	V	f	v
7	7	G	W	g	w
8	8	H	X	h	x
9	9	I	Y	i	y
A	A	J	Z	j	z
B	B	K	A	k	a
C	C	L	I	l	i
D	D	M	J	m	j
E	E	N	K	n	k
F	F	O	L	o	l

57	56	55	54	53	52	51	50
1	1	1	1	1	1	1	1
ΔX	ΔX	ΔY	ΔY	Direction	Direction	Direction	Direction
0	0	0	0	0	0	0	0
1	1	1	1	1	1	1	1

ΔX or ΔY	Vector length
0 0	0 step
0 1	1 step
1 0	2 steps
1 1	3 steps

## XLP und YLP Register

Diese Register enthalten die Bildschirmadresse, die durch die erste steigende Flanke am Lichtgriffelanschluß LPCK ermittelt wird.

Eine detaillierte Beschreibung des internen Aufbaus wollen wir hier nicht geben, ebenso wenig den genauen Signalverlauf oder die elektrischen Grenzwerte. Diese Angaben müssen Sie einem Datenblatt der Firma Thomson entnehmen.

Diese Beschreibung soll aber die notwendigen Informationen zum Test der Grafikkarte liefern (der hoffentlich nicht nötig ist) und vor allem bei der Programmierung des GDP behilflich sein.

## Programmierung des GDP

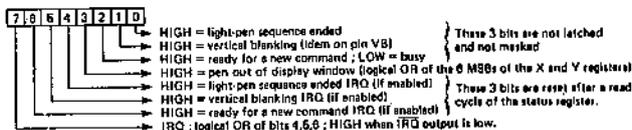
Zur Ergänzung der etwas kurz gehaltenen Beschreibung empfehle ich das Studium der Listings in Heft 1/84 und in dieser Ausgabe.

## STATUS Register

Dieses Register kann nur gelesen werden. Es gibt folgende Zustände des GDP an:

- Bit0 LOW = ein Lichtgriffel-Durchgang wird gerade ausgeführt
- Bit1 HIGH während des vertical blanking
- Bit2 LOW = es wird gerade ein Befehl ausgeführt  
HIGH= Ready
- Bit3 HIGH=X- und Y-Register außerhalb des Bildschirmfensters
- Bit4 HIGH=Lichtgriffeldurchgang mit Int. beendet
- Bit5 HIGH=vertical blanking interr.erzeugt
- Bit6 HIGH=Interr.durch Beendigung eines Befehls ausgelöst
- Bit7 HIGH=ein Interrupt wurde ausgelöst

STATUS REGISTER (Read only)



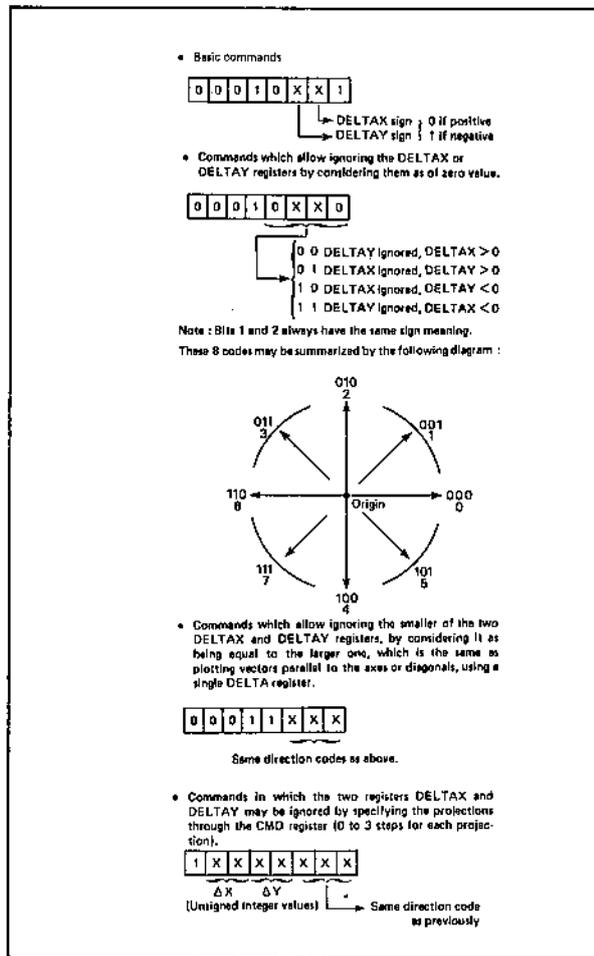
## Plotten von Vektoren

Die Ausgangskordinaten einer Linie werden durch Register 98H bis 9BH (X und Y) festgelegt. Die Register 95H und 97H (DELTA, DELTAY) bestimmen den Abstand auf den Koordinaten-Achsen. Das Vorzeichen von DELTA und DELTAY wird durch Bit1 und Bit2 im Plotbefehl definiert (siehe Tabelle). Die X- und Y-Register zeigen nach dem Zeichenvorgang auf das Ende der Linie, sodaß durch einfaches Ändern der DELTA- Werte kontinuierliche Linienzüge möglich sind. Das Löschen einer Linie geschieht durch nochmaliges Plotten der Linie, wobei Bit1 in CNTRL1 rückgesetzt werden muß (eraser). Durch Kombinationen von Bit0 und Bit1 in CNTRL2 kann beim Linienzeichnen zwischen 4 verschiedenen Linienarten unterschieden werden. (Tabelle)

In oben beschriebener Weise wird mit den "basic commands" 11H bis 17H (nur die ungeraden Werte, also stets Bit0 gesetzt) gearbeitet, indem man sie jeweils ins Command-Register 90H schreibt.

Es gibt noch drei andere Arten von Plotbefehlen, die schnelle Richtungsänderungen bei kurzen Vektoren erlauben. Hier eine Zusammenfassung; die Tabellen sollten zur Demon-

stration ausprobiert werden:  
 11-17H (ungerade) Basic commands  
 18-16H (gerade) DELTA nur auf einer Achse  
 18-1FH 8 Richtungen  
 80-FFH 0 bis 3 Einzelschritte



Die verbleibenden Werte 20 bis 7FH werden zum schreiben von

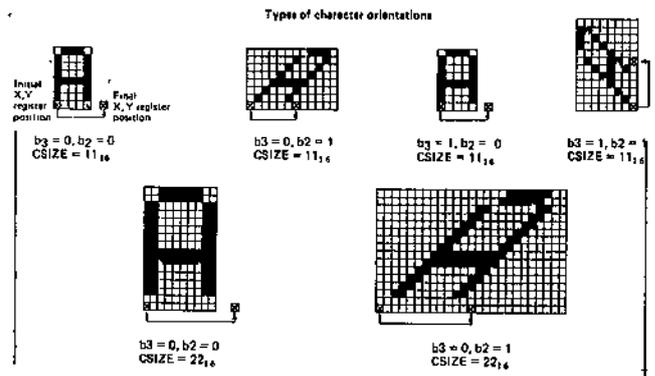
Buchstaben und Symbole(n)

verwendet und entsprechen dem normalen ASCII-Code.

Bit2 high im CNTRL2 Register stellt die Buchstaben schräg dar; Bit3 high schreibt auf der Senkrechten.

Die Größe der Buchstaben wird im CSIZE Register 93H vorgegeben. 01H ist die kleinste, 00 die größte Ausdehnung in einer Richtung.

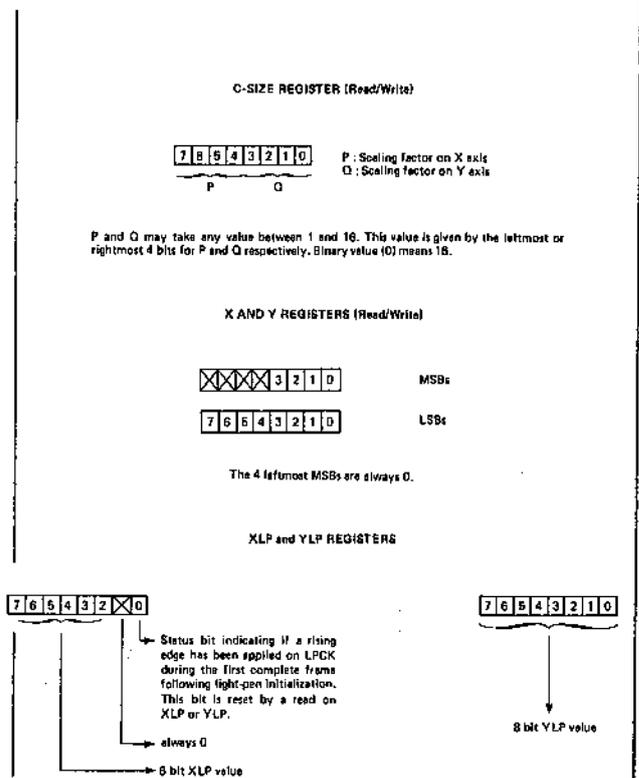
Nach dem Schreiben eines Buchstabens zeigt das X-Register auf zwei Einheiten rechts neben dem Buchstaben, sodaß ohne erneute



Programmierung weitergeschrieben werden kann.

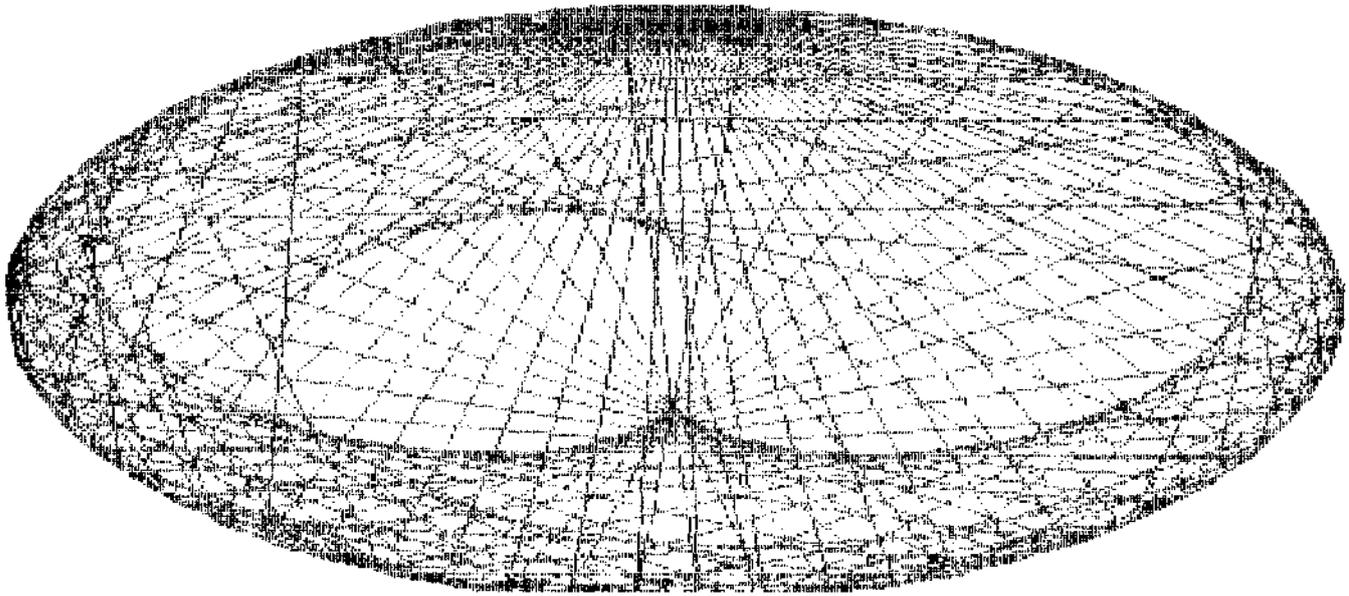
Der Code 0AH und 0BH schreibt einen Block von 5x8 bzw 4x4, ebenfalls abhängig von CSIZE. Mit CNTRL1 im ERASE-Modus können damit leicht Buchstaben gelöscht werden.

Dies sollte genügen, um einmal erste Programmerversuche mit dem GDP zu machen. Die Praxis wird sicher Fragen aufwerfen, die wir hoffentlich im nächsten Heft klären können. Projekte, die mit der Grafik vielleicht gemeinsam gelöst werden könnten, gibt es einige. So z.B. eine Schachgrafik (für unser Sargon) oder ein Programm zum Plotten von Platinen-Layouts. Letzteres wäre besonders für uns Dauerlötter eine sehr wünschenswerte Sache.



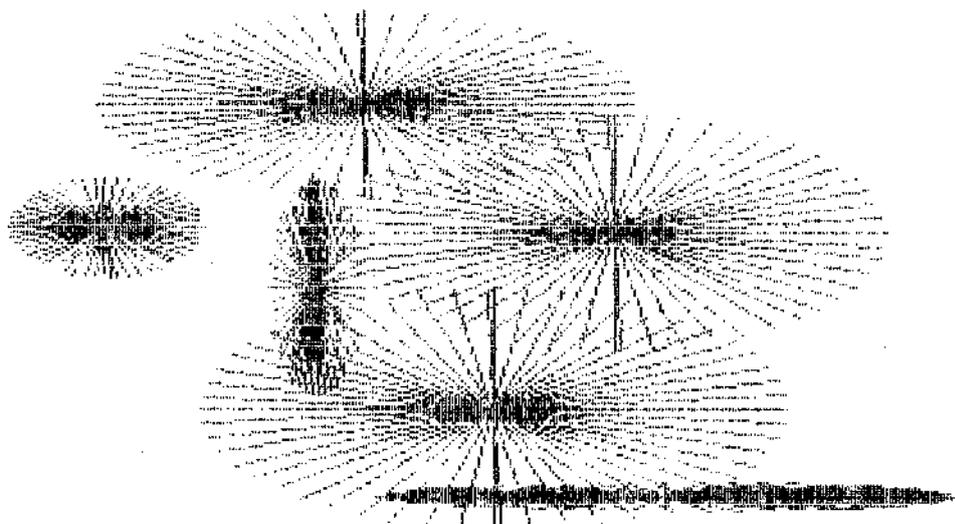
ELLIPSE I

Page 14



ELLIPSE II

Page 15



# PRINTGRAF

VON GÜNTER BÖHM

Das folgende Programm ermöglicht den Ausdruck von Grafikseiten der EF 9366 Karte mit einem Drucker.

Es ist für den TANDY Lineprinter VI ausgelegt, kann aber leicht für jeden anderen grafikfähigen Drucker angepaßt werden.

Der Programmteil INIT, initialisiert die mc IN/OUT- Karte, die SIOA für die serielle Druckerausgabe benutzt und PIOA Bit7 für die Abfrage des BUSY- Signals. NASSYS wird in diesem Unterprogramm für die Benutzung der mc- Karte modifiziert. Dieses Programm ist bei der Verwendung des UART nicht notwendig.

Der Programmteil DRUCK muß bei den meisten Druckern geändert werden, da sie im Gegensatz zum TANDY mit 8 Nadeln drucken.

Z1010 setzt den Drucker in den Grafik-Modus

Z1040 Faktor 32 für 8 Nadeln

Z1140 Differenz=448

Z1250 8 Nadeln

Z1400 entfällt

Die direkte Ausgabe an den Drucker PRINT (Z1450 ff) muß natürlich auch angepaßt werden.

## ZEAP Z80 Assembler - Source Listing

```

0010 ;PRNTGRAF mit EF9366
0020 ;Printerausdruck einer Grafikseite
0030 ;nach Routinen von G.Böhm Journal 1/83
0040 ;und Klaassen/Schulm. Journal 1/84
0050 ;
0060 ;G.Böhm Karlsruher Ver. 0.0 9.6.84
0070 ;-----
0080          ORG #C000
0090 ;
00A0 0090  0100 GRA  EQU #90  ;graphic base port
00B0 4000  0110 DFIELD EQU #4000 ;begin of data field
0120 ;
00C0 C3E3C0 0121 START JP  INIT
0130 ;
00D3 AF  0140 EXIT  XOR  A
00E4 D39F  0150  OUT  (GRA+15),A ;to normal video

00F6 DF5B  0160  DEFW #5BDF  ;return to NAS-S
YS
0170 ;
0180 ;----- STORE PICTURE -----
0190 ;
008  CD52C0 0200 STORE CALL SIDE
009  210040 0210 LD  HL,DFIELD
00E  3E80  0220 LD  A,#80
00F  D39F  0230 OUT  (GRA+15),A
012  CD4BC0 0240 CALL READY
0015  3E05  0250 LD  A,5
0017  D390  0260 OUT  (GRA),A ;reset X/Y-Registers

```

```

C019 0EFF  0270 LD  C,255 ;set Y-pointer
C01B 110000 0280 LD  DE,0 ;clear X-pointer
C01E CD4BC0 0290 STOR1 CALL READY
C021 7A  0300 LD  A,D
C022 D398  0310 OUT  (GRA+8),A ;out MSB X-pointe
r
C024 7B  0320 LD  A,E-
C025 D399  0330 OUT  (GRA+9),A ;out LSB X-pointe
r
C027 79  0340 LD  A,C
C028 D39B  0350 OUT  (GRA+11),A ;load LSB Y-poin
ter
C02A 3E0F  0360 LD  A,15 ;ask for memory
C02C D390  0370 OUT  (GRA),A ;free cycle
C02E CD4BC0 0380 CALL READY
C031 DB9F  0390 IN  A,(GRA+15) ;read graphics b
yte
C033 77  0400 LD  (HL),A ;store in Nascom
C034 23  0410 INC  HL
C035 E5  0420 PUSH HL
C036 210000 0430 LD  HL,8
C039 19  0440 ADD  HL,DE
C03A EB  0450 EX  DE,HL ;new graphics X-poin
ter
C03B E1  0460 POP  HL
C03C 7A  0470 LD  A,D
C03D FB02  0480 CP  2 ;2 means completed
C03F 20DD  0490 JR  NZ,STOR1 ;one line
C041 110000 0500 LD  DE,0 ;clear X-pointer
C044 0D  0510 DEC  C ;and decrement Y-point
er
C045 AF  0520 XOR  A
C046 B1  0530 OR  C ;test reg. C
C047 2841  0540 JR  Z DRUCK
C049 18D3  0550 JR  STOR1
0560 ;-----
C04B DB90  0570 READY IN  A,(GRA) ;read GDC status
C04D CB57  0580 BIT  2,A ;low-busy
C04F 28FA  0590 JR  Z,READY ;loop until ready
C051 C9  0600 RET
0610 ;-----
C052 EF  0620 SIDE RST #28
C053 0D0D  0630 DEFW 0D0DH
C055 53696465 0640 DEFM /Side 0, 1, 2 or 3?/
20302C20
312C2032
206F7220
333F
C067 0D00  0650 DEFW 0DH
C069 DF62  0660 IN  DEFW #62DF ;scan for an input
C06B 30FC  0670 JR  NC,IN
C06D D630  0680 SUB  30H
C06F 2815  0690 JR  Z,SIDE0
C071 3D  0700 DEC  A
C072 280E  0710 JR  Z,SIDE1
C074 3D  0720 DEC  A
C075 2807  0730 JR  Z,SIDE2
C077 3D  0740 DEC  A
C078 20EF  0750 JR  NZ,IN
C07A 3E0F  0760 LD  A,0FH
C07C 1809  0770 JR  SOUT
C07E 3E0A  0780 SIDE2 LD  A,0AH
C080 1805  0790 JR  SOUT
C082 3E05  0800 SIDE1 LD  A,5
C084 1801  0810 JR  SOUT
C086 AF  0820 SIDE0 XOR  A
C087 D39E  0830 SOUT OUT  (GRA+14),A
C089 C9  0840 RET
1000 ;-----
C08A 3E12  1010 DRUCK LD  A,18 ;Grafik Code
C08C CDD6C0 1020 CALL PRINT
C08F 21C03F 1030 LD  HL,DFIELD-64
C092 0624  1040 LD  B,36 ;Faktor f. 256 Lines/7
Nadeln
C094 0E40  1050 IN LD  C,64 ;Bytes per line
C096 3E0D  1060 LD  A,#D ;CRLF
C098 CDD6C0 1070 CALL PRINT

```

```

C09B C5      1080 NXTBYT PUSH BC
C09C CDADC0  1090 CALL GETBYT
C09F C1      1100 POP BC
C0A0 23      0110 INC HL
C0A1 0D      1120 DEC C
C0A2 20F7    1130 JR NZ NXTBYT
C0A4 118001  1140 LD DE,384 ;Diff.next line-bloc
k
C0A7 19      1150 ADD HL,DE
C0A8 10EA    1160 DJNZ NXTLIN
C0AA C303C0  1170 JP EXIT
1180 ;-----
C0AD 110000  1190 GETBYT LD DE,0
C0B0 14      1200 NXTBIT INC D ;Bitnumber
C0B1 3E08    1210 LD A,8
C0B3 BA      1220 CP D
C0B4 D8      1230 RET C
C0B5 E5      1240 PUSH HL
C0B6 0E07    1250 GETBIT LD C,7 ;Drucknadeln
C0B8 42      1260 BITNUM LD B,D
C0B9 D5      1270 PUSH DE
C0BA 114000  1280 LD DE,64 ;Differenz next line
C0BD 19      1290 ADD HL,DE
C0BE D1      1300 POP DE
C0BF 7E      1310 LD A,(HL)
C0C0 CB3F    1320 SHIFT SRL A
C0C2 10FC    1330 DJNZ SHIFC
C0C4 3802    1340 JR C WEI
C0C6 CBFB    1350 SET 7,E
C0C8 CB3B    1360 WEI SRL E ;Bit0 in E ganz rechts
C0CA 0D      1370 DEC C ;line-block finished?
C0CB 20EB    1380 JR NZ BITNUM
C0CD 7B      1390 LD A,E
C0CE CBFF    1400 SET 7,A ;für TANDY Bit7 immer H
IGH
C0D0 CDD6C0  1410 CALL PRINT
C0D3 E1      1420 POP HL ;"erste" Zeile
C0D4 18DA    1430 JR NXTBIT
1440 ;
C0D6 F5      1450 PRINT PUSH AF ;OUTPUT an Drucker
C0D7 DBF4    1460 BUSY IN A,(PIOAD)
C0D9 CB7F    1470 BIT 7,A
C0DB 28FA    1480 JR 2 BUSYz C0DD F1 1490
POP AF
C0DE D66F    1500 DEFW #6FDF ;SRLX
C0E0 FF      1510 RST #38 ;zweimal Delay bei 4MHz

C0E1 FF      1520 RST #38
C0E2 C9      1530 RET
1540 ;-----
2000 ;INITIIERUNG der mc-PIO/SIO-KARTE
2010 ;für TANDY LINEPRINTER VI
2020 ;
C0E3 00F0    2030 SIOADA EQU #F0 ;DATEN
C0E3 00F1    2040 SIOAST EQU #F1 ;STATUS
C0E3 00F5    2050 PIOAC EQU #F5 ;CONTROL
C0E3 00F4    2060 PIOAD EQU #F4 ;DATA (BIT7 als BUSY)
C0E3 005B    2070 SRLX EQU #005B
2080 ;
C0E3 180F    2090 INIT JR INITS
C0E5 010003C1 2100 TABSIO DEFB 1,0,3,#C1,4,#4C,5,#68
044C0568
C0ED F5D3F0DB 2110 TABX DEFB #F5,#D3,#F0,#DB,#F1,#CB,#57

F1CB57
2120 ;
C0F4 0EF1    2130 INITS LD C,SIOAST ;SIO INITIIEREN
C0F6 0608    2140 LD B,8
C0F8 21E5C0  2150 LD HL,TABSIO
C0FB EDB3    2160 OTIR
2170 ;
C0FD 3E4F    2180 INITP LD A,#4F ;PIOA INPUT MODE
C0FF D3F5    2190 OUT (PIOAC),A
2200 ;
C101 21EDC0  2210 INITX LD HL TABX ;NASSYS SRLX UMSCHR
EIB.
C104 115800  2220 LD DE,SRLX
C107 010700  2230 LD BC,7
C10A EDB0    2240 LDIR
2250 ;
C10C C308C0  2260 JP STORE

```

Mit obigem Programm wird die Grafik in 256 Zeilen zu 512 Punkten im Maßstab 1:1 ausgegeben. Da nun aber der EF 9366 ein quadratisches Bild erzeugt, erscheint die gedruckte Grafik üblicherweise verzerrt, d.h. in Y-Richtung gestaucht; aus einem Kreis wird eine Ellipse, aus einem Quadrat ein Rechteck.

Um nun den Ausdruck im gleichen Verhältnis wie der Bildschirm darzustellen, kann von der X-Achse jeder zweite Punkt unterdrückt werden. Wie sich das Erscheinungsbild verändert, sehen Sie in den abgedruckten Beispielen.

Folgende Zeilen sind dazu im Listing zu ändern:

```

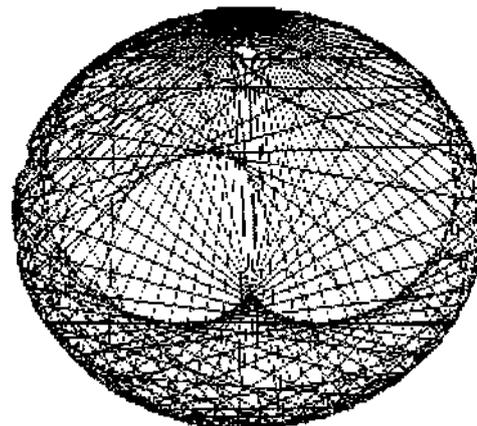
Z1190 GETBYT LD DE,-1
Z1200 NXTBIT INC D
Z1201 INC D

```

Dadurch werden nur die ungeraden Bits an den Drucker ausgegeben.

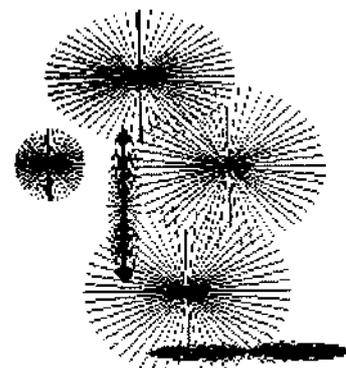
ELLIPSE I

Press 'NL'



ELLIPSE II

Press 'NL'



# Taschenrechner

von UWE BROCKMÖLLER

## WISSENSCHAFTLICHER TASCHENRECHNER

=====

AUCH DIESES PROGRAMM HAT EINMAL IN DER 'CHIP' BESTANDEN. ES WAR URSPRUEGLICH FUER EINEN APPLE GESCHRIEBEN UND ICH HABE ES LEDIGLICH FUER DEN NASCOM ANGEPAESST. ES IST GANZ INTERESSANT EINEN HP-TASCHENRECHNER AUF DEM BILDSCHIRM ZU SIMULIEREN. DIE EINGABEN WERDEN IN SOG. UMGEKEHRTER POLNISCHER NOTATION GEMACHT. DAS HEISST, ZUERST EINGABE DER ZAHLEN (IMMER MIT NL ABSCHLIESSEN), DANN ZULETZT DIE FUNKTION. MAN KANN SICH JEDERZEIT DURCH EINGABE EINES '?' ALLE FUNKTIONEN AUF DEN BILDSCHIRM HOLEN.

```
10 REM ** WISSENSCHAFTLICHER TASCHENRECHNER**
20 REM ** MIT SPEICHER.
30 REM ** FUER NASCOM UMGESCHRIEBEN VON **
40 REM ** UWE BROCKMOELLER **
50 CLS: INPUT "WOLLEN SIE INSTRUKTIONEN?" : INST$
60 IF LEFT$(INST$, 1) = "J" THEN GOSUB 880
70 DIM MEM(9) : REM RESERVIERT 10 SPEICHERSTELLEN
80 CLS : PRINT "X=" : X : " Y=" : Y : " Z=" : Z : " }=" : }
81 PRINT " ? FUER INFO!!!"
90 INPUT TAST$
100 IFTAST$ = "+" THEN X = X + Y : Y = Z : Z = T : T = 0 : GOT080
110 IFTAST$ = "-" THEN X = Y - X : Y = Z : Z = T : T = 0 : GOT080
120 IFTAST$ = "*" THEN X = X * Y : Y = Z : Z = T : T = 0 : GOT080
130 IFTAST$ = "/" THEN X = X / Y : Y = Z : Z = T : T = 0 : GOT080
140 IFTAST$ = "SQRT" THEN X = SQR(X) : GOT080
150 IFTAST$ = "ABS" THEN X = ABS(X) : GOT080
160 FORUW = 1 TO 90 : NEXTUW
170 X = Y / X : Y = Z : Z = T : T = 0 : GOT080
180 IFTAST$ = "EXP" THEN X = EXP(X) : GOT080
190 IF Y = 0 THEN 210
200 GOT0240
210 PRINT "*** UNERLAUBTE RECHENOPERATION ***"
220 PRINT "Y IST < ODER = 0"
230 FORG = 1 TO 999 : NEXTG : GOT080
240 X = Y * X : Y = Z : Z = T : T = 0 : GOT080
250 IFTAST$ = "1/X" THEN X = 1 / X : GOT080
260 IFTAST$ = "XY" THEN STACK = X : X = Y : Y = STACK : GOT080
270 IFTAST$ = "Y" THEN STACK = T : T = Z : Z = Y : Y = X : X = STACK : GOT080
280 IFTAST$ = "X" THEN STACK = X : X = Y : Y = Z : Z = T : T = STACK : GOT080
290 IFTAST$ = "1/X" THEN 310
300 IFTAST$ = "1/X" THEN X = 1 / X : GOT080
310 IFTAST$ = "W" THEN X = SQR(ABS(X)) : GOT080
320 REM WURZEL AUS NEGATIVER ZAHL NICHT DEFF.
330 IFTAST$ = "Q" THEN X = X * X : GOT080
340 IFTAST$ = "E" THEN X = 2.71828 ^ X : GOT080
350 IFTAST$ = "SIN" THEN X = SIN(X / 57.29577954341) : GOT080
360 IFTAST$ = "COS" THEN X = COS(X / 57.29577954341) : GOT080
370 IFTAST$ = "TAN" THEN X = TAN(X / 57.29577954341) : GOT080
380 IFTAST$ = "ASN" THEN 440
390 IF X < -1 OR X > 1 THEN 410
400 GOT0430
410 PRINT "*** ARCSIN NUR VON -1 BIS +1 DEFF. **"
420 FORUW = 1 TO 90 : NEXTUW : GOT080
430 X = ATN(X / (SQR(1 - X * X))) * 57.29577954341 : GOT080
440 IFTAST$ = "ACS" THEN 500
450 IF X < -1 OR X > 1 THEN 470
460 GOT0490
470 PRINT "*** ARCCOS NUR VON -1 BIS +1 DEFF. ***"
480 FORGU = 1 TO 999 : NEXTGU : GOT080
490 X = (1.570796327 - ATN(X / (SQR(1 - X * X)))) * 57.2577954341 : GOT080
500 IFTAST$ = "ATN" THEN X = ATN(X) * 57.29577954 : GOT080
510 IFTAST$ = "LN" THEN 560
```

```
520 IF X = 0 THEN 540 :
530 X = LOG(X) : GOT080
540 PRINT "X IST < ODER = 0" : FORGU = 1 TO 90 : NEXTGU
550 GOT080
560 IFTAST$ = "LOG" THEN 590
570 IF X < 0 THEN 540
580 X = LOG(X) / LOG(10) : GOT080
590 IFTAST$ = "INT" THEN X = INT(X) : GOT080
600 IFTAST$ = "ABS" THEN X = ABS(X) : GOT080
610 IFTAST$ = "MOD" THEN 640
620 IF Y = 0 THEN 630
630 X = Y - X : GOT080
640 IFTAST$ = "PI" THEN 650
650 IFTAST$ = "N!" THEN 690
660 IF X = 0 THEN 670
670 B = 1
680 FORN = 1 TO X : B = B * N : NEXTN : X = B : GOT080
690 IFTAST$ = "S+" THEN S0 = S0 + X : S1 = S1 + 1 : S2 = S2 + X * X : GOT080
700 IFTAST$ = "S-" THEN S0 = S0 - X : S1 = S1 - 1 : S2 = S2 - X * X : GOT080
710 IFTAST$ = "MIT" THEN X = S0 / S1 : GOT080
720 IFTAST$ = "ABW" THEN X = SQR((S2 - S0 * S0 / S1) / (S1 - 1)) : GOT080
730 IFTAST$ = "+-" THEN X = X * (-1) : GOT080
740 IFTAST$ = "C" THEN X = 0 : GOT080
750 IFTAST$ = "CM" THEN 770
760 FORN = 0 TO 9 : MEM(N) = 0 : NEXTN : S0 = 0 : S1 = 0 : S2 = 0 : GOT080
770 IF LEFT$(TAST$, 1) = "M" THEN 810
780 B = VAL(RIGHT$(TAST$, 1))
790 IF B = 0 AND B < 10 THEN MEM(B) = X : GOT080
800 REM NUR SPEICHER 0 BIS 9 ERLAUBT
810 IF LEFT$(TAST$, 1) = "R" THEN 840
820 B = VAL(RIGHT$(TAST$, 1))
830 IF B = 0 AND B < 10 THEN T = Z : Z = Y : Y = X : X = MEM(B) : GOT080
840 IFTAST$ = "?" THEN GOSUB 930 : GOT080
850 T = Z : Z = Y : Y = X : X = VAL(TAST$) : GOT080
860 REM EINGABE EINER ZAHL
870 CLS
880 PRINT "WISSENSCHAFTLICHER RECHNER MIT SPEICHER" : PRINT
890 PRINT "=====
900 PRINT "DER RECHNER ARBEITET MIT UMGEKEHRTER POLNISCHER NOTATION"
910 PRINT "ALSO ERST EINGABE EINER ZAHL, DANN DER FOLGENDE FUNKTIONEN SIND PROGRAMMIERT" : PRINT
920 PRINT "X MAL Y (ERGEBNIS IM X REGISTER)"
930 PRINT "X MAL Y (ERGEBNIS IM Y REGISTER)"
940 PRINT "INHALT DES X-REGISTERS WIRD ZUM Y-REGISTER SUBTRAHIERT"
950 PRINT "X MAL Y (ERGEBNIS IM X REGISTER)"
960 PRINT "Y GETEILT DURCH X (ERGEBNIS IM X-REGISTER)"
970 INPUT "DRUECKEN SIE 'W' FUER WEITER" : DUMMY$
980 PRINT "X=Y HOCH X"
990 PRINT "X=X% VON Y"
1000 PRINT "XY INHALT VON X Y REG. VERTAUSCHEN"
1010 PRINT "RECHTS SCHIEBEN (X IN Y, Y IN X USW)"
1020 PRINT "LINKS SCHIEBEN (Y IN X, X IN Y USW)"
1030 PRINT "X=1 GETEILT DURCH X"
1040 PRINT "X=WURZEL AUS X"
1050 PRINT "X=ZUM QUADRAT"
1060 PRINT "X=ZAHL E(2,71828) HOCH X"
1070 PRINT "X=SINUS VON X (ANGABE IN ALTGRAD)"
1080 PRINT "X=COSINUS VON X"
1090 PRINT "X=TANGENS VON X"
1100 PRINT "X=ARCUSSINUS VON X"
1110 PRINT "X=ARCUSCOSINUS VON X"
1120 INPUT "DRUECKEN SIE 'W' FUER WEITER" : DUMMY$
1130 PRINT "X=ARCUSTANGENS VON X"
1140 PRINT "X=DEKADISCHER LOGARITHMUS VON X"
1150 PRINT "X=NATUERLICHER LOGARITHMUS VON X"
1160 PRINT "X=GANZZAHLIGER ANTEIL VON X"
1170 PRINT "X=ABSOLUTWERT VON (-2,66->2,66)"
```

```

1180 PRINT"MOD VOM Y-REGISTER WIRD X SO OFT WI
E"
1190 PRINT"MOEGELICH ABGEZOGEN UND INS X-REG. GE
SCHOBEN."
1200 PRINT"PI ZAHL PI(3,14) WIRD INS X-REG. A
UFGENOMMEN."
1210 PRINT"N! X=4*2*3*.....*X"
1220 PRINT"S+ SUMMIERT ZAHLEN ZUR MITTELWERTB
ERECHNUNG"
1230 PRINT"S- ERLAUBT FALSCH EINGEGEBENE ZAHL
ZU ENTFERNEN"
1240 INPUT"DRUECKEN SIE 'W' FUER WEITER";DUMMY#
1250 PRINT"MIT BERECHNET MITTELWERT DER EINGEG
EB. ZAHLEN"
1260 PRINT"ABW BERECHNET DIE STANDARTABWEICHUN
G"
1270 PRINT"+- WECHELT DAS VORZEICHEN IM X-REG
ISTER"
1280 PRINT"C SETZT X-REGISTER AUF NULL"
1290 PRINT"CM SETZT ALLE SPEICHER AUF NULL"
1300 PRINT"MR SCHIEBT X-REGISTER IN SPEICHER
R(R=0BIS9)"
1310 PRINT"RR HOLT SPEICHER R IN X-REGISTER(R
=0BIS9)"
1320 PRINT:INPUT"DRUECKE 'W' FUER WEITER";XYZ#
1330 RETURN
OK

```

# Schwingkreis

von REINER STEGEMANN

```

1 CLS
2 REM **** REINER STEGEMANN ****
3 REM ****          ****
4 REM **** RASTEDE ****
10 PRINT"SCHWINGKREISBERECHNUNGEN MIT DEM NASCO
M-1"
11 PRINT:PRINT"DIESES PROGRAMM ERMOEGLICHT DIE
BERECHNUNG VON VERSCHIEDENEN SCHWINGKREISWER
TEN."
12 PRINT"UNG VON VERSCHIEDENEN SCHWINGKREISWER
TEN."
13 PRINT"ES BRAUCHT NURDIE ENTSPRECHENDE ZAHL";
14 PRINT" EINGEBEN WERDEN,DANN"
15 PRINT"WERDEN DIE ZUR BERECHNUNG ERFORDERLICH
EN"
16 PRINT"ANGABEN ABGEFRAGT UND DANN DIE BERECHN
UNG"
17 PRINT"DURCHGEFUEHRT!"
28 PRINT:PRINT"SCHWINGKREISBERECHNUNG"
30 PRINT"1.FREQUENZBERECHNUNG"
32 PRINT"2.INDUKTIVITAETSBERECHNUNG"
34 PRINT"3.KAPAZITAETSBERECHNUNG"
36 PRINT"4.BERECHNUNG ABSTIMMBEREICH"
38 INPUT#
40 IFA#="" THEN38
41 IFA#="1" THEN52
42 IFA#="2" THEN62
44 IFA#="3" THEN72
46 IFA#="4" THEN82
48 IFA#("<"1"ANDA#("<"2"ANDA#("<"3"ANDA#("<"4" THEN4
9
49 PRINT"NUR 1 BIS 4";GOTO28
50 END
52 CLS:INPUT"INDUKTIVITAET IN UH";L
54 PRINT:INPUT"KAPAZITAET IN PF";K
56 F=459/SQR(L*K)
58 PRINT"FREQUENZ IN MHZ";F
60 PRINT:PRINT:GOTO28
62 PRINT:INPUT"FREQUENZ IN MHZ";F
64 INPUT"KAPAZITAET IN PF";K
66 L=25330/((F^2)*K)
68 PRINT"INDUKTIVITAET IN UH";L
70 PRINT:PRINT:GOTO28
72 PRINT:INPUT"FREQUENZ IN MHZ";F
74 INPUT"INDUKTIVITAET IN UH";L
76 C=25330/((F^2)*L)
78 PRINT"KAPAZITAET IN PF";C
80 GOTO28
82 PRINT:PRINT"CA=ANFANGSKAPAZITAET DES DREKOS"

```

```

84 PRINT"CE=ENDKAPAZITAET DES DREKOS"
86 PRINT"FO=OBERE GRENZFREQUENZ ERREICHT MIT CA
"
88 PRINT"FU=UNTERE GRENZFREQUENZ ERREICHT MIT C
E"
90 PRINT:INPUT"ANFANGSKAPAZITAET CA";CA
92 INPUT"ENDKAPAZITAET CE";CE
94 INPUT"UNTERE GRENZFREQUENZ";FU
96 FO=FU*SOR(CE/CA)
98 PRINT"OBERE GRENZFREQUENZ ="FO
100 GOTO28
OK

```

# Leserbrief

SEIT EINIGER ZEIT FAHRE ICH NUN SCHON EIN 8" SHUGARD UND EIN 5 1/4" BASF DRIVE IM GEMISCHTBETRIEB AN MEINEM NASCOM-1-. ES SIND BIS JETZT KEINE GROSSEN PROBLEME AUFGETRETEN. DIESES SPRICHT FUER DIE HERVORRAGENDE CONTROLLERKARTE (FINDE ICH WENIGSTENS). ICH ARBEITE ZTZ. NOCH MIT DEM GEAENDERTEN PHEAS AUS HEFT 12/83. UM MIT DEM 8" LAUFWERK AUCH 80 SPUREN BEARBEITEN ZU KOENNEN,WAR EINE KLEINE AENDERUNG IN PHEAS NOTWENDIG. DAS 5 1/4" LW IST BEI MIR DRIVE "D" UND DAS 8" LW DRIVE "C". FUER 8" LW BENUTZE ICH DIE DS8B AUF ADR. A9DF. DORT STEHT BEI MIR AUF ADR.A9E1=BD. SOMIT KANN EMDOS NUN 80 SPUREN VERWALTEN UND DAS PROGRAMM "STAT" ERKENNT AUCH DIE GROESSERE KAPAZITAET. DAS 5 1/4" LW BENUTZT DIE DS8A AUF ADR.A9DC. DORT HABE ICH DANN AUF A9DC 28 08 57 EINGETRAGEN. LW "A" UND "B" BENUTZE ICH NOCH NICHT, DA ICH KEIN DD FAHREN KANN.

UM DIE 8" DISKETTEN AUCH MIT 80 SPUREN ZU FORMATIEREN, MUSS IM FORMATIERPROGRAMM AUS HEFT 10/11 1983 AUCH EINE KLEINE AENDERUNG VORGENOMMEN WERDEN.DIE MAX. SPUR ABFRAGE AUF ADR. 8454 ( CP 40) MUSS AUF ( CP 80 ) ENTSPRICHT ADR.8454=FE25 GEAENDERT WERDEN.

MITTLERWEILE HABE ICH AUCH DAS COPIERPROGRAMM VON GUENTER BOEHM EINGETIPPT. ICH HABE ES FUER DAS ALTE EMDOS GEAENDERT UND KANN NUN IN SD VON DEM 5 1/4" AUF DAS 8" DRIVE COPIEREN.

ZWEI FRAGEN SIND BIS JETZT AUFGETAUCHT. BEI DEM PROGRAMM "STAT" WERDEN MIR DIE PROGRAMMSIZE AUF DER DISKETTE FALSCH ANGEZEIGT. BEISPIEL EIN 1 KB PROGRAMM ERSCHEINT IN STAT MIT 20KB GROSSE. ICH HABE "STAT" MIT PRUEFSUMME EINGETIPPT. FUEUERLEICHT KANN MIR DA JEMAND HELFEN. \*

DANN WURDE, WENN ICH ES RICHTIG VERSTANDEN HABE GESCHRIEBEN, DAS EMDOS ZUMINDESTENS DEN DIRECTORY VON CP/M DISKETTEN ANZEIGEN KANN. ICH KONNTE DIESE ERFAHRUNG NICHT MACHEN. BEI CP/M STEHT DER DIRECTORY AUF SPUR 3 (VON 0 AUS GERECHNET) UND BEI EMDOS STEHT ER AUF SPUR 4. AENDERE ICH DEN ERSTEN EINTRAG IN DER DS8A VON 28 (DAS ENTSPRICHT JA SO 4 SPUREN A 10 SEKTOREN) AUF 1E (ENTSPRICHT SO 3 SPUREN A 10 SEKTOREN) DANN KANN ICH DEN DIRECTORY EINER CP/M DISKETTE OB NUN 8" ODER 5 1/4" ANZEIGEN LASSEN. STIMMT DA NUN DIE LAGE UNSERES EMDOS DIRECTORY'S NICHT ODER MACHE ICH EINEN DENKFEHLER? \*\* UWE BROCKMOELLER 2900 OLDENBURG

Antwort der Redaktion:  
\* und\*\*siehe Floppy-Seite

# Floppyseite

von GÜNTER BÖHM

FDC-Seite

Seit langem ist nun die "Seite für Floppy-Einsteiger" schon ein fester Bestandteil des Journals. Es wäre an der Zeit, sie endlich hinauszuerwerfen, aber leider ergeben sich immer wieder Umstände, die erneute "Nachlesen" für Hard- und Software notwendig machen.

Zunächst ein Käfer, der die Funktion des STAT.COM zur Statusanzeige einer Diskette beeinträchtigt.

Im Hex-Listing (Heft 10/11-83 Seite 18) hat sich aus ungeklärten Gründen ein Byte beim Ausdruck verändert. Korrigieren Sie also bitte:

13D2 63 -> 23

Damit läuft das Programm. (Danke, Herr Flockau).

Der Kopierbefehl "C" des EMDOS funktionierte bisher auch nicht, d.h. das Directory der kopierten Diskette schien einwandfrei, die zuerst kopierten Programme ließen sich aber nicht laden.

Das Problem liegt daran, daß sich die alten Versionen von PHEAS nicht merken, welche Sour vor dem Umschalten auf ein anderes Laufwerk gerade bearbeitet wurde. Nach erneutem Umschalten geht dann alles drunter und drüber.

Beim BACKUP-Programm hatte ich diesem Umstand schon Rechnung getragen. Nun habe ich auch das PHEAS entsprechend erweitert, sodaß nun leider schon Version 2.5 vorliegt.

Im folgenden ist der Programmteil nochmals als Assembler abgedruckt; die Änderungen sind markiert. Dann folgt der Hex-Ausdruck des gesamten PHEAS2.5. Der Assembler des neuen PHEAS befindet sich auf der Diskette zu diesem Heft (Disk2). Im Rahmen einer "Software-Pflege" würde ich Ihnen aber auch jederzeit Ihre Diskette entsprechend ändern (unentgeltlich versteht sich).

```
0210 ;-----  
0220 PSEL CALL TRKMEM ;TRACK speichern ←  
0230 LD A,C  
0240 LD (LWNR),A ;neues Laufwerk ←  
0250 OR A  
0260 JR Z PSELA
```

```
0270 CP 1  
0280 JR Z PSELB  
0290 CP 2  
0300 JR Z PSELC  
0310 CP 3  
0320 JR Z PSELD  
0330 LD A,#17  
0340 SCF  
0350 RET  
0360 ;  
0370 PSELA LD A,#21 ; LAUFW. A DD  
0380 OUT (10H),A  
0390 LD HL,DSBA  
0400 LD A,16 ;Sekt.  
0410 LD (SEKT),A  
0420 LD A,64  
0430 LD (SYS),A  
0440 LD A,(TRACKA) ;gespeicherte Tracknummer ←  
0450 OUT (FDCTRK),A ;aktualisieren ←  
0460 RET  
0470 ;  
0480 ;  
0490 PSELB LD A,#22 ; LAUFWERK B DD  
0500 OUT (10H),A  
0510 LD HL,DSBA ;gleiches Format wie Lw A  
0520 LD A,16 ;Sekt.  
0530 LD (SEKT),A  
0540 LD A,64  
0550 LD (SYS),A  
0560 LD A,(TRACKB) ←  
0570 OUT (FDCTRK),A ←  
0580 RET  
0590 ;  
0600 PSELC LD A,#31 ;Laufwerk A SD  
0610 OUT (#10),A  
0620 LD HL,DSBB  
0630 LD A,10 ;Sekt.  
0640 LD (SEKT),A  
0650 LD A,40  
0660 LD (SYS),A  
0670 LD A,(TRACKC) ←  
0680 OUT (FDCTRK),A ←  
0690 RET  
0700 ;  
0710 PSELD LD A,#32 ;Laufwerk B SD  
0720 OUT (#10),A  
0730 LD HL,DSBB  
0740 LD A,10 ;Sekt.  
0750 LD (SEKT),A  
0760 LD A,40  
0770 LD (SYS),A  
0780 LD A,(TRACKD) ←  
0790 OUT (FDCTRK),A ←  
0800 RET  
0810 ;
```

```

0820 DSBA DEFB,64;4 X SEKTORANZAHL (res.System)
0830 DEFB 8 ;8 Sekt. DIRECTORY
0840 DEFB 143 ;max.Gruppenanzahl-1
0850 ;
0860 DSBB DEFB 40,8,87 ;NEUES FORMAT mc !
0870 SEKT DEFW 0 ;SEKTOREN PRO SPUR
0880 SYS DEFW 0 ;SEKT. FUER SYSTEM
0890 TRACKA DEFB 0
0900 TRACKB DEFB 0
0910 TRACKC DEFB 0
0920 TRACKD DEFB 0
0930 LWNR DEFB 0
0940 ;
0950 TSTHOM CALL HOME ;WARUM IST LAUFWERK
0960 CALL CTDEL ;NICHT BEREIT?
0970 CALL TRKMEM ;TRACK0 aktualisieren
0980 IN A,(FDCSTA)
0990 BIT 2,A
1000 JR NZ NODISK
1010 LD A,#1F ;KEIN LAUFWERK
1020 JR TSTERR
1030 NODISK BIT 1,A
1040 JR Z NOTRDY
1050 LD A,#10 ;KEINE DISKETTE
1060 JR TSTERR
1070 NOTRDY BIT 7,A
1080 RET Z
1090 LD A, #1A
1100 TSTERR SCF
1110 RET
1120 ;-----
1130 TRKMEM PUSH HL ;Routine zum Abspeichern
1140 PUSH BC ;des aktuellen Tracks
1150 LD A,(LWNR)
1160 LD B,A
1170 INC B
1180 LD HL,TRACKA-1
1190 LOOP INC HL
1200 DJNZ LOOP ;akt. Laufwerk
1210 IN A,(FDCTRK)
1220 LD (HL),A
1230 POP BC
1240 POP HL
1250 RET
1260 ;
1270 ;-----

```

Ein weiterer Käfer betrifft die Benutzung von EMDOS durch Maschinenprogramme, wie sie in Heft 1/84 beschrieben wurde. Die Aufrufnummern beginnen mit 61 (nicht 60). File lesen wird also durch 61H, File schreiben durch 62H etc. aufgerufen. Hoffentlich haben Sie nicht so lange ergebnislos herumprobiert

wie ich! Das werde ich dem Helmut Emmelmann nicht verzeihen!

Als Wiedergutmachung hat er dafür einige Informationen über den Aufbau der File-Header und des Directory herausgerückt, die ich hiermit veröffentlichen will.

Directory

← Ein Eintrag umfaßt 32 Bytes. Diese haben folgende Bedeutung:

- ← 1. 00 Beginn des Eintrags
- ← 2.-9. Filename
- ← 10.-12. Name-Extension z.B. BAS, NAS etc.
- ← 13. Extension Nr (beim ersten Eintrag 0)
- ← 14.-15. unbenutzt
- ← 16. Länge des Files in Rec. a 128 Bytes
- ← 17.-32. Jeweilige Gruppennummern

Die Sektoren sind zu Gruppen von je 4 (entspricht 1 KByte) zusammengefaßt. Das Directory beginnt mit der ersten Gruppe auf der Diskette (Gruppe 0).

Lange Programme belegen 2 Einträge im Directory, die hintereinander liegen, wobei aber auch andere Einträge dazwischengeschoben sein können. EMDOS kann nur eine solche "Extension" verarbeiten, weshalb die Maximallänge eines Files auch 31 KByte beträgt.

File Header

Der Beginn eines Files, wie er auf der Diskette gespeichert ist, wird File Header genannt. Er besteht jeweils aus 16 Bytes mit folgender Bedeutung:

- 11 22 44 48 Kennung für Beginn
- Ladeadresse
- Länge
- Startadresse
- Funktion (ausführen, Call etc.)
- 4 Bytes unbenutzt

Vielleicht geht nun jemand daran, die EMDOS Aufrufe in ZEAP, NASPEN etc. einzubauen. Damit wäre ein viel komfortableres Arbeiten möglich.

Inzwischen hat sich EMDOS sehr bewährt, und ich kann mir nicht mehr vorstellen, wie ich so lange mit Cassetten arbeiten konnte. Die Austauschbarkeit mit den mc-Benutzern ist inzwischen auch durch die Praxis bewiesen; EMDOS kann Texte lesen, die unter CP/M mit einem mc-Computer abgespeichert wurden. Bei 8"-Laufwerken hat sich aber herausge-

stellt, daß mc auf diesen Disketten das Directory auf Spur 3 legt (also nur 3 Sournen für das System reserviert). 8"-Benutzer sollten Ihr PHEAS an diesen Standard anpassen.

Über die Hardware gibt es nicht viel Neues. Ein kleiner Tip für Benutzer der mc-CPU-Karte. Auf dieser Karte sind die IEI- und IEO- Leitungen verbunden. Wenn Sie nun einen durchverbundenen Bus haben (sehr häufig anzutreffen), so erhält die PIO auf der FDC-Karte kein Interrupt-Enable Signal (high). Sie müssen also die Verbindung auf der CPU-Karte zwischen 11c und 16c durchtrennen.

Diesen kleinen Abschnitt schon früher zu lesen (wenn es ihn gegeben hätte) hätte Peter Brendel kürzlich die Gebühr für ein eineinhalbstündiges Ferngespräch erspart. Nichts Neues zur DMA, ebensowenig zum "Verschiebeprobem". Letzteres wurde ja bei mir, wie im letzten Heft erwähnt, durch Entfernen des Datentreibers gelöst. Da bisher keine nachprüfbar Fehlerquelle für das Problem entdeckt bzw. eingeschickt wurde, haben wir uns entschlossen, den Preis unseres Preisausschreibens an Michael Grimme zu "verleihen", der uns den Tip zur Fehlerbeseitigung gegeben hat. Gratuliere, Herr Grimme, der überwältigende Preis von DM 20,- ist unterwegs. Vielleicht findet sich für unser nächstes Preisausschreiben ein Spender (vielleicht irgendwelche Sachspenden?), damit es nicht so ärmlich aussieht. Aber mit der augenblicklichen Finanzlage trauen wir uns eben nicht, größer einzusteigen.

So weit zur Floppy. Wenn sich der Zippel nicht mit Neuigkeiten zur DMA oder dem versprochenen neuen Schaltplan meldet, haben wir vielleicht im nächsten Heft Ruhe mit der Floppy-Seite. Wird ja auch Zeit!

## Bandpass

von UWE BROCKMÖLLER

FILTERBERECHNUNG // BANDPASS 2. ORDNUNG //

UWE BROCKMÖLLER (DF3DT)

DIESES PROGRAMM WURDE MAL IN DER ZEITSCHRIFT CHIP FUER DEN TRS 80 VERÖFFENTLICHT. DA DIE FILTERSCHALTUNG SEHR SCHÖN MIT DER BLOCKGRAFIK AUF DEM BILDSCHIRM DARGESTELLT WIRD, HABE ICH DAS PROGRAMM MAL FUER DEN NASCOM ANGEPA SST.

OK  
LIST

```

1 REM *** FUER NASCOM ANGEPA SST ***
2 REM *** UWE BROCKMÖLLER ***
3 REM *** ██████████ ██████████ ***
4 REM *** ██████████ OLDBURG ***
5 PI=3.1415926:
8 CLS
9 SCREEN 10,1
10 PRINT"**** FILTERBERECHNUNG ****"
11 PRINT
12 SCREEN 10,3
13 PRINT"ACTIVER BANDPASS 2.ORDNUNG"
40 FORI=1TO200:NEXT:CLS
100 SCREEN10,1:INPUT"VERSTAERKUNG ";V
110 SCREEN10,3:INPUT"MITTENFREQUENZ [HZ]";F
120 SCREEN10,5:INPUT"BANDBREITE [HZ]";B
130 SCREEN10,7:INPUT"KONDENSATOR [UF]";C
140 Q=F/B:C1=C*1E-6;R3=Q/(PI*F*C1);R1=R3/V
145 PRINT
150 R2=R3*(2*PI*C1*F)^2;R2=1/R2
155 PRINT"R2=";R2;" OHM"
156 INPUT" NEUER WERT FUER R2 (J/N) ";A$
160 PRINT"R3=";R3;" OHM"
162 INPUT" NEUER WERT FUER R3 (J/N) ";B$
165 IFA$="J"THENINPUT"WIDERSTAND R2 (NEU)=";R2
170 IFB$="J"THENINPUT"WIDERSTAND R3 (NEU)=";R3
175 R1=R3/V;F=BQR((R1+R2)/(R1*R2*R3))/(2*PI*C1)
176 PRINT
180 PRINT"MITTENFREQUENZ (NEU) =";F;" HZ"
185 B=1/(PI*C1*R3);
186 PRINT"BANDBREITE (NEU) =";B;" HZ"
190 INPUT"NEUE WERTE R2/R3 (J/N) ";A$
200 IFA$="J"GOTO165
210 GOTO1000
215 SCREEN 1,9
220 PRINT"R1 =";R1;" OHM ";R2 =";R2;" OHM"
223 PRINT"R3 =";R3;" OHM "
230 PRINT" C =";C1=C1*1E6;PRINTC:PRINT" MICROF
240 PRINT"MITTENFREQUENZ=";F;" HZ"
250 PRINT"BANDBREITE =";B;" HZ"
255 SCREEN 30,14:INPUT"WEITER? (J/N)";A$
260 IFA$="J"THEN 265
264 GOTO300
265 CLS:INPUT"GLEICHE VERSTAERKUNG? ";A$
270 IFA$="N" THEN CLS: GOTO100
275 INPUT"GLEICHE MITTENFREQUENZ ? ";A$
280 IFA$="N" THEN CLS: GOTO110
285 INPUT"GLEICHE BANDBREITE ";A$
290 IFA$="N" THEN CLS: GOTO120
295 GOTO130
300 END
1000 CLS:FOR Y=1TO23:SET(47,Y):NEXT
1010 FOR Y=4TO10:SET(60,Y):NEXT
1020 FOR X=22TO68:SET(X,10):NEXT
1030 FOR X=60TO68:SET(X,16):NEXT
1040 FOR X=60TO94:SET(X,4):NEXT
1045 GOSUB 1050
1048 GOTO1200
1050 FOR X=47TO94:SET(X,11):NEXT
1060 FOR X=22TO95 :SET(X,23):NEXT
1070 FOR Y=16TO23:SET(60,Y):NEXT
1080 FOR X=85TO95 :SET(X,13):NEXT
1090 FOR Y=6TO20:SET(69,Y):NEXT
1100 FOR Y=1TO13:SET(94,Y):NEXT
1110 FOR X=70TO85:Y=-INT(X/2)+55:SET(X,Y):NEXT
1120 FOR X=70TO85:Y=INT(X/2)-29:SET(X,Y):NEXT
1130 RETURN
1200 SCREEN40,1:PRINT" C "
1210 SCREEN40,2:PRINT" R3 "
1215 SCREEN36,4:PRINT"- "
1216 SCREEN36,6:PRINT"+ "
1218 SCREEN15,4:PRINT" R2 "
1219 SCREEN22,6:PRINT" R1 "
1220 SCREEN26,4:PRINT" C1 "
1221 SCREEN10,6:PRINT"-- UE --"
1222 SCREEN44,7:PRINT"--UA--"
1250 GOTO215

```

OK

# PHEAS 2.5

# USR(N)

TAB47 AD9A

```

AE47 B7 CA 93 AD 3D CA E3 AB
AE4F 3D 28 0C 3D CA 18 AC 3D
AB57 CA 84 AC 37 C9 74 AD CD
AB5F 05 AC 79 32 E2 AB B7 28
AB67 10 FE 01 28 23 FE 02 28
AB6F 36 FE 03 28 49 3E 17 37
AB77 C9 3E 21 D3 10 21 D4 AB
AB7F 3E 10 32 DA AB 3E 40 32
AB87 DC AB 3A DE AB D3 0D C9
ABBF 3E 22 D3 10 21 D4 AB 3E
AB97 10 32 DA AB 3E 40 32 DC
AB9F AB 3A DF AB D3 0D C9 3E
ABA7 31 D3 10 21 D7 AB 3E 0A
ABAF 32 DA AB 3E 28 32 DC AB
ABB7 3A E0 AB D3 0D C9 3E 32
ABBF D3 10 21 D7 AB 3E 0A 32
ABC7 DA AB 3E 28 32 DC AB 3A
ABCF E1 AB D3 0D C9 40 08 EF
ABD7 2B 08 57 10 00 40 00 00
ABDF 00 00 00 00 CD E3 AC CD
ABE7 41 AC CD 05 AC DB 0C CB
ABEF 57 20 04 3E 1F 18 0D CB
ABF7 4F 28 04 3E 10 18 05 CB
ABFF 7F CB 3E 1A 37 C9 E5 C5
AC07 3A E2 AB 47 04 21 DD AB
AC0F 23 10 FD DB 0D 77 C1 E1
AC17 C9 C5 E5 CD 53 AC D5 E1
AC1F 01 02 05 CD FC AC B7 28
AC27 13 10 F8 0D 28 0E CD E3
AC2F AC E1 E5 CD 53 AC D5 E1
AC37 06 05 18 E7 E1 C1 B7 20
AC3F 0F C9 F5 C5 06 04 0E FA
AC47 CD 60 AD 10 F9 C1 F1 C9
AC4F 3E 11 37 C9 DB 0C CB 7F
AC57 C4 E3 AB 30 04 E1 E1 E1
AC5F C9 C5 CB 7C 28 07 CB BC
AC67 ED 48 DC AB 09 D5 ED 5B
AC6F DA AB 3E FF B7 45 ED 52
AC77 3C 30 FA D1 CD ED AC 7B
AC7F 3C C1 C3 F9 AC C5 E5 CD
AC87 53 AC D5 E1 06 0A CD 1C
AC8F AD B7 28 02 10 F8 E1 C1
AC97 B7 20 01 C9 CB 77 3E 15
AC9F 20 02 3E 16 37 C9 F3 3E
ACA7 CF D3 11 3E C0 D3 11 3E
ACAF CF D3 13 3E F0 D3 13 3E
ACB7 B7 D3 11 3E 7F D3 11 3E
ACBF 08 D3 10 0E 01 CD 60 AD
ACC7 3E 28 D3 10 3E D0 D3 0C
ACCF E3 E3 DB 0C 3E AB ED 47
ACD7 3E 5C D3 11 3E 21 D3 10
ACDF FB ED 5E C9 3E 03 D3 0C
ACE7 FB 18 FE DB 0C C9 D3 0F
ACEF 3E 1B FB D3 0C 18 FE DB
ACF7 0C C9 D3 0E C9 C5 D5 E5
ACFF CD 3C AD 21 5B AD CD 4B
AD07 AD E1 0E 0F 3E BC E5 FB
AD0F D3 0C 18 FE DB 0C CD 4B
AD17 AD E1 D1 C1 C9 C5 D5 E5
AD1F CD 3C AD 21 5C AD CD 4B
AD27 AD E1 0E 0F 3E AC FB E5
AD2F D3 0C 18 FE DB 0C CD 4B
AD37 AD E1 D1 C1 C9 21 66 00
AD3F 11 54 AD 01 04 00 ED B0
AD47 C9 21 54 AD 11 66 00 01
AD4F 04 00 ED B0 C9 C3 7D 0C
AD57 E5 ED A2 ED 45 ED A3 ED
AD5F 45 E5 D5 C5 06 64 17 29
AD67 29 05 C2 65 AD 0D C2 63
AD6F AD C1 D1 E1 C9 F5 C5 E5
AD77 21 06 00 39 4E 23 46 0A
AD7F FE 18 20 0A 03 0A FE FE
ADB7 20 04 03 70 2B 71 E1 C1
AD8F F1 ED 4D FB CD A5 AC CD
AD97 E3 AB C9 AB AC CD E3 AB
    
```

von GÜNTER BÖHM

In seinem SPRITE-EDITOR hat Peter Brendel von BASIC aus auf mehrere Maschinenprogramme zugegriffen, indem er dem Befehl USR ein entsprechendes Argument beigab.

Im folgenden möchte ich Ihnen eine Möglichkeit skizzieren, wie Sie auf eine sogar etwas schnellere Art zwischen mehreren Maschinenprogrammen wählen können. Das Programm hat keinen praktischen Wert; es ist nur zur Demonstration und zum Test gedacht. Eine praktische Anwendung werde ich Ihnen demnächst in einem Verwaltungsprogramm geben, welches wir zur Berechnung unserer Vorsteuerabrechnung verwenden.

TEAP 180 Assembler - Source Listing

```

0001 !AUFRUF MEHRERER MASCHINENPROGRAMME
0002 !DURCH BASIC
0003 !G. Böhm Karlsruhe 3.3.84
0004 !VIELLEICHT WARE ES VORTEILHAFT, DIE REGIS
0005 !ZU RETTEN. HIER WEISS ICH NICHT, WAS BASI
0006 !VOR DEM AUFRUF MACHT.
0007 ;
BCB0 0010 ORG BCB0
BCB0 CDBBE9 0020 CALL #E908 !ARGUMENT VON BASIC
BCB3 1D 0048 DEC E
BCB4 2B1A 0050 JR Z TEST1
BCB6 1D 10 0060 DEC E
BCB7 2B26 0070 JR Z TEST2
BCB9 1D 0080 DEC E
BCBA 2B39 0090 JR Z TEST3 !UND SO WEITER ...
BCBC EF 0091 DEFB #EF
BCBD 46414C53 0092 DEFM *FALSCHES ARGUMENT
43404553
20415247
354D454E
54
BC9E 00 0093 DEFB 0
BC9F C9 0094 RET
BCA0 EF 0100 TEST1 DEFB #EF
BCA1 45525354 0110 DEFM *ERSTER TEST0
45522054
45535430
BCAD 00 0120 DEFB 0
BCAE C9 0130 RET
BCAF EF 0140 TEST2 DEFB #EF
BCB0 44494553 0150 DEFM *DIES IST DER ZWEITE
20495354
20444552
205A5745
495445
BCC3 00 0160 DEFB 0
BCC4 C9 0161 RET
BCC5 EF 0170 TEST3 DEFB #EF
BCC6 534E4420 0180 DEFM *UND HIER DER LETZTE
48494552
20444552
204C4554
5A5445
BCD9 00 0170 DEFB 0
BCDA C9 0200 RET
    
```

# SPRITE-Editor

von PETER BRENDEL

## USR-ROUTINEN

Nascom-User, die mit ihrem Rechner etwas Vernünftiges tun, können diesen Beitrag vergessen. Die USR-Routinen sind vor allem als Hilfe für Spieleprogrammierer gedacht. Eigentlich ist der Nascom ja alles andere als ein Spielcomputer; denn unsere Grafik ist, wie wir alle wissen, gelinde gesagt bescheiden. (Du mußt dir eben die neue Grafikkarte zulegen! Günter).

Durch die USR-Routinen wird die Grafik leider auch nicht besser. Allerdings kann man mit ihnen einige Sachen machen, die in Basic nicht möglich wären.

Vor der Beschreibung der USR-Routinen jedoch die Beschreibung des USR-Befehls.

Mit USR(xx) kann man das Basic-Programm verlassen, um ein Maschinenunterprogramm ausführen zu lassen. Dabei kann man sowohl einen Wert aus dem Basic ins Maschinenprogramm als auch umgekehrt übergeben.

Vorher muß man die Startadresse des Masch. Progr. in USRLOC (4100 dez) ein-"doken". In den meisten Fällen wird das wohl #C80 d.h. 3200dez sein. Mit "CALL #E98B" wird der Inhalt der Klammer ins DE-Register geladen. Will man ein Rechenergebnis ins Basic übergeben, läßt man das Low-Byte ins B, das High-Byte ins A-Register und springt mit "JP #F0F2" ins Basic zurück. (Es gibt noch die Möglichkeit mit "RET", aber darüber möchte ich mich nicht auslassen, weil ich nicht weiß, warum es funktioniert).

Jetzt aber die Beschreibung der USR-Routinen.

Grundsätzlich unterscheidet das Programm zwei Eingaben:

USR (kleiner 255) und USR (größer 255)

-255 wird als ADRESSE behandelt. Das Programm hält die Adresse fest, und alle späteren Eingaben beziehen sich auf diese Adresse.

USR (0-16) sind verschiedene Hilfsprogramme.

USR (0) ist die INKEY-ROUTINE (geklaubt bei Pack-Mann von Herrn Schröder)

USR (12) verschiebt die Zeilen 1-15 um eine Stelle nach links, USR (13) nach rechts,

USR (14) kopiert den Speicherbereich (1024 Bytes), dessen Anfang vorher mit USR (xxxx) festgelegt wurde, in den Bildschirm.

USR (15) füllt einen Bereich von 1024 Bytes, dessen Anfang vorher mit USR (xxxx) festgelegt wurde, mit Spaces.

USR (16) verschiebt die Zeile 16 um eine Stelle nach links (Laufschrift).

USR (1-10) rufen Bilder auf, die aus einzelnen Grafikzeichen aufgebaut sind. Diese Bilder müssen am Anfang des Programms einmal definiert werden. Dazu werden die horizontalen und vertikalen Abmessungen in die Speicherzellen 3541=Horiz. und 3542=Vertik. eingetragen. (POKE)

Dann das eigentliche Bild im gleichen Raster in der Reihenfolge, wie man liest, wobei die linke obere Ecke der erste Speicherplatz des entsprechenden Bildes ist (SIEHE TABELLE).

Hat ein Bild mehr als 25 Speicherplätze, darf die nächste Bild-Nr nicht verwendet werden.

Das Bild wird dort erzeugt, wo es vorher mit USR (xxxx) festgelegt wurde.

USR (11) füllt den Raster, der vorher mit USR (xxxx) festgelegt wurde, mit Spaces. (Bild löschen)

## TABELLE

BILD 1	3543	BILD 2	3569
BILD 3	3595	BILD 4	3621
BILD 5	3647	BILD 6	3673
BILD 7	3699	BILD 8	3725
BILD 9	3751	BILD 10	3777

## C80 EDB

0C80	DF	5F	DF	5B	FE	00	20	18	3A
0C88	7A	FE	00	20	13	21	01	0C	6D
0C90	06	09	36	00	23	10	FB	DF	EE
0C98	62	20	01	47	AF	C3	F2	F0	C2
0CA0	7A	FE	00	20	4C	7B	FE	01	0A
0CA8	CA	5C	0D	FE	02	CA	62	0D	20
0CB0	FE	03	CA	68	0D	FE	04	CA	C8
0CB8	6E	0D	FE	05	CA	74	0D	FE	8B
0CC0	06	CA	7A	0D	FE	07	CA	80	72
0CC8	0D	FE	08	CA	86	0D	FE	09	4B
0CD0	CA	8C	0D	FE	0A	CA	92	0D	B0
0CD8	FE	0B	CA	98	0D	FE	0C	28	8E
0CE0	21	FE	0D	28	3C	FE	0E	28	B0
0CE8	0D	FE	0F	28	53	FE	10	28	BF
0CF0	5F	ED	53	D3	0D	C9	11	00	55
0CF8	08	2A	D3	0D	01	BF	03	ED	C6
0D00	B0	C9	7A	FE	00	20	EF	06	13
0D08	0F	3E	20	21	3A	08	11	40	36
0D10	00	77	19	10	FC	21	01	08	E3
0D18	11	00	08	01	BF	03	ED	B0	9E
0D20	C9	7A	FE	00	20	D0	06	0F	73
0D28	3E	20	21	09	08	11	40	00	16
0D30	77	19	10	FC	21	BE	08	11	D4
0D38	BF	0B	01	BF	03	ED	B8	C9	40
0D40	2A	D3	0D	01	BF	03	ED	5B	62
0D48	D3	0D	13	36	20	ED	B0	C9	04
0D50	11	C9	0B	21	CA	0B	01	30	69
0D58	00	ED	B0	C9	DD	21	D7	0D	AD
0D60	18	52	DD	21	F1	0D	18	4C	37

```

0D68 DD 21 0B 0E 18 46 DD 21 E8
0D70 25 0E 18 40 DD 21 3F 0E 53
0D78 18 3A DD 21 59 0E 18 34 88
0D80 DD 21 73 0E 18 2E DD 21 50
0D88 8D 0E 18 28 DD 21 A7 0E 23
0D90 18 22 DD 21 C1 0E 18 1C D8
0D98 3A D6 0D 47 2A D3 0D 16 29
0DA0 00 3A D5 0D 4F 3E 40 91 27
0DA8 5F 3E 20 77 23 0D 20 FB 34
0DB0 19 10 EE C9 3A D6 0D 47 01
0DB8 2A D3 0D 16 00 3A D5 0D 01
0DC0 4F 3E 40 91 5F DD 7E 00 E5
0DC8 77 23 DD 23 0D 20 F6 19 AB
0DD0 10 EB C9 00 00 00 00 86 27
0DD8 20 20 20 20 20 20 20 E5
0DE0 20 20 20 20 20 20 20 ED
0DE8 20 20 20 20 20 20 20 F5
0DF0 20 20 20 20 20 20 20 FD
0DF8 20 20 20 20 20 20 20 05
0E00 20 20 20 20 20 20 20 0E
0E08 20 20 20 20 20 20 20 16
0E10 20 20 20 20 20 20 20 1E
0E18 20 20 20 20 20 20 86 FF 6B
0E20 01 FF 08 FF 00 86 20 20 FB
0E28 86 20 20 20 20 20 20 9C
0E30 20 20 20 20 20 20 20 3E
0E38 20 20 20 20 20 20 20 46
0E40 20 20 20 20 20 20 20 4E
0E48 20 20 20 20 20 20 20 56
0E50 20 20 20 20 20 20 20 5E
0E58 20 20 20 20 20 20 20 66
0E60 20 20 20 20 20 87 20 D5
0E68 20 FF 00 FF 00 FF 00 FF 92
0E70 0B FF 00 FF 00 FF 00 FF 85
0E78 00 FF E0 FF 00 FF 00 FF 62
0E80 14 FF 04 FF 00 FF 00 FF A2
0E88 00 FF 00 FF 00 FF 00 FF 92
0E90 09 FF 0D FF 00 FF 00 FF B0
0E98 00 FF 00 FF 00 FF 00 FF A2
0EA0 0F FF 00 FF 00 FF 00 FF B9
0EA8 00 FF 90 FF 00 FF 00 FF 42
0EB0 02 FF 01 FF 00 FF 00 FF BD
0EB8 00 FF 70 FF 00 FF 00 FF 32
0ECO 03 FF 04 FF 00 FF 00 FF D1
0EC8 00 FF 00 FF 00 FF 00 FF D2
0ED0 14 FF 09 FF 00 FF 00 FF F7
0ED8 00 FF 04 CB 92 D0 49 6F CE

```

Mit dem SPRITE-EDITOR kann man Bilder erzeugen, ohne mühsame Eintragung; das wird alles vom Programm erledigt.

Der SPRITE-EDITOR benutzt die USR-Routinen, um die reservierten Speicherplätze für die Optionen Bild 1...10 mit den entsprechenden Bildern zu programmieren.

Vor Starten des Programms also zuerst die USR-Routinen laden.

Nach Start des Basic-Programms erscheint auf der rechten Seite des Bildschirms ein rechteckiges Feld, auf der linken Seite der gesamte Zeichenvorrat des NASCOM. Ein blinkender Punkt dient als Cursor.

Durch Stellen des Cursors auf ein Zeichen und "ENTERN" wird es übernommen; das Blinken erlischt.

Jetzt kann das Zeichen in das Feld verschoben, und mit "ENTER" plaziert werden (cursor blinkt wieder).

Auf diese Weise setzt man Zeichen an Zeichen und erzeugt so ein Bild.

TASTE "E" erfasst das Bild und legt die Daten im Speicher ab. Dabei werden die nötigen Basic-Befehle für die spätere Verwendung dieses Bildes angezeigt (ausgenommen USR(adresse) für die Plazierung auf dem Bildschirm)

TASTE "W" bedeutet "Weiter"; das nächste Bild kann erzeugt werden.

Mit TASTE "B" wird ein bereits erfaßtes Bild nochmals ins Feld gebracht. Aber VORSICHT! Zuerst Cursor auf den Platz stellen, wo das Bild entstehen soll. Dieses Bild kann dann modifiziert werden und erhält nach "E" die neueste Zeilen Nr.

Das Ändern eines bereits erfaßten Bildes ist nicht möglich.

TASTE "A" bedeutet Abspeichern. Das Programm zeigt noch den Speicherbereich an, der unter NASSYS auf Band abgespeichert werden muß, und springt dann in den Monitor.

```

10 CLS:D=29:POKE3120,40:HO=2540
20 FORI=1TO10:READBS(I):NEXT
30 DATA3543,3569,3595,3621,3647
40 DATA3673,3699,3725,3751,3777
50 RN=1
60 DOKE4100,3200:V1=12048:V2=13065:A=27:B=13
70 P=V1+29:U=12059:W=12075:C=1:G1=46
80 S=USR(V2):S=USR(15):S=USR(V1):S=USR(15)
90 GOSUB760
100 DOKE3113,2059
110 S=USR(14):PRINT"SPRITE-EDITOR
120 FORI=1TO15
130 FORJ=UTOW:IFG=256THENG=255
140 POKEJ,G:G=C+1:NEXT
150 U=U+64:W=W+64:NEXT
160 POKEJ,G
170 S=USR(14):P=P-10001
180 S=USR(0):IFC=0THENG1=46
190 IFS=17THENP=P-1:GOTO280
200 IFS=18THENP=P+1:GOTO280
210 IFS=19THENP=P-64:GOTO280
220 IFS=20THENP=P+64:GOTO280
230 IFS=65THEN1060
240 IFS=66THEN920
250 IFS=69THEN350
260 IFS=87THEN850
270 IFS=13THENC=C*-1:ONSGN(C)+2GOSUB300,10,310
280 G=PEEK(P):POKEP,G1:N=80:GOSUB340:POKEP,G
290 GOTO180
300 G1=PEEK(P):GOSUB340:RETURN
310 POKEP,G1:POKEP+10000,G1:POKEP+11017,G1
320 N=800:G1=46
330 GOSUB340:RETURN
340 FORI=1TON:NEXT:RETURN
350 IFPEEK(2830)=32THENS=87:GOTO190
360 P=V2+29:GOSUB760:S=USR(V2):S=USR(14)
370 SCREEN1,1:PRINT"ERFASSEN":PRINT
380 E1=V1-9970:E2=E1+896
390 FORI=E1TOE1+26
400 IFPEEK(I+64)-32THENI=E1+26:NEXT:GOTO470
410 NEXT
420 E1=E1+64:GOTO390
430 FORI=E2TOE2+26
440 IFPEEK(I-64)-32THENI=E2+26:NEXT:GOTO480
450 NEXT
460 E2=E2-64:GOTO430

```

# Nachtrag

VON KLAUS MOMBAUR

```

470 E3=I+64:GOTO430
480 B=(E2-E1)/64-1:E1=E1+63:E2=E2-65
490 FORI=E1TOE2STEP64
500 IFPEEK(I+1)<-32THENQ1=E1:I=E2:NEXT:GOTO530
510 NEXT
520 E1=E1+1:E2=E2+1:GOTO490
530 E1=E3:E2=E1+(B*64)
540 FORI=E1TOE2STEP64
550 IFPEEK(I-1)<-32THENQ2=E1:I=E2:NEXT:GOTO580
560 NEXT
570 E1=E1-1:E2=E2-1:GOTO540
580 A=Q2-Q1-1
590 IFA<1ORB<1THENPRINT" NIX":S=76:GOTO190
600 PRINT"HORIZONTAL";A
610 PRINT"VERTIKAL ";B
620 P=Q1-64:GOSUB760:P=P+10000:GOSUB760:P=2076
630 FORJ=1TOB
640 FORI=Q1+1TOQ2-1:POKEBS(BN),PEBK(I)
650 BS(BN)=BS(BN)+1:NEXT
660 Q1=Q1+64:Q2=Q2+64:NEXT
670 POKE3541,A:POKE3542,B:A(BN)=A:B(BN)=B
680 PRINT:PRINT"DOKE 3541,";B*256+A
690 PRINT"BILD";BN;PRINT"=USR(";BN)";P=HO
700 WN=BN:BZ(BN)=BN
710 FORI=QTOA*B/26:BN=BN+1:NEXT:PRINT
720 PRINT" ABSCHREIBEN!":PRINT
730 IFBN=10THENPRINT" LETZTES BILD":GOTO750
740 PRINT"naechstes Bild =";BN:DOKE3113,2058
750 G1=46:GOTO180
760 POKEP,144
770 FORI=P+1TOP+A:POKEI,152
780 NEXT:POKEI,145
790 POKEP+B*64+64,146
800 FORI=P+1+B*64+64TOP+A+B*64+64:POKEI,152
810 NEXT:POKEI,147
820 FORI=P+64TOV2+1024STEP64
830 IFPEEK(I)<-32THENI=V2+1024:NEXT:RETURN
840 POKEI,148:POKEI+1+A,148:NEXT:RETURN
850 GOSUB1030
860 A=27:B=13:P=V1+29:GOSUB760:P=HO
870 SCREEN1,3:PRINT" WEITER"
880 N=500:GOSUB340
890 S=USR(V2):S=USR(15)
900 S=USR(V1):S=USR(14)
910 GOTO190
920 IFPEEK(2830)=32THENS=87:GOTO190
930 GOSUB1030:BI=BN
940 SCREEN1,3:INPUT"BILD-Nr. ";BN
950 IFBN=<B1ORBN=>0THENGOSUB1030:GOTO940
960 FORI=1TO WN:IFBZ(I)<-BNTHENNEXT
970 IFBZ(I)=BNTHENI=WN:NEXT:GOTO990
980 I=WN:GOTO940
990 POKE3541,A(BN):POKE3542,B(BN)
1000 S=USR(P):S=USR(BN):S=USR(P+10000):S=USR(BN)
)
1010 S=USR(P+11017):S=USR(BN)
1020 BN=B1:GOTO900
1030 SCREEN1,2:PRINT" "
1040 PRINT" "
1050 RETURN
1060 Q=BS(WN):REM DEZ-HEX
1070 FORI=1TO7:READA:POKEQ+I,A:NEXT
1080 DATA 205,139,233,235,223,102,201
1090 DOKE4100,Q+1:CLS
1100 PRINT"ABSPEICHERN UND AUFHEBEN"
1110 PRINT"Einfach Cursor drueber + Enter"
1120 PRINT:PRINT"W O C 80 ";
1130 S=USR(Q):DOKE4100,3200:PRINT:PRINT
1140 MONITOR

```



```

521 REM Aenderungen fuer *Buchfuehrung*
522 REM =====
523 REM
524 REM Journal 10/11 1983 v. K.Mombaur
525 REM Bildschirmausgabe der Daten per Masch.
526 REM Groe Datenmengen schnell auflisten
527 REM
529 RESTORE 530
530 DATA 29747
531 DATA 8481
532 DATA 2058
533 DATA 10530
534 DATA 10764
535 DATA 3255
536 DATA 16446
537 DATA 14146
538 DATA 4741
539 DATA 8353
540 DATA 10761
541 DATA 12827
542 DATA 3486
543 DATA 11807
544 DATA 4584
545 DATA 10987
546 DATA 3113
547 DATA 14593
548 DATA 4853
549 DATA 14402
550 DATA 10983
551 DATA 3567
552 DATA 25975
553 DATA 29801
554 DATA 29285
555 DATA 16160
556 DATA 8448
557 DATA 1635
558 DATA 15886
559 DATA 2277
560 DATA 4926
561 DATA 4343
562 DATA 11784
563 DATA 4629
564 DATA 10587
565 DATA 10996
566 DATA 3902
567 DATA 24451
568 DATA 386
569 DATA 10282
570 DATA 4840
571 DATA 6304
572 DATA 9207
573 DATA 9189
574 DATA 12930
575 DATA 6088
576 DATA 816
577 DATA 6187
578 DATA 7932
579 DATA 6353
580 DATA 11029
581 DATA 6
582 DATA 307
583 DATA 7923
584 DATA 16081
585 DATA 28664
586 DATA 24451
587 DATA 574
588 DATA 20352
589 DATA 6
590 DATA 20243
591 DATA 3390
592 DATA 13833
593 DATA 386
594 DATA 14290
595 DATA 8964
596 DATA 2024
598 FORJ=3458TO3588STEP2:READB:DOKEJ,B:NEXT
599 REM 0080 - 0E04
OK
8301 IF DR=1THEN8303
8302 DOKE4100,3456+A:USR(6):GOSUB200:RETURN
8303 L=0
OK
8355 IFB=64THENIFSD=0THENGOSUB8400
8356 IFB=64THENGOSUB200:RETURN
8360 BN=BN+CHR$(B):TB=TB+1:GOTO 8345
OK
8425 IFE1="*THENE1="*-
OK
8434 PRINT:PRINTTAB(15)*"-K- wenn korrigieren"
8435 PRINTTAB(15)*"-S- wenn speichern"
8436 GOSUB 8100
8437 IFE="*K"THEN 8400
8438 IFE("<")S"THEN 8436
8439 PRINTCHR$(19);CHR$(27);CHR$(19);CHR$(27)
8440 IFE2=0THENE2="*":RETURN
8444 E2=STR$(E2)
OK
8560 IFLEN(E2)=4THENE2="* 0*RIGHT$(E2,3)
8565 FORA=1TOLEN(E2)
OK
8870 TB=UE+1:SD=1
8880 REM CLS:GOSUB8200:PRINT
8890 GOSUB8320:SD=0:RETURN
OK

```

# Star Wars

von W. MAYER-GÖRR

\* STAR WARS \*

Ziel des Spiels ist es, innerhalb einer vorgegebenen Sternzeit alle Klingons abzuschiessen, die das Universum bedrohen.

Der Spieler ist Kapitän des Raumschiffs Enterprise (E).

Das Universum ist in 8 \* 8 Quadranten eingeteilt. Jeder Quadrant wiederum besteht aus 8 \* 8 Sektoren.

Zum Manövrieren dienen folgende Befehle:

- 0 Motor an  
Es folgt die Frage nach der Richtung, man gibt dann eine Ziffer zwischen 0 und 7 ein.

```

      0
     7 1
    6 E 3
     5 3
      4
  
```

Dan wird nach der Weite gefragt. Will man innerhalb des Quadranten fliegen, gibt man einen Punkt gefolgt von einer Ziffer und dann NEWLINE ein.

Beim Flug über Quadranten besteht die Eingabe nur aus einer Ziffer und NEWLINE.

ACHTUNG: Es ist gefährlich, die Grenze der Galaxie zu überschreiten!

- 1 SHORT RANGE SCANNER  
Zeigt ein Bild des aktuellen Quadranten und den augenblicklichen Status.  
E = Enterprise  
K = Klingon  
\* = Stern  
S = Supernova  
. = frei  
B = Basis

Die Basis dient zum Auffrischen der Energie und zum Nachladen der Torpedos. Dazu muß man auf ein Feld neben der Basis fliegen und Befehl 1 betätigen.

- 2 LONG RANGE SCANNERS  
Zeigt den aktuellen und die umliegenden Quadranten. Der Inhalt der Quadranten wird als vierstellige Zahl angezeigt.  
Einer : Klingons  
Zehner : Sterne  
Hunderter : Basis  
Tausender : Supernova

Fliegt man in einen Quadranten mit einer Supernova, wird die Enterprise leider zerstört.

Die Grenze der Galaxie wird mit ++++ symbolisiert.

- 3 PHASER  
Eine Waffe, die Energie in alle Richtungen ausstrahlt. Dient zum Zerstören von Klingons. Schild muß unten sein!
- 4 TORPEDO  
Bei Spielbeginn hat man 15 Torpedos, die Richtung wird nach dem Kompaß wie bei 1 eingegeben.
- 5 DAMAGE REPORT  
Zeigt, wie lange verschiedene Geräte defekt sind.
- 6 SCHILD HOCH  
Das Schild schützt teilweise vor den Angriffen der Klingons.
- 7 SCHILD UNTEN  
MUß vor Befehl 3 erfolgen.
- 8 TELEPORTER  
Eine noch nicht ganz ausgereifte Entwicklung für die Enterprise. Nach Ablauf einer bestimmten Zeit kann die Enterprise direkt zum Quadranten mit der Basis gebracht werden. Funktioniert aber nicht immer!
- 9 SELBSTZERSTÖRUNG  
Nach Eingabe des Kennworts kann man sich selbst zerstören. Bei Spielbeginn wird ein Kennwort erfragt.
- 1 Zeigt die Karte der Galaxie. Alle Quadranten, die einmal von Befehl 2 erfaßt wurden, geben die Zahl der Objekte in ihnen an.
- 1 STERNZEIT  
So viel Sternzeit bleibt noch.

Das Programm existiert leider noch nicht auf Kassette, da mein Interface nicht funktionsfähig ist. Eine Version auf Diskette (CLD-DOS) überspiele ich gerne. Das Programm läuft unter NAS-SYS und wird mit E1000 gestartet.

Star Wars

```

1000  E3 00 11 3E 20 FE 0A 20
1009  02 3E 00 F7 C9 CF FE 00
1010  20 02 3E 0A C9 C7 03 05
1018  55 7E FE 3D 23 18 0D A7
1020  C2 21 4F 11 CA 55 ED 04
1028  51 C3 5C 4F 11 CA 55 21
1030  8B 55 CD 4C 3E DA 4A 50
1038  FE 2C C2 4E 4F 21 A6 55
1040  CD 4C 3E DA 4A 50 FE 3D
1048  C2 68 50 CD 04 51 CD D0
1050  50 DA EA 4E 21 A6 55 CD
1058  4C 3E 21 A6 55 11 CA 50
1060  01 6A 55 FF 2B 3A 6A 55
1068  E6 01 CA 15 50 21 A6 55
1070  11 CA 50 3E 02 E5 FF 22
1078  E1 D2 8A 4F CD A1 50 C3
1080  EA 4E 3A 3C 4E 21 3D 4E
1088  96 32 4F 4E 21 B7 43 11
1090  BF 20 01 99 00 CD AA 18
1098  3A 26 4E E6 80 C2 CF 4F
10A0  3A 25 4E A7 C2 CF 4F 2A
10A8  D0 20 E8 00 FF 01 FF 01
10B0  00 01 01 00 01 FF 01 FF
10B8  00 FF FF 2E 2A 4B 42 A2
10C0  13 63 15 3B 16 89 16 6B
10C8  17 F5 17 3A 18 30 1B 69
10D0  18 AE 18 05 18 4B 19 CD
  
```

1008	05 10 23 7E FE 04 20 F7	1360	10 C9 21 57 10 7B 07 CD	1508	16 21 27 1D CD DB 10 2A	1670	18 21 09 1E C3 1A 12 3A
1009	C9 23 CD EA 10 18 FA CD	1361	BB 1B 7A FE 04 FA 73 13	1509	41 10 CD 92 1A 3A 40 10	1671	43 10 B7 C2 17 12 CD F3
1010	EA 10 E5 21 FF 1B CD DB	1362	23 06 04 57 5F B7 7E 2B	1510	B7 2B 05 21 3B 1D 1B 03	1672	10 FE B0 3B 04 21 43 10
1011	10 E1 C9 E5 C3 21 14 10	1363	05 0F 0F 1D 20 FB E6 03	1511	21 41 1D CD DB 10 C3 1D	1673	34 CD F3 10 FE B0 30 FE
1012	06 08 7E 07 AE 17 17 23	1364	C9 CD AE 1B 3E 03 B9 20	1512	1608 12 CD EA 10 2A 31 10 EB	1674	2A 18 10 3E 0A BC 2B 06
1013	7E 17 77 2B 7E 17 77 05	1365	05 EB 22 1A 10 E0 CD 67	1513	1610 95 2A 2E 10 CD AB 1B 3E	1675	22 2C 10 C3 C7 14 CD AE
1014	20 F0 C1 E1 C9 31 00 10	1366	13 20 EE 79 15 FA 9C 13	1514	1611 45 2B 0A CD 62 13 21 BB	1676	1B EB 22 2C 10 21 E3 1E
1015	CD E7 10 21 01 1C CD E2	1367	07 07 1B FB B6 77 05 20	1515	1620 10 CD BB 1B 7E CD 05 10	1677	CD E2 10 C3 37 14 21
1016	10 06 93 21 18 10 CD 8C	1368	E0 C9 CD E3 14 21 39 1E	1516	1628 CD 03 10 D1 14 7A FE 0B	1678	1C CD E2 10 21 E6 1F 46
1017	1B CD EA 10 21 15 1C CD	1369	CD E2 10 CD D2 12 FE 0A	1517	1630 20 DE 7B 3C 32 31 10 C9	1679	23 CD 0D 10 8B 20 10 FE
1018	E2 10 CD 0D 10 FE 4B 2B	1370	F2 1D 12 32 22 10 3A 20	1518	1638 3A 3E 10 B7 C2 17 12 CD	1680	0A 20 F4 21 AD 1F CD E2
1019	04 21 34 10 34 21 6B 10	1371	10 B7 20 0B 3C 32 10 10	1519	1640 EA 10 21 9D 1D CD E2 10	1681	10 CD 0A 10 C3 D1 1B 21
1020	0E 40 16 0D CD F3 10 FE	1372	3A 3A 10 B7 C2 17 12 CD	1520	1648 CD 2C 1B CD E7 10 2A 2C	1682	03 1B C3 1A 12 CD EA 10
1021	A0 3B 10 14 FE E0 3B 0B	1373	0D 10 FE 1B CA 10 12 FE	1521	1650 10 EB 15 1D 0E 03 7B FE	1683	21 EF 1E CD E2 10 CD 2C
1022	14 FE F7 3B 06 14 FE FC	1374	0A 20 F4 CD FC 14 2A 2E	1522	1658 FF 2B 04 FE 0B 20 09 06	1684	1B 1E 00 CD EA 10 16 00
1023	3B 01 14 3A 34 10 B7 2B	1375	10 EB CD 11 15 2B 29 05	1523	1660 03 21 E9 1D CD DB 10 05	1685	05 21 47 1D CD 65 13 D1
1024	0E CD F3 10 FE CD 3B 07	1376	CD 62 13 C2 72 14 D1 EB	1524	1668 20 F7 1B 19 06 03 7A FE	1686	20 0B 21 E9 1D CD DB 10
1025	14 14 FE F0 3B 01 14 72	1377	22 2E 10 CD F3 10 FE 80	1525	1670 FF 2B 1C FE 0B 2B 1B CD	1687	1B 03 CD 97 16 14 3E 0B
1026	3A 69 10 B2 32 69 10 CD	1378	3B 0A 06 01 CD 30 15 06	1526	1678 1E 19 CD 97 16 14 05 20	1688	BA 20 E5 1C 3E 0B DB 20
1027	F3 10 E6 3B B6 77 23 0D	1379	03 CD 59 15 21 1E 10 35	1527	1680 ED 7A 06 03 57 CD EA 10	1689	BA 2A 1B 10 3E 0A BC 2B
1028	20 CD CD EA 10 21 34 1C	1380	20 D4 3A 10 10 B7 2B 32	1528	1688 1C CD 20 CA C3 1D 12 21	1690	0A EB 21 34 1C CD E2 10
1029	CD E2 10 CD AE 1B 62 6B	1381	3A 22 10 32 1E 10 2A 2C	1529	1689 E9 1D CD DB 10 1B E6 CD	1691	0C 5A 1B C3 1D 12 D5 21
1030	22 1B 10 CD 5A 1B CD 61	1382	10 EB CD 11 15 CA BC 14	1530	1698 61 1B 7E E6 80 07 CD 1F	1692	47 10 CD 65 13 3E 03 15
1031	1B 3E 40 B6 77 CD B4 12	1383	EB 22 2C 10 3A 1D 10 B7	1531	16A0 1B 7E E6 40 0F 0F CD 26	1693	FA 2F 19 07 07 1B F8 B6
1032	CD F3 10 32 23 10 CD F3	1384	2B 49 3E 01 32 1C 10 06	1532	16A8 1B 7E E6 38 07 CD 26 1B	1694	77 D1 C9 CD 61 1B 7E 07
1033	10 E6 1F F6 04 32 26 10	1385	06 CD 30 15 06 1E CD 59	1533	16B0 7E E6 07 CD 1F 1B C3 03	1695	30 0B 3E 53 CD 05 10 C3
1034	CD F3 10 E6 0F C6 15 47	1386	15 21 1E 10 35 20 07 CD	1534	16B8 10 3A 3D 10 B7 C2 17 12	1696	03 10 0F E6 07 CD 1F 1B
1035	3A 34 10 B7 7B 2B 08 CD	1387	F2 12 AF 32 30 10 2A 2C	1535	16C0 3A 40 10 B7 2B 06 21 6B	1697	C3 03 10 21 5E 1C CD E2
1036	F3 10 E6 0F C6 15 80 32	1388	10 ED 2A 1B 10 CD AB 1B	1536	16C8 1D C3 1A 12 21 7C 1D CD	1698	10 21 27 10 3A 23 10 96
1037	23 10 CD AE 1B EB 22 2C	1389	C2 1D 12 3A 1B 10 3C 21	1537	16D0 ED 10 2A 35 10 CD 92 1A	1699	47 3A 1F 10 4F B7 20 03
1038	10 CD F2 12 21 47 1C CD	1390	2F 10 96 FE 03 D2 1D 12	1538	16D8 21 92 1D CD E2 10 CD CE	1700	0E 0A 04 05 6B 26 00 CD
1039	E2 10 21 E6 1F CD 0E 10	1391	3A 1A 10 3C 2B 96 FE 03	1539	16E0 1A EB 2A 35 10 CD AB 1B	1701	92 1A 3E 2E CD C5 10 3E
1040	77 23 FE 0A 20 F7 CD E7	1392	D2 1D 12 3E 02 32 30 10	1540	16E8 3A 06 21 A3 1D C3 1A 12	1702	0A 91 CD 1F 1B C3 1D 12
1041	10 21 32 1C CD E2 10 CD	1393	C3 1D 12 06 07 CD 30 15	1541	16F0 CD F3 10 FE FA 3B 0B 21	1703	CD AE 1B CD 61 1B 7E E6
1042	11 1B 21 0D 1D CD E2 10	1394	1B C5 FE 02 CA 83 14 D1	1542	16F8 B3 1D CD E2 10 1B 5C 3A	1704	07 47 3A 69 10 90 32 69
1043	CD 0F 1A 21 5E 1C CD E2	1395	21 42 1E CD E2 10 CD 5A	1543	1700 67 1D 07 2B 56 05 2A 2B	1705	10 36 80 CD EA 10 21 EF
1044	10 2A 23 10 CD 92 1A 21	1396	1B 1B 87 CD 93 1B D1 21	1544	1708 10 CD AB 1B 30 3F 21 00	1706	1D CD E2 10 CD 1E 19 CD
1045	6B 1C CD E2 10 CD 2C 1D	1397	51 1E CD E2 10 AF 32 1D	1545	1710 00 22 2B 10 21 4D 1F CD	1707	5A 1B 2A 2C 10 CD AB 1B
1046	21 EF 1C CD E2 10 CD 1A	1398	10 3C 32 1E 10 21 67 10	1546	1718 E2 10 21 BD 1E CD E2 10	1708	CA C3 1B 2A 1B 10 CD AB
1047	10 CD 37 1B C3 1D 12 21	1399	35 23 34 23 35 CD 5A 1B	1547	1720 3A 67 10 32 68 10 47 3A	1709	1B C2 1D 12 21 0A 0A 22
1048	06 1E CD E2 10 31 00 10	1400	21 FF 1D CD E2 10 06 72	1548	1728 69 10 90 32 69 10 AF 32	1710	1B 10 C3 1D 12 2A 45 10
1049	21 23 10 3A 27 10 BE F2	1401	CD 6B 1B 21 BD 1E CD E2	1549	1730 67 10 CD BF 1A 21 57 10	1711	3A 6B 10 47 3A 37 10 07
1050	CB 1B 3A 69 10 B7 CA D9	1402	10 D5 CD BF 1A CD 63 1A	1550	1738 0E 10 06 04 7E 0F 3B 02	1712	07 07 80 47 3A 3B 10 B7
1051	1B 3A 44 10 B7 2B 0C CD	1403	D1 C3 E7 13 21 66 1E CD	1551	1740 E6 FE 0F 05 20 F7 77 23	1713	2B 10 7E E6 BF 77 3E 0A
1052	EA 10 21 77 1C CD E2 10	1404	E2 10 3A 39 10 3E FE 03	1741	1748 0B 20 EF 1B 0D CD C6 1A	1714	32 19 10 32 1B 10 AF 32
1053	C3 C3 1B CD B5 19 3A 30	1405	20 0C CD EA 10 21 79 1E	1742	1750 19 22 2B 10 21 BF 1D CD	1715	10 37 10 7E 90 77 AF 32 6B
1054	10 FE 02 2B 4F 3A 67 10	1406	CD E2 10 C3 D1 1B 32 39	1743	1758 E2 10 D1 2A 35 10 CD C6	1716	10 32 37 10 32 3B 10 C9
1055	B7 2B 02 3E 01 32 30 10	1407	10 3A 1C 10 B7 CA 3A 14	1744	1760 1A 19 22 35 10 C3 1D 12	1717	3A 67 10 B7 C8 CD F3 10
1056	CD F3 10 FE FD 3B 0E 21	1408	C3 37 14 AF 32 1B 10 32	1745	1768 3A 3E 10 B7 C2 17 12 3A	1718	FE B0 D0 2A 2B 10 7C B7
1057	B3 1C CD E2 10 21 3F 10	1409	1C 1D 21 32 1E CD E2 10	1746	1770 2A 10 B7 20 06 21 8D 1E	1719	20 0E 16 00 7D FE 0F 30
1058	34 34 C3 30 10 CD F3 10	1410	CD D2 12 32 21 10 FE 0B	1747	1778 C3 1A 12 C3 F3 10 E6 0F	1720	0D CD F3 10 E6 0F 1B 09
1059	FE 8D D2 7B 19 CD EB 19	1411	F9 C3 1D 12 3A 22 10 B7	1748	1780 F6 04 32 22 10 21 2A 10	1721	1A0B CD F3 10 E6 03 57 CD F3
1060	3A 34 10 B7 FA CD 1B 20	1412	CA 1D 12 32 1E 10 3A 1D	1749	1788 35 CD E3 14 CD FC 14 2A	1722	10 3F CD AB 1B 3B 0F 21
1061	0C 3A 35 10 B7 CA CD 1B	1413	10 B7 CB 3E 0F 32 1E 10	1750	1790 2E 10 EB CD 11 15 20 06	1723	0A 00 19 EB 3A 40 10 B7
1062	3E 03 32 30 10 3A 42 10	1414	C9 21 AB 10 3A 21 10 07	1751	1798 21 9B 1E C3 1A 12 CD EA	1724	20 1F 62 6B CD E7 10 9D
1063	B7 F2 9C 12 21 40 10 06	1415	CD BB 1B 7A 86 57 23 7B	1752	17A0 10 CD 5A 1B 35 CD 62 13	1725	CD 92 1A 01 21 9F 1F CD
1064	03 CD 8C 1B CD EA 10 21	1416	86 5F 7A FE FF CD FE 0B	1753	17AB 20 09 D1 21 1E 10 35 2B	1726	DB 10 2A 35 10 CD C6 1A
1065	A5 1C CD E2 10 CD D2 12	1417	CB 7B FE FF CD FE 0B C9	1754	17B0 E7 1B E0 47 CD 93 1B D1	1727	19 22 35 10 06 FA C3 6B
1066	07 21 BF 10 CD BB 1B 5E	1418	3A 1F 10 60 32 1F 10 FE	1755	17B8 21 A5 1E CD E2 10 7B FE	1728	1B CD EA 10 21 CA 1F CD
1067	23 56 EB E9 06 07 21 3A	1419	0A FB D6 0A 32 1F 10 2A	1756	17C0 02 20 1A 21 67 10 35 23	1729	EB 10 2A 41 10 CD C6 1A
1068	10 CD 8C 1B 3E 0B 32 35	1420	2B 10 23 22 25 10 21 27	1757	17C8 3A 23 35 21 B5 1E CD DB	1730	19 22 41 10 8B EB 2A 35
1069	10 32 41 10 3E 0B 32 3A	1421	10 34 21 3A 10 0E 06 35	1758	17D0 10 CD E1 10 CD 8F 1A CD	1731	10 19 22 35 10 21 B0 1C
1070	10 32 42 10 3E 0F 32 2A	1422	F2 54 15 34 23 CD 20 F7	1759	17D8 63 1A C3 10 12 FE 01 20	1732	1A60 C3 E2 10 2A 2B 10 5A 5D
1071	10 C9 AF 32 20 10 CD 0D	1423	C9 21 35 10 7E 90 77 D0	1760	17E0 0C 32 37 10 21 CD 1E CD	1733	3A 67 10 FE 02 FA 7B 1A
1072	10 FE 1B CA 10 12 FE 2E	1424	23 35 C9 3A 3B 10 B7 C2	1761	17E8 DB 10 C3 10 12 32 3B 10	1734	FE 0A FA 7B 1A CD 0A 1B
1073	2B 0F D6 30 FA EA 12 FE	1425	17 12 CD EA 10 3A 30 10	1762	17F0 21 D3 1E 1B F2 CD EA 10	1735	CD 0A 1B CD 0A 1B CD C6
1074	0C FB 21 BE 1C CD DB 10	1426	FE 02 CC B4 12 AF 32 31	1763	17F8 21 12 1E CD E2 10 0E 00	1736	1A 19 11 0A 00 CD AB 1B
1075	1B E0 06 11 21 57 10 CD	1427	10 2A 2C 10 EB CD 1E 19	1764	1800 21 3A 10 79 CD BB 1B 7E	1737	30 01 EB 22 2B 10 C9 2A
1076	8C 1B 2A 2C 10 EB CD 61	1428	CD 09 1A 21 C3 1C CD DB	1765	1808 B7 2B 1C 47 21 1A 1E 79	1738	69 10 0A 80 11 1B FE CD
1077	1B 22 45 10 7E 32 33 10	1429	10 CD 11 1B CD 09 16 21	1766	1810 07 07 CD BB 1B CD E2 10	1739	B2 1A 11 9C FF CD B2 1A
1078	E6 07 2B 09 47 32 67 10	1430	D1 1C CD DB 10 3A 30 10	1767	1818 CD 03 10 CD 03 10 CD 03	1740	11 F6 FF CD B2 1A 11 FF
1079	0E 02 CD 81 13 3A 33 10	1431	07 07 07 21 4B 1D CD BB	1768	1820 10 6B 26 00 CD 92 1A CD	1741	FF CD B2 1A FE 80 CD C3
1080	E6 3B 2B 09 0F 0F 0F 47	1432	1B CD DB 10 CD 09 16 21	1769	1828 79 FE 06 20 D3 C3 1D 12	1742	1A80 1F 1B 19 30 03 04 1B FA
1081	0E 01 CD 81 13 3A 33 10	1433	DF 1C CD DB 10 CD 2C 1B	1770	1830 AF 32 40 10 21 B0 1C C3	1743	CD C6 1A 19 7B FE 80 CD
1082	E6 40 2B 07 06 01 0E 03	1434	CD 09 16 21 ED 1C CD DB	1771	1838 1A 12 3A 40 10 B7 20 23	1744	1A00 CD 1F 1B 06 00 C9 AF 93
1083	CD 81 13 3A 33 10 B7 F2	1435	10 CD 37 1B CD 09 16 21	1772	1840 3A 42 10 B7 20 0B 3A 41	1745	5F 3E 09 9A 57 C9 21 00
1084	3D 13 32 44 10 CD AE 1B	1436	FD 1C CD DB 10 2A 35 10	1773	1848 10 FE C9 DA EA 1A 3A 3F	1746	00 01 00 00 CD 0D 10 FE
1085	EB 22 2E 10 EB CD 62 13	1437	CD 92 1A CD 09 16 21 0B	1774	1850 10 B7 C2 17 12 21 40 10	1747	0A 2B 15 FE 1B CA 1D 12
1086	20 F3 3A 67 10 47 21 00	1438	1D CD DB 10 CD 8F 1A CD	1775	1858 3A 2A 41 10 11 3B FF 19	1748	D6 30 FA 09 1B FE 0A FE
1087	0D CD F3 10 FE CD 30 F9	1439	09 16 21 19 10 CD DB 10	1776	1860 22 41 10 21 35 1D C3 1A	1749	09 1B 41 4C 65 6F 1B E4
1088	CD BB 1B 05 20 F3 22 2B	1440	2A 2A 10 CD 92 1A CD 09	1777	1868 12 3A 27 10 FE 0C F2 77	1750	7C 26 00 11 0A 00 CD 04

1AF8 1B 79 11 64 00 CD 04 1B  
 1B00 7B 11 EB 03 30 FB 19 1B  
 1B08 FB 21 BE 1C CD 0B 10 1B  
 1B10 0D 2A 25 10 CD 92 1A 3E  
 1B18 2E CD 05 10 3A 1F 10 EA  
 1B20 0F C6 30 C3 05 10 0F 0F  
 1B28 0F 0F 1B F3 3A 2D 10 CD  
 1B30 4B 1B 3A 2C 10 1B 09 3A  
 1B38 2F 10 CD 4B 1B 3A 2E 10  
 1B40 CD 57 1B 3E 29 C3 05 10  
 1B48 F5 3E 2B CD 05 10 F1 CD  
 1B50 57 1B 3E 2C C3 05 10 3C  
 1B58 1B C5 7A CD 4B 1B 7B 1B  
 1B60 0F 21 6B 10 7B 07 07 07  
 1B68 82 1B 50 21 3A 10 0E 0A  
 1B70 CB F3 10 BB 5B 08 CD F3  
 1B78 10 EA 03 37 8E 77 23 0D  
 1B80 20 EE 3A 3F 10 07 C8 AF  
 1B88 32 40 10 C9 3A 00 23 05  
 1B90 20 FA C9 3E FC 15 FA 9D  
 1B98 1B 07 07 1B F8 A6 77 C9  
 1BA0 AF 7A 1F 57 7B 1F 5F C9  
 1BA8 7C BA CD 7D BB C9 CD F3  
 1BB0 10 EA 07 57 C9 F3 10 EA  
 1BB8 07 5F C9 85 6F 00 2A C6  
 1BC0 CD E7 10 21 9B 1E 1B 06  
 1BC8 CD E7 10 21 06 1F CD E2  
 1BD0 10 21 17 1F CD E2 10 1B  
 1BD8 15 CD E7 10 21 32 1F CD  
 1BE0 E2 10 CD E1 10 21 5A 1F  
 1BE8 CD DB 10 CD E1 10 21 0A  
 1BF0 1F CD E2 10 CD 00 10 FE  
 1BF8 4E C2 00 11 C3 15 10 0D  
 1C00 04 0C 57 4D 47 20 20 20  
 1C08 53 54 43 52 20 57 41 52  
 1C10 53 0B 0A 0A 0A 4B 75 72  
 1C18 7A 2D 20 6F 64 65 72 20  
 1C20 4C 61 6E 67 66 61 73 73  
 1C28 75 6E 67 20 2B 4B 2F 4C  
 1C30 29 3F 20 04 42 61 73 69  
 1C38 73 20 69 6E 20 51 75 61  
 1C40 64 72 61 6E 74 20 04 4B  
 1C48 65 6E 6E 77 6F 72 74 3F  
 1C50 20 04 53 74 65 72 6E 7A  
 1C58 65 69 74 3A 20 04 53 7A  
 1C60 65 72 6E 6A 61 68 72 65  
 1C68 3A 20 04 51 75 61 64 72  
 1C70 61 6E 74 3A 20 51 04 07  
 1C78 53 75 70 65 72 6E 6F 7A  
 1C80 61 21 04 07 53 6F 6E 6E  
 1C88 65 6E 73 74 75 72 6D 3A  
 1C90 20 53 63 6B 69 6C 64 20  
 1C98 62 65 73 63 6B 61 65 64  
 1CA0 69 67 74 21 04 4B 6F 6D  
 1CA8 6D 61 6E 64 6F 3A 20 04  
 1CB0 53 63 6B 69 6C 64 20 75  
 1CB8 6E 74 65 6E 21 04 07 20  
 1CC0 3F 20 04 20 20 53 74 65  
 1CC8 72 6E 7A 65 69 74 3A 20  
 1CD0 04 20 20 53 74 61 74 75  
 1CD8 73 20 20 20 3A 20 04 20  
 1CE0 20 51 75 61 64 72 61 6E  
 1CE8 74 20 3A 20 04 20 20 33  
 1CF0 65 63 74 6F 72 20 20 20  
 1CF8 3A 20 04 53 04 20 20 45  
 1D00 6E 65 72 67 69 65 20 20  
 1D08 3A 20 04 20 20 4B 6C 69  
 1D10 6E 67 6F 6E 73 20 3A 20  
 1D18 04 20 20 3A 6F 72 70 65  
 1D20 64 6F 73 20 3A 20 04 20  
 1D28 20 53 63 6B 69 6C 64 20  
 1D30 20 20 3A 20 04 53 63 6B  
 1D38 69 6C 64 20 6F 62 65 6E  
 1D40 04 20 75 6E 74 65 6E 04  
 1D48 47 72 75 65 6E 20 20 04  
 1D50 52 6F 74 20 20 20 20 04  
 1D58 44 6F 63 6B 20 20 20 04  
 1D60 47 65 6C 62 20 20 20 04  
 1D68 53 63 6B 69 6C 64 20 6B  
 1D70 65 72 61 62 6C 61 73 73  
 1D78 63 6E 21 04 56 65 72 66

1D80 75 65 67 62 61 72 65 20  
 1D88 45 6E 65 72 67 69 65 3A  
 1D90 20 04 50 6B 61 73 65 72  
 1D98 65 6E 65 72 67 69 65 20  
 1DA0 3D 20 04 5A 75 77 65 6E  
 1DA8 69 67 20 45 6E 65 72 67  
 1DB0 69 65 04 07 46 65 6B 6C  
 1DB8 73 63 6B 75 73 73 04 4B  
 1DC0 6C 69 6E 67 6F 6E 20 62  
 1DC8 65 73 63 6B 61 65 64 69  
 1DD0 67 74 04 55 6E 67 75 65  
 1DD8 6C 74 69 67 04 42 69 6C  
 1DE0 64 20 66 75 65 72 20 51  
 1DE8 04 2B 2B 2B 2B 20 4A 07  
 1DF0 53 75 70 65 72 6E 6F 76  
 1DF8 61 20 69 6E 20 51 04 73  
 1EQ0 63 6B 77 65 72 20 62 65  
 1EQ8 73 63 6B 61 65 64 69 67  
 1E10 74 04 53 74 61 74 75 73  
 1E18 20 04 4D 4F 54 04 53 52  
 1E20 53 04 4C 52 53 04 50 48  
 1E28 53 04 54 52 50 04 53 43  
 1E30 4B 04 4B 75 72 73 3A 20  
 1E38 04 4B 61 6B 74 6F 72 3A  
 1E40 20 04 42 6C 6F 63 6B 6B  
 1E48 65 72 74 20 69 6E 20 53  
 1E50 04 4B 6C 69 6E 67 6F 6E  
 1E58 20 67 65 72 61 6D 6D 74  
 1E60 20 69 6E 20 33 04 47 72  
 1E68 65 6E 7A 65 20 64 65 72  
 1E70 20 47 61 6C 61 7B 69 65  
 1E78 04 33 2E 20 4D 61 6C 2E  
 1E80 20 44 61 73 20 72 65 69  
 1E88 63 6B 74 21 04 52 6F 6B  
 1E90 72 65 20 6C 65 65 72 04  
 1E98 45 6E 65 72 67 69 65 20  
 1EA0 61 6C 6C 65 04 54 6F 72  
 1EA8 70 65 64 6F 20 74 72 69  
 1EB0 66 6E 74 20 04 4B 6C 69  
 1EB8 6E 67 6F 6E 04 4E 6F 63  
 1EC0 6B 20 4B 6C 69 6E 67 6F  
 1EC8 6E 73 3A 20 04 53 74 63  
 1ED0 72 6E 04 42 61 73 69 73  
 1ED8 04 5A 75 20 66 72 75 65  
 1EE0 6B 21 04 07 46 63 6B 6C  
 1EE8 73 63 6B 6C 61 67 04 42  
 1EF0 65 6B 61 6E 6E 74 65 20  
 1EF8 47 61 6C 61 7B 69 65 20  
 1F00 76 6F 6E 20 51 04 5A 63  
 1F08 69 74 20 61 62 67 65 6C  
 1F10 61 75 66 65 6E 21 04 4D  
 1F18 69 73 73 69 6F 6E 20 77  
 1F20 61 72 20 65 69 6E 20 46  
 1F28 65 6B 6C 73 63 6B 6C 61  
 1F30 67 04 07 2A 2A 2A 20 47  
 1F38 6C 75 65 63 6B 77 75 6E  
 1F40 73 63 6B 20 2A 2A 2A 04  
 1F48 41 6C 6C 65 04 6F 65 72  
 1F50 74 6C 69 63 6B 65 6E 20  
 1F58 4B 6C 69 6E 67 6F 6E 73  
 1F60 20 7A 65 72 73 74 6F 65  
 1F68 72 74 04 44 69 65 20 46  
 1F70 6F 65 64 65 72 61 74 69  
 1F78 6F 6E 20 69 73 74 20 67  
 1F80 65 72 65 74 74 65 74 21  
 1F88 07 04 4E 6F 63 6B 20 65  
 1F90 69 6E 20 53 70 69 65 6C  
 1F98 3F 20 04 20 45 69 6E 6B  
 1FA0 65 69 74 65 6E 20 74 72  
 1FAB 61 66 65 6E 20 45 6E 74  
 1FB0 65 72 70 72 69 73 65 04  
 1FB8 20 61 75 66 67 65 6C 6F  
 1FC0 65 73 74 04 4B 6C 69 6E  
 1FC8 67 6F 6E 73 20 67 72 65  
 1FD0 69 66 65 6E 20 61 6E 2E  
 1FD8 20 53 63 6B 69 6C 64 20  
 1FE0 6B 61 65 6C 74 04 6B 6C  
 1FE8 73 63 6B 6C 61 67 00 42  
 1FF0 63 6B 61 6E 6E 74 65 20  
 1FF8

# CPU Platine

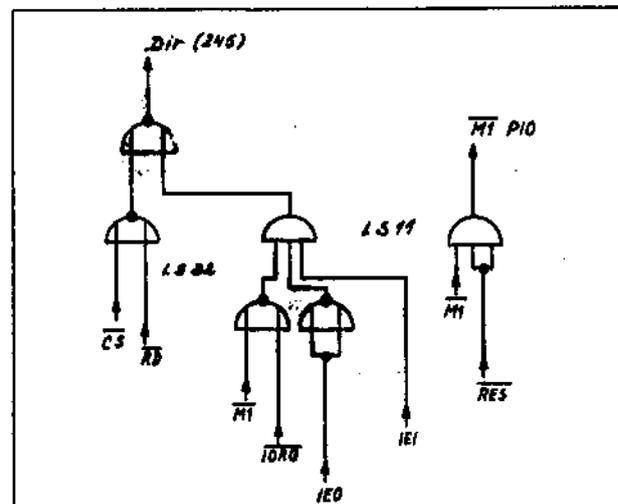
von KARL SCHULMEISTER

In Heft 1/84 wurde die neue CPU- Platine von Karl Schulmeister beschrieben. Daß die Karte noch nicht in Serie gegangen ist (obwohl sich schon einige Interessenten gemeldet haben) liegt daran, daß die Karte inzwischen völlig neu entwickelt wurde. Sie ist nun bei Karl Schulmeister seit einem Monat zur Zufriedenheit in Betrieb und ging heute an die Platinenfirma. Den Preis der Karte erfahren Sie später in einer gesonderten Mini- Ausgabe.

Die neue Konzeption hat gegenüber der früheren Karten einige Verbesserungen bezüglich der Taktaufbereitung und Reset- Signalerzeugung (dem NASCOM II angeglichen) und ist sowohl alleinstehend als EMUF als auch am ECB- Bus voll interruptfähig. Ein besonderer Dank gebührt hier Jörg Wittich, von dem die Schaltung betreffend Adreßdekodierung und Buspufferansteuerung stammt, welche die Interruptfähigkeit erst möglich macht.

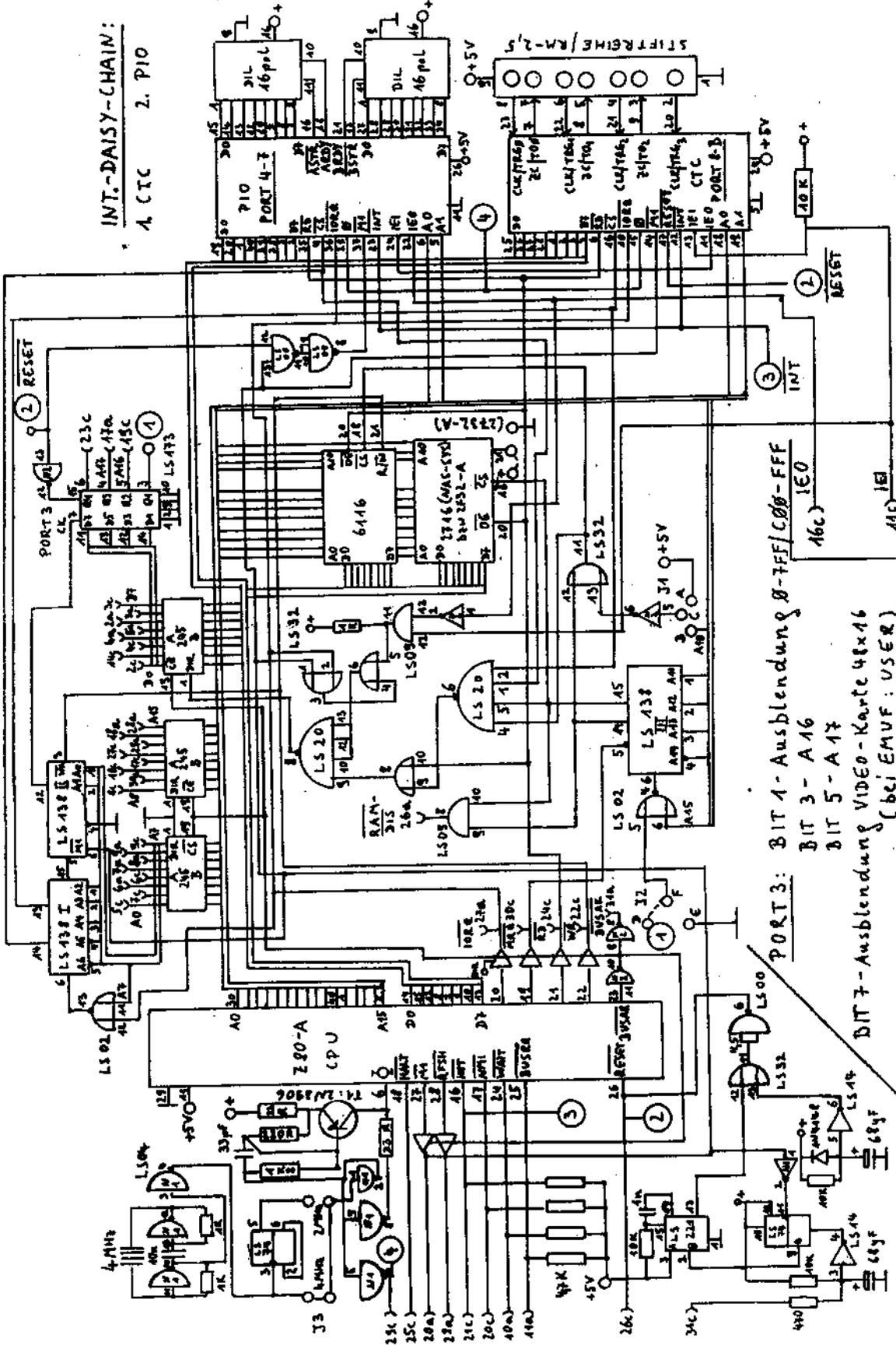
Seit kurzem läuft mit der CPU-Karte eine Softwareuhr mittels CTC und IM2, die in der Titelzeile unabhängig von laufenden Programmen die Zeit anzeigt. Wir werden dieses Programm wohl demnächst veröffentlichen können.

Anschließend finden Sie die schematische Schaltung von Jörg Wittich, wie sie allgemein zur Dekodierung Interruptfähiger Z80 Peripherie verwendet werden sollte. Leider wird sie bei den meisten Konzeptionen nicht berücksichtigt, sodaß professionelle I/O-Karten oft nicht interruptfähig sind, oder zumindest die PIOs nicht zurückgesetzt werden können.

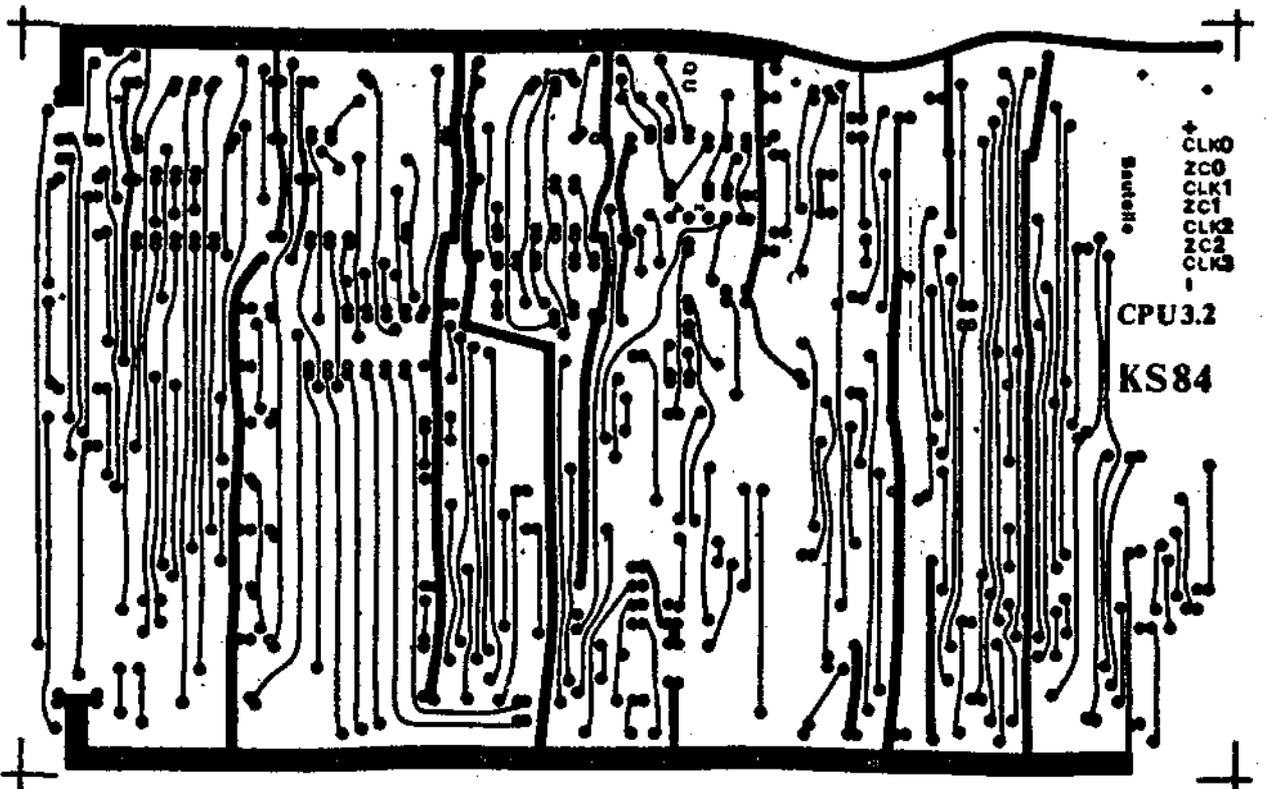
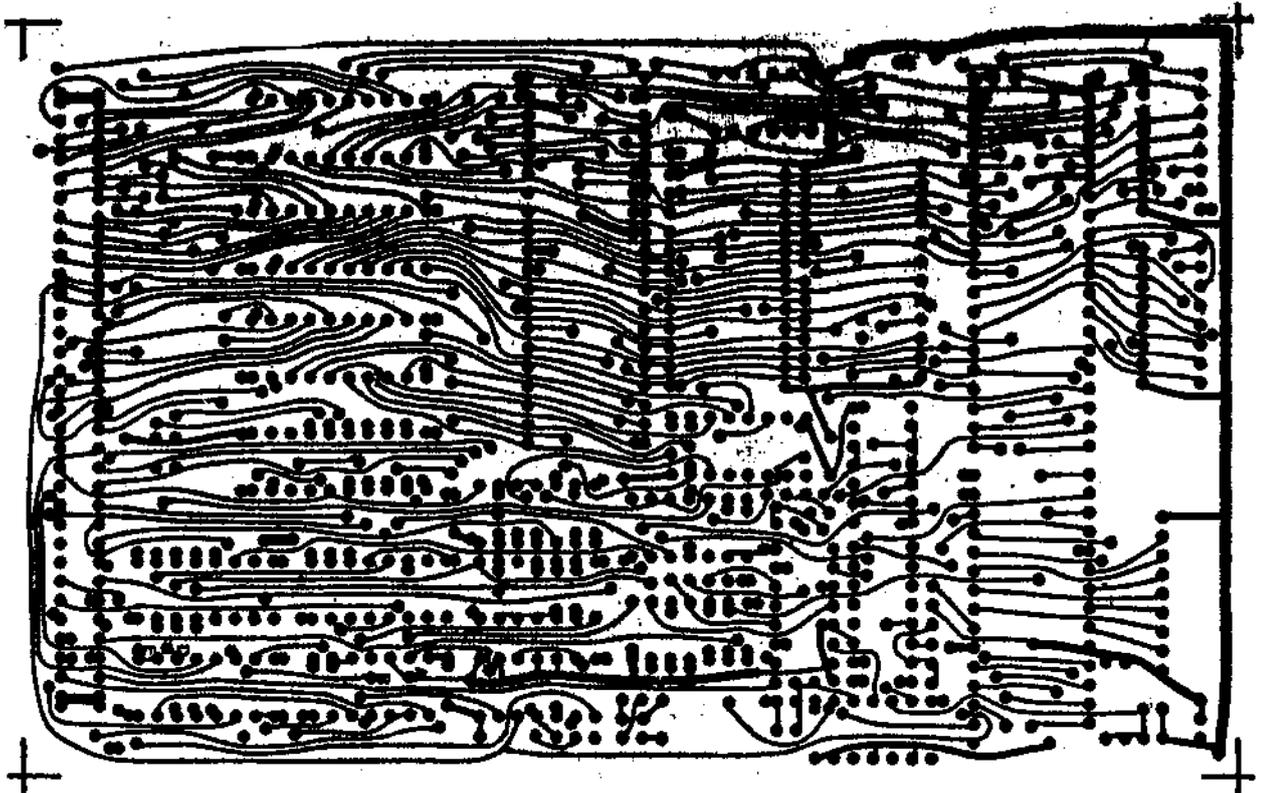




INT.-DAISY-CHAIN:  
 1 CTC 2. PIO



CPM - CPU-KARTE für NASCOM-ECB (c) 1984 by Karl SCHULMEISTER/Sörg WITTICH



# mc IN/OUT

von GÜNTER BOHM

ZEAP Z80 Assembler - Source Listing

```

0010 ; S R L O U T 1   v.1.0
0020 ;-----
0030 ;MODIFIKATION VON NASSYS
0040 ;ZUR BENUTZUNG VON "OUT1" (mc Karte)
0050 ;ALS SERIELLE EIN-/AUSGABE
0060 ;
0070 ;VER 1.0  GÜENTER BOEHM 4.1.84
0080 ;
0090 ;AUFRUF VON FLOPPY MIT SRLOUT1.COM
0100 ;-----
24FA 00F0 0120 SIOADA EQU #F0 ;DATEN
24FA 00F1 0130 SIOAST EQU #F1 ;STATUS
24FA 0087 0140 SRLIN EQU #0087 ;NASSYS SER.ABFRAGE
24FA 005B 0150 SRLX EQU #005B ;NASSYS SER.AUSGABE
0160 ;
0170 ;-----
1000 0190 ;          ORG #1000
0200 ;
1000 0EF1 0210 INITS LD C,SIOAST ;SIO INITIIBREN
1002 0608 0220 LD B,8
1004 213310 0230 LD HL,TABSIO
1007 EDB3 0240 OTIR
0250 ;
1009 212510 0260 LD HL,TABIN
100C 118700 0270 LD DE,SRLIN
100F 010700 0280 LD BC,7
1012 EDB0 0290 LDIR
0300 ;
1014 212C10 0310 LD HL,TABX
1017 115B00 0320 LD DE,SRLX
101A 010700 0330 LD BC,7
101D EDB0 0340 LDIR
0350 ;
101F 3E70 0360 LD A,#70 ;IN TABELLE WIEDER AUF SRLIN
1021 327D07 0370 LD (#7D),A ;STELLEN (WURDE IN MEINER
0380 ; MODIFIKATION ZUNAECHEST UNTERBUNDEN,
0390 ; DA KEINE SERIELLE KARTE VORHANDEN WAR.
0400 ;
1024 C9 0410 RET ;ZU EMDOS
0420 ;
0430 ;FOLGENDE AENDERUNGEN WERDEN GELADEN
0440 ;SRLIN
0450 ;IN A,(SIOAST)
0460 ;RRA ;BIT 1
0470 ;RET NC
0480 ;IN A,(SIOADA)
0490 ;RET
0500 ;
0510 ;SRLX
0520 ;PUSH AF
0530 ;OUT (SIOADA),A
0540 ;SRLA IN A,(SIOAST)
0550 ;BIT 2,A
0560 ;JR Z SRL4
0570 ;POP AF
0580 ;RET
0590 ;
1025 DBF11FD0 0600 TABIN DEFB #DB,#F1,#1F,#D0,#DB,#F0,#C9
DBF0C9 0610 ;
102C F5D3F0DB 0620 TABX DEFB #F5,#D3,#F0,#DB,#F1,#CB,#57
F1CB57 0630 ;
1033 010003C1 0640 TABSIO DEFB 1,0,3,#C1,4,#4C,5,#68
044C9568 0650 ;-----

```

```

0010 ; S R L O U T 2   v.1.1
0020 ;-----
0030 ;MODIFIKATION VON NASSYS
0040 ;ZUR BENUTZUNG VON "OUT1" (mc Karte)
0050 ;ALS SERIELLE EIN-/AUSGABE
0051 ;
0052 ;VERS. 1.1 BENUTZT BEIDE SIOKANAELE,
0053 ;WOBEI SIOA MIT 1200 BAUD SENDET (DER
0054 ;ALTE NASCOM VERKRAFTET NICHT MEHR) UND
0055 ;SIOB MIT 9600 BAUD EMPFAENGT.
0056 ;
0060 ;
0070 ;VER 1.0  GÜENTER BOEHM 4.1.84
0080 ;
0090 ;AUFRUF VON FLOPPY MIT SRLOUT1.COM
0100 ;-----
0110 ;
0120 SIOADA EQU #F0 ;DATEN
0130 SIOAST EQU #F1 ;STATUS
0131 SIOBDA EQU #F2
0132 SIOBST EQU #F3
0140 SRLIN EQU #0087 ;NASSYS SER.ABFRAGE
0150 SRLX EQU #005B ;NASSYS SER.AUSGABE
0160 ;
0170 ;-----
0180 ;
0190 ;          ORG #1000

```

```

1000 1044
1000 0E F1 06 08 21 3C 10 ED 77
1008 B3 0E F3 06 08 21 3C 10 47
1010 ED B3 21 2E 10 11 87 00 B7
1018 01 07 00 ED B0 21 35 10 33
1020 11 5B 00 01 07 00 ED B0 41
102B 3E 70 32 7D 07 C9 DB F3 33
1030 1F D0 DB F2 C9 F5 D3 F0 7D
1038 DB F1 CB 57 01 00 03 C1 FB
1040 04 4C 05 68 C9 E5 21 00 DC

```

## SINUS für GRAFIKKARTE

G.Bohm

```

1 REM SINUS1 3.6.84
10 OUT159,128
20 OUT144,7
40 OUT145,3
45 FORI=0T050STEP.005
55 X=INT(I*60) ;Y=70*SIN(I)+128
56 IFY<255THEN100
57 IFX<511THEN100
60 OUT152,0;IFX<255THENOUT152,1;X=X-256
65 OUT153,X
70 OUT155,Y;OUT144,16
80 NEXT
100 INPUTZ
110 OUT159,0

```

Dies ist noch ein kleines Beispielprogramm für die Grafikkarte. Es stellt eine Sinuskurve dar. Sie können ja mit den Werten ein bißchen herumspielen.



# CENTRONIX

von DIETER METZLER

## DRUCKERINTERFACE: ECB - CENTRONIX PARALLELE SCHNITTSTELLE

von Dieter Metzler

G. Böhm hat mich gebeten, als 80-Bus Journal-Leser auch einmal etwas für die anderen Leser zu tun, was ich hiermit beginnen möchte. Die Interessengruppe, der ich angeschlossen bin, arbeitet zwar mit SHARP-Systemen, die peripheren Geräte werden aber mit dem ECB-Bus angesteuert, sodaß Beiträge von unserer Seite durchaus möglich sind.

Unser System ist so aufgebaut, daß die einzelne Karte immer nur eine oder sehr wenige Funktionen zu verwalten hat. Wir legen also mehr Wert auf leicht überschaubare Einheiten, als auf eng geschichtelte Multifunktionskarten, welche eine Fehlersuche oftmals zum Alptraum werden lassen. Auch die vorliegende Schaltung ist einfach gehalten, sie ist im wesentlichen der internen Druckeransteuerung des SHARP MZ-700 nachempfunden und kann über Brücken oder DIL-Schalter für jede Adresse von 00H bis FFH programmiert werden, wobei jeweils 2 Adressen belegt werden. Die Karten laufen seit mehreren Monaten fehlerfrei, betrieben entweder am SHARP-Monitor oder durch CP/M angesteuert, welches bei uns zwischenzeitlich auch fehlerfrei läuft.

### Funktionsbeschreibung:

Das auszugebende Datenbyte wird in V1 abgespeichert und über V2 dem Drucker zur Verfügung gestellt. Anschließend wird dem Drucker über V5 das Strobe-Signal geschickt, welches ihm das Anliegen eines gültigen Datenbytes signalisiert. Die Übernahme der Daten quittiert der Drucker mit 'Busy' und zeigt damit seine Bereitschaft für neue Taten an. Die Abfrage des Druckers erfolgt über V3, welches die anliegenden Zustände auf den Datenbus legt.

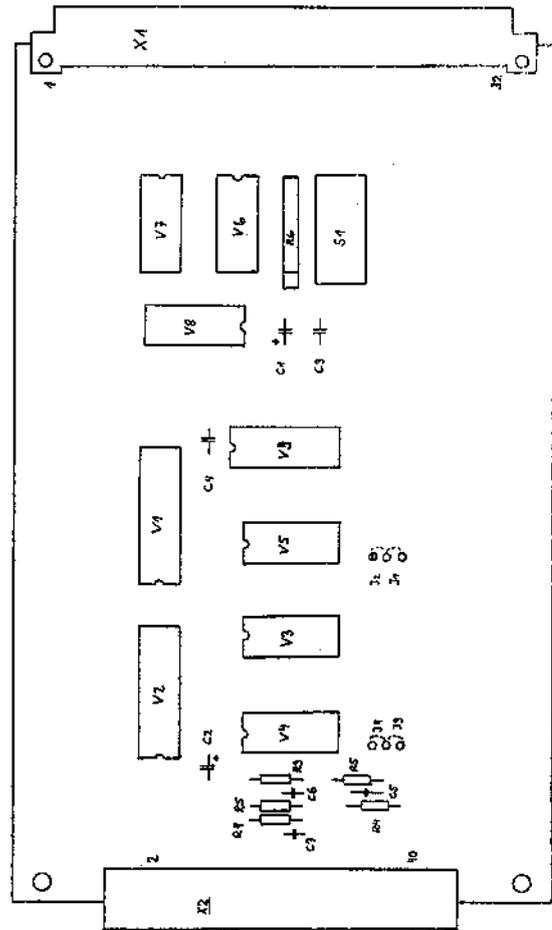
Die Daten D0...D2 können im Programm einzeln ausgewertet werden, je nachdem ob der einzelne Anwender dies wünscht oder nicht, abgestimmt auf die Möglichkeiten des vorhandenen Druckers. Die einzelnen Leitungen zum Drucker sind alle gepuffert, sodaß eine längere Zuleitung zum Drucker keine Probleme macht.

Die Anschlußbelegung der Druckerleitung ist im Schaltbild zu ersehen. Mein Triumph-Adler Drucker hat einen 40-pol. Platinenanschluß, der aber Centronix - Pin Komp. ist. Aus diesem Grund wird die Karte mit einer 40-pol. Stiftleiste bestückt. (Die nächst kleinere hat nur 34 Pole.) Das Flachbandkabel benötigt natürlich nach Centronix Norm nur 36 Pole.

### Stückliste:

C1 Tant.Elko 10uF	R1 1K
C2 dto.	R2 2,2K
C3 ker. Kond. 0,1uF	R3 1K
C4 dto.	R4 2,2K
C5 ker. Kond. 100pF	R5 1K
C6 dto.	R6 Wid.Netzwerk 7x1K
C7 dto.	
V1 74 LS 273	S1 8-fach DIL-Schalter
V2 74 LS 244	X1 VO-Leiste 64 pol.
V3 74 LS 125	X2 40 pol. Stiftleiste
V4 7400	
V5 74 LS 74	V6 74 LS 86
V7 74 LS 86	V8 74 LS 30
V9 74 LS 42	

Hinzu kommt die Platine, welche wir bei entsprechender Nachfrage in der gewohnten Qualität, durchkontaktiert mit verzinnenden Leiterbahnen anbieten werden. Bestellungen richten Sie bitte an G. Böhm.



MACRO-80 3.43 27-Jul-81 PAGE 1

```
.ZBO
;VORSCHLAG
;CP/M DRUCKERPROGRAMM
;GIBT ZEICHEN IN C ZUM DRUCKEN AUS
```

```
0000' F5          LIST:PUSH AF
0001' C5          PUSH BC
0002' 79          LD A,C
0003' 0E 00      LD C,00H
0005' 47          LD B,A
0006' CD 001D'   CALL RDA
0009' 30 3E      JR C,PNESS
000B' 78          LD A,B
000C' D3 FF      OUT (0FFH),A;  AUSGABE DATENBYTE
000E' 3E 80      LD A,80H
0010' D3 FE      OUT (0FEH),A;  STROBEAUSGABE
0012' 0E 01      LD C,01H
0014' CD 002A'   CALL RDA1;  BUSYABFRAGE
0017' AF          XDR A
001B' D3 FE      OUT (0FEH),A;  STROBE ZURÜCKSETZEN
001A' C1          POP BC
001B' F1          POP AF
001C' C9          RET

001D' DB FE      RDA: IN A,(0FEH);  ABFRAGE DRUCKER BEREIT?
001F' E6 0F      AND 0FH
0021' B9          CP C
0022' C8          RET Z
0023' E6 0E      AND 0EH
0025' B9          CP C
0026' 2B F5      JR Z,RDA
002B' 37          SCF
0029' C9          RET
```

```
002A' DB FE      RDA1: IN A,(OFEH)
002C' E6 OF      AND OFH
002E' B9         CP C
002F' CB         RET Z
0030' 1B FB      JR RDA1
```

```
0032' DB FE      RDA2: IN A,(OFEH)
0034' E6 OF      AND OFH
0036' B9         CP C
0037' CB         RET Z
0038' 37         SCF
0039' C9         RET
```

;DRUCKER AUF BEREITSCHAFT PROFEN

```
003A' C5         LISTST: PUSH BC
003B' 0E 00      LD C,00H
003D' CD 0032'   CALL RDA2
0040' 3B 04      JR C,NOT
0042' 3E FF      LD A,OFFH
0044' C1         POP BC
0045' C9         RET
```

```
0046' AF         NOT: XOR A
0047' C1         POP BC
0048' C9         RET
```

```
0049'           PMESS:
;ROUTINE ZUM AUSGEBEN EINER MELDUNG

;ABSCHLIESSEND ROCKSPRUNG INS CP/M
0049' C3 0000   JP 0000H
```

END

Macros:

Symbol:

```
0000' LIST      003A' LISTST      0046' NOT
0049' PMESS     001D' RDA         002A' RDA1
0032' RDA2
```

;VORSCHLAG FOR NASCOM:

;OBER '4' DRUCKER EINSCHALTEN UND JEDES AN DEN  
;BILDSCHIRM AUSGEBENDE ZEICHEN PARALLEL AUS-  
;DRUCKEN:

```
0000' F5         PRINT:PUSH AF
0001' C5         PUSH BC
0002' 0E 00      LD C,00H
0004' 47         LD B,A
0005' CD 001A'   CALL RDA; ABFRAGE DRUCKER BEREIT?
0008' 7B         LD A,B
0009' D3 FF      OUT (OFFH),A; ZEICHENAUSGABE
000B' 3E 80      LD A,80H
000D' D3 FE      OUT (OFEH),A; STROBEAUSGABE
000F' 0E 01      LD C,01
0011' CD 001A'   CALL RDA; BUSYABFRAGE
0014' AF         XOR A
0015' D3 FE      OUT (OFEH),A; STROBE ZURÜCKSETZEN
0017' C1         POP BC
0018' F1         POP AF
0019' C9         RET

001A' DB FE      RDA:IN A,(OFEH)
001C' E6 OF      AND OFH

;HIER KANN EINZELAUSWERTUNG VON DO ..D2 EINGEBAUT WERDEN
;Z. B.: DRUCKER NICHT BEREIT, ODER PAPIERENDE MIT ENT-
;SPRECHENDER ANZEIGE AN BILDSCHIRM
```

```
001E' B9         CP C
001F' CB         RET Z

;HIER KANN UNTERBRECHUNGSMÖGLICHKEIT EINGEBAUT WERDEN
```

```
0020' 1B FB      JR RDA
```

Verkaufe FDC- Karte (wie im Journal be-  
schrieben) fertig aufgebaut und getestet  
(habe zwei und brauche nur eine).  
Dieter Metzler 0761/581355

# Zahlenwandlung

von EBERHARD HORCH

```
1 REM * Eberhard Horch, ██████████ *
2 REM * ██████████ Hannover ██████████ *
3 REM * ----- *
4 REM
5 REM + Für Nascom umgeschrieben nach einem +
6 REM + VC 20 Vorschlag von F. Reinking in +
7 REM + der Zeitschrift Computer Persönlich +
8 REM -----
9 REM
10 CLS: CLEAR: DIM A$(20)
20 PRINT" -----
---"
30 PRINT" *** Programm zur Zahlenumwandlung
***"
40 PRINT" -----
---"
50 PRINT:PRINT" Zahlengröße ist begrenzt
auf"
60 PRINT" Hexa $FFFF == Dual= 16 Stellen
":PRINT
70 REM
80 REM +++ Basiseingaben in Ziffern +++
90 REM +++ z.B. für Hexa =16 +++
100 REM
110 INPUT"Eingabe Basis: ";B$
120 INPUT"Ausgabe Basis: ";BA$
130 INPUT" Zahl: ";A$
140 L=LEN(A$):DZ=0:HP=L-1
150 FOR N=1 TO L
160 ZS$=MID$(A$,N,1)
170 IF ZS$="A" THEN ZS$="10.0001"
180 IF ZS$="B" THEN ZS$="11"
190 IF ZS$="C" THEN ZS$="12"
200 IF ZS$="D" THEN ZS$="13"
210 IF ZS$="E" THEN ZS$="14"
220 IF ZS$="F" THEN ZS$="15"
230 ZS=VAL(ZS$)*BEIHP
240 DZ=DZ+ZS:HP=HP-1
250 NEXT N
260 PRINT:PRINT"Ergebnis:"
270 ZE=DZ:AZ=2772
280 FOR LV=16 TO 1 STEP-1
290 ZA=(ZE)/BAILV
300 IF ZA<1 AND ZE=DZ THEN 350
310 IF INT(ZA)=10 THEN U=48
320 IF INT(ZA)=11 THEN U=55
330 E=INT(ZA)+U:POKE AZ,E
340 ZE=ZE-INT(ZA)*BAILV:AZ=AZ+1
350 NEXT LV
360 ZE=ZE+5E-2
370 IF ZE<10 THEN U=48
380 IF ZE=10 THEN U=55
390 POKE AZ,ZE+U
400 FOR I=1 TO 47:PRINT"-";:NEXT:END
OK
```

# Postscheck

von KLAUS MOMBAUR

Ausfüllen einer Postschecküberweisung auf dem Drucker  
von Klaus Mombaur 8500 Nürnberg 60 ITOH 8510

So sieht der Dialog über den Bildschirm aus:

Ich füelle eine Ueberweisung aus!

Spannen Sie dazu das braune Formular mit  
Oberkante 'FUER' ans Blech ein.

An weni? \_\_\_\_\_  
? \_\_\_\_\_  
? \_\_\_\_\_  
In: ? \_\_\_\_\_

PSA: nur wenn (<) Nbg  
? \_\_\_\_\_

Kto Nr: ? \_\_\_\_\_

DM - Betrag in Ziffern: ?

Pfg - Betrag in Ziffern: ?

Datum: ? \_\_\_\_\_

Zweck: ? \_\_\_\_\_  
? \_\_\_\_\_  
? \_\_\_\_\_  
? \_\_\_\_\_

Zusatz: ? \_\_\_\_\_  
? \_\_\_\_\_

Drucken ?  
Wiederh.= w ?

```

4 REM Post-Ueberweisung
6 REM
10 REM C by K.Mombaur, Nuernberg,
20 REM
30 REM Drucker initialisieren
40 OUT6,15:OUT7,207:OUT7,245
50 OUT5,255:OUT5,247:OUT5,255
60 REM Betr.Art Kl
70 POKE3111,1
75 REM Scan Keyboard
80 DOKE3948,25311:DOKE3942,312:DOKE3844,18351
82 DOKE3946,18927:DOKE3948,-8179:POKE3858,233
84 DOKE4188,3848
100 CLEAR500:CLS
110 PRINT"Ich füelle eine Ueberweisung aus!"
120 PRINT:PRINT"Spannen Sie dazu das braune For-
mular mit
130 PRINT"Oberkante 'FUER' ans Blech ein."
140 PRINT
145 PRINT"An weni: _____"
146 PRINT"
147 PRINT"
150 SCREEN8,6:INPUTA1$
151 IFLEN(A1$)>12THEN2500
152 SCREEN8,7:INPUTA2$
153 IFLEN(A2$)>12THEN2530
154 SCREEN8,8:INPUTA3$
155 IFLEN(A3$)>12THEN2560
156 IFA1$=" "THENA1$=" "
157 IFA2$=" "THENA2$=" "
158 IFA3$=" "THENA3$=" "
160 SCREEN1,10
165 PRINT" In: _____"
166 SCREEN8,10:INPUT0$
167 IFLEN(0$)>16THEN2600
168 IF0$=" "THEN0$=" "
170 SCREEN1,12:PRINT" PSA: nur wenn (<) Nbg
171 PRINT"
172 SCREEN8,13:INPUT$
173 IFA$=" "THEN$="N"nberg"
174 IFLEN($$)>15THEN2630
175 SCREEN1,15:PRINT"Kto Nr: _____"
176 SCREEN8,14:INPUTK$
177 IFA$=" "THEN$=" "
178 IFLEN(K$)>14THEN2660
180 PRINT:INPUT" DM - Betrag in Ziffern:";B$
185 IFLEN(B$)>5THENPRINT"Zu hoch!";GOTO180
190 IFLEN(B$)<1THENB$="0"
200 PRINT:INPUT"Pfg - Betrag in Ziffern:";P$
210 IFLEN(P$)>2THENPRINT"Nur 2 Ziffern!";GOTO200
214 IFLEN(P$)<1THENP$="00"
216 IFLEN(P$)<2THENP$="0"+P$
230 SCREEN1,15:PRINT"Datum: _____"

```

```

231 SCREEN8,14:INPUT 0$
232 IFA$=" "THEN$=" "
235 IFLEN(0$)>18THEN2690
240 PRINT:PRINT" Zweck: _____"
241 PRINT"
242 PRINT"
243 PRINT"
245 SCREEN8,11:INPUTV1$:IFLEN(V1$)>20THEN245
246 SCREEN8,12:INPUTV2$:IFLEN(V2$)>20THEN246
247 SCREEN8,13:INPUTV3$:IFLEN(V3$)>20THEN247
248 SCREEN8,14:INPUTV4$:IFLEN(V4$)>20THEN248
250 IFV1$=" "THENV1$=" "
251 IFV2$=" "THENV2$=" "
252 IFV3$=" "THENV3$=" "
253 IFV4$=" "THENV4$=" "
260 PRINT:PRINT"Zusatz: _____"
261 PRINT"
264 SCREEN8,13:INPUTZ1$:IFLEN(Z1$)>12THEN264
265 SCREEN8,14:INPUTZ2$:IFLEN(Z2$)>12THEN265
267 IFZ1$=" "THENZ1$=" "
268 IFZ2$=" "THENZ2$=" "
280 PRINT:M$="0":PRINT"Drucken ?"
285 INPUT"Wiederh.= w ";M$
290 IFM$="w"THEN100
300 REM Druckart vorbereiten
305 FOR J=L TO1:READ X:GOSUB2200: NEXT
310 REM Einschub Ueberweisung
315 GOTO 5000
320 REM DM-Betrag in Buchst. wandeln/ausgeb
322 X=ASC(" "):GOSUB2200:GOSUB2200
325 L=LEN(B$):ZL=0
330 W1=VAL(MID$(B$,1,1)):W2=VAL(MID$(B$,2,1)
340 IFL<>5THENG550
345 REM Zehntausender *****
350 A=1
355 REM Zt.Stelle=1lor12?
370 IFW1=1THENIFW2=1THEN$="e1":GOSUB2000:
570
380 IFW1=1THENIFW2=2THEN$="zw1":GOSUB200
70570
385 REM 7.Stelle=0?
390 IFW2<>8THEN430
400 RESTORE
410 READS$:IFS$("<")+" "THENW10
420 GOSUB2130:GOSUB2000:GOTO570
425 REM Zt.Stelle=1?
430 IFW1<>1THEN 480
440 A=2:GOSUB2100:GOSUB2000
450 S$="zehn"
460 GOSUB2000
470 GOTO 570
475 REM Zt.Stelle, 'und', T.Stelle x10
480 A=2:GOSUB2100:GOSUB2000
490 S$="und"
500 GOSUB2000
510 A=1:RESTORE
520 READS$:IFS$("<")+" "THEN$20
530 GOSUB2130:GOSUB2000
540 GOTO570
545 REM Tausender*****
550 IFL<>4THEN$00
560 A=1:GOSUB2100:GOSUB2000
570 S$="tausend":GOSUB2000
575 REM hundert nur wenn (<0)
580 IFL=4THENIFW2=0THENW=3:GOSUB2300:GOTT
590 IFL<>5THEN$00
592 W3=VAL(MID$(B$,3,1))
593 IFW3=8THENW=4:GOSUB2300:GOTO820
595 REM Hunderter *****
600 REM
610 IFL<>4THEN$00
615 A=3:IFL=4THENW=2
620 GOSUB2100:GOSUB2000
630 S$="hundert":GOSUB2000
650 REM IFZL(38THENGOSUB2300:X=13:GOSUB2
655 REM H.Stelle wenn (4 Stellen *****
700 IFL<>5THEN750
705 REM X=42:GOSUB2200:GOSUB2200
710 A=1:GOSUB2100:GOSUB2000
720 S$="hunder":GOSUB2000
730 A=2:GOTO823
750 IFL=5THENW=4:GOTO822
760 IFL=4THENW=3:GOTO822
795 REM Zehner *****
800 IFL<>2THEN990
810 A=1
812 GOTO 822

```

Dieses Programm druckt Ihnen auch ein Nachweis-  
blatt der ausgefüllten Überweisungen.  
Spannen Sie dazu einfach, nachdem der Drucker  
das Formular ausgefüllt hat, ein DIN A4 - Blatt  
ein und beantworten die Computerfrage mit "n".

Wer nun dieses Programm oder das Ausfüllen de. DB 3: (NP PRIME (neg) )  
Postschecks aus Heft 6/82 (mit dem man nat. auch DB 4: (NP BUSY )  
Bankschecks aber keine Euroschecks ausstellen DB 5: SELECT )  
kann) so wie ich über die Centronics - Parallel- DB 6: PAPER EMPTY )  
Schnittstelle laufen lassen will, braucht nur DB 7: BUSY )  
noch folgende Belegung der NASCOM 2 - PIO:  
Port A: DB 0 bis DB 7 für die Daten  
Port B: für die Steuersignale  
DB 0: frei  
DB 1: STROBE (neg) )  
DB 2: FAULT (neg) )

... und so eine ausgefüllte Anweisung:

1234 56 - 123	*55555* 23	*55555* 23 *****	*55555* 23
		**fünfundfünzigtausendfünfhundertfünfundfünzig*****	
	Herrn Roland		Herrn
	Thomasius		Roland
	5650 Solingen 12	1234 56 - 123	Thomasius
		Köln	5650 Solingen 12
		24.4.1985	
Rechnungsnr.: 345 678 444 - 66 v. 23.12.00 Ku.Nr. 9876 54321			

```

815 REM Z+E.Stelle=lor12?
820 REM X=13:GOSUB2200
822 REM X=42:GOSUB2200:GOSUB2200
823 W1=VAL(MID$(B$,A,1)):W2=VAL(MID$(B$,A+1,1))
824 IFW1=0 AND W2=0 THEN1100
824 IFW1=0 THEN970
828 IFW1=1THENIFW2=1THENS$="elf":GOSUB2000:GOTO
1100
930 IFW1=1THENIFW2=2THENS$="zwillf":GOSUB2000:GO
T01100
835 REM E.Stelle=0?
940 IFW2()THEN800
950 RESTORE
840 READS$:IFS$()?"*":THEN850
870 GOSUB2130:GOSUB2000:GOTO1100
875 REM Z.Stelle=1?
880 IFW1()THEN 930
890 A=A+1:GOSUB2100:GOSUB2000
900 S$="zehn"
910 GOSUB2000
920 GOTO1100
925 REM E.Stelle, "und", Z.Stelle x10
930 A=A+1:GOSUB2100:GOSUB2000
940 S$="und"
950 GOSUB2000
960 RESTORE
970 READS$:IFS$()?"*":THEN970
980 A=A+1:GOSUB2130:GOSUB2000:GOTO1100
985 REM Einer *****
990 A=A+1:IFL()THEN1000
995 X=42:GOSUB2200:GOSUB2200
1000 IFW1=0 AND L=1 THENS$="null":GOTO1020
1010 GOSUB2100
1020 IFS$="ein"THEN S$="eine"
1030 GOSUB2000
1100 GOSUB2300
1110 REM Rest 3.Zeile ESC N+HT
1115 X=27:GOSUB2200:X=78:GOSUB2200:X=9:GOSUB220
0
1120 X=ASC("*"):GOSUB2200
1130 S$=S$:GOSUB2000
1140 X=ASC("*"):GOSUB2200
1150 X=9:GOSUB2200:REM HT
1160 S$=P$:GOSUB2000
1170 X=13:GOSUB2200
1200 GOTO 5400
2000 REM U-Ausgabe *****
2010 FORA2=1TO LEN(S$)
2020 X=ASC(MID$(S$,A2,1)):ZL=ZL+I
2030 GOSUB 2200
2040 NEXT A2
2050 RETURN
2100 REM U-DATA SETZEN
2110 RESTORE
2120 READS$:IFS$()?"*":THEN2120
2130 FORA1=1TO VAL(MID$(B$,A,1)):READS$:NEXT
2140 RETURN
2200 REM U-Drucken
2210 OUT4,X:OUT5,253:OUT5,255:WAITS,16,16
2220 RETURN
2300 REM U-Auffuellen
2310 FOR A2=1TO59-ZL
2320 K=ASC("*"):GOSUB2200
2330 NEXT ZL=0
2340 RETURN
2500 SCREEN6,6
2510 GOSUB2000
2520 SCREEN6,6:PRINT"-----" *;GOTO
150
2530 SCREEN6,7
2540 GOSUB2000
2550 SCREEN6,7:PRINT"-----" *;GOTO

```

```

152
2560 SCREEN8,8
2570 GOSUB2000
2580 SCREEN8,8:PRINT"-----" *;GOTO
154
2600 SCREEN8,10
2610 GOSUB2000
2620 GOTO160
2630 SCREEN8,13
2640 GOSUB2000
2650 GOTO170
2660 SCREEN8,14
2670 GOSUB2000
2680 GOTO175
2690 SCREEN8,14
2700 GOSUB2000
2710 GOTO230
2800 PRINT" Zu lang, bitte neu!"
2810 FORX=1TO2000:NEXT
2820 RETURN
2990 REM ESC L002 ESC N ESC T 47 00
3000 DATA27,76,48,48,50,27,70,27,84,52,55,*
3010 DATAein,zwei,drei,vier,f,sechs,sieben
3020 DATAacht,neun,t,zehn,zwanzig,drei'ig
3030 DATAvierzig,finfzig,sechzig,siebzig
3040 DATAachtzig,neunzig
5000 REM Ausgabe Kto Nr
5010 S$=K$:GOSUB2000:X=13:GOSUB2200
5020 REM TAB setzen 2.Zeile: ESC(13,22,33,39.
5040 RESTORE 5050
5050 DATA27,40,40,49,51,44,48,50,50
5052 DATA44,40,51,51,44,40,51,57,46
5060 FORY=1TO10
5070 READ X: GOSUB2200: NEXT
5100 REM Ausgabe 2.Zeile
5110 X=ASC("*"):GOSUB2200
5120 S$=S$:GOSUB2000
5130 X=ASC("*"):GOSUB2200
5140 X=9:GOSUB2200:REM HT
5150 S$=P$:GOSUB2000
5170 X=9:GOSUB2200:REM HT
5190 X=ASC("*"):GOSUB2200
5190 S$=S$:GOSUB2000
5200 X=ASC("*"):GOSUB2200
5210 X=9:GOSUB2200:REM HT
5220 S$=P$:GOSUB2000
5240 X=9:GOSUB2200:REM HT
5250 FORY=1TO19:X=ASC("*"):GOSUB2200:NEXT
5260 X=13:GOSUB2200
5300 REM TAB setzen 3.Zeile
5305 REM Clear+ESC(22,64,75)+ESC 0+HT
5310 RESTORE 5320
5320 DATA27,30,27,40,40,50,50,44,48,54,52,44,40
5330 DATA55,53,40,27,81,9
5340 FORY=1TO19
5350 READ X: GOSUB2200: NEXT
5360 REM Ausgabe 3.Zeile
5370 GOTO 320
5400 REM TAB 4.Zeile: Clear,ESC(22,65)+HT
5410 RESTORE 5420
5420 DATA27,30,27,40,40,50,50,44,48,54,53,46,9
5440 FORY=1TO13
5450 READ X: GOSUB2200: NEXT
5460 REM Ausgabe 4.Zeile
5470 S$=A1$:"*":A2$:GOSUB2000
5480 X=9:GOSUB2200
5490 S$=A1$:GOSUB2000
5500 X=13:GOSUB2200
5600 REM TAB 5. Zeile bleibt
5650 REM Ausgabe 5.Zeile
5660 X=9:GOSUB2200
6740 RESTORE 6770
6770 DATA27,30,27,40,48,52,48,44,48,54,50,44
6780 DATA40,56,52,44,49,40,54,46,9,99
6800 READ X:IFX=99THEN6820
6810 GOSUB2200:GOTO6000
6820 REM Ausgabe Nachweis 2.Zeile
6840 S$=V1$:GOSUB2000:X=9:GOSUB2200
6850 S$=V2$:GOSUB2000:X=9:GOSUB2200
6860 S$=V3$:GOSUB2000:X=9:GOSUB2200
6870 S$=V4$:GOSUB2000:X=9:GOSUB2200
6880 X=13:GOSUB2200
6900 CLS:PRINT:PRINT:PRINT:PRINT" Noch ein
Formular = f"
6910 PRINT" Programm beenden = e"
6920 A=USR(0):IF CHR$(A)="e"THEN CLS:END
6930 IF A=0 THEN 6920
6940 GOTO 100

```

```

5470 S$=A3$:GOSUB2000
5480 X=9:GOSUB2200
5490 S$=A2$:GOSUB2000
5495 X=13:GOSUB2200
5700 REM TAB 6.Zeile:Clear+ESC(22,65.
5710 RESTORE 5720
5720 DATA27,30,27,40,48,50,50,44,48,52,50,44
5730 DATA40,54,53,46
5740 FORY=1TO16
5750 READ X: GOSUB2200: NEXT
5760 REM Ausgabe 6.Zeile
5770 S$=Z1$:GOSUB2000
5780 X=9:GOSUB2200
5790 S$=O$:GOSUB2000
5800 X=9:GOSUB2200
5810 S$=K$:GOSUB2000
5820 X=9:GOSUB2200
5830 S$=A3$:GOSUB2000
5840 X=13:GOSUB2200
5900 REM TAB 7.Zeile:Clear+ESC(42,65.
5910 RESTORE 5920
5920 DATA27,30,27,40,48,52,50,44,48,54,53,46
5930 FORY=1TO12
5940 READ X: GOSUB2200: NEXT
5950 REM Ausgabe 7.Zeile
5960 S$=Z2$:GOSUB2000
5970 X=9:GOSUB2200
5990 S$=A$:GOSUB2000
6000 X=9:GOSUB2200
6010 S$=O$:GOSUB2000
6020 X=13:GOSUB2200
6100 REM TAB 8.Zeile:Clear+ESC(32+HT+ESC T 24
6110 RESTORE 6120
6120 DATA27,30,27,40,40,51,50,46,9,27,84,50,52
6130 FORY=1TO13
6140 READ X: GOSUB2200: NEXT
6150 REM Ausgabe 8.Zeile
6160 S$=O$:GOSUB2000
6170 X=13:GOSUB2200
6200 REM 9.-12.Zeile:ESC 0
6210 X=27:GOSUB2200:X=81:GOSUB2200
6250 REM Ausgabe 8.Zeile
6260 S$=V1$:GOSUB2000
6270 X=13:GOSUB2200
6280 S$=V2$:GOSUB2000
6290 X=13:GOSUB2200
6300 S$=V3$:GOSUB2000
6310 X=13:GOSUB2200
6320 S$=V4$:GOSUB2000
6330 X=13:GOSUB2200
6400 REM Endebehandlung
6405 CLS
6410 SCREEN 5,10
6420 PRINT"Die Unterschrift nicht vergessen !!!"
6430 PRINT:PRINT" Selben Druck wdh = w"
6435 PRINT" Nachweis drucken = n"
6440 FORA=1TO100
6450 A1=USR(0):IFAL()0 THEN 6510
6460 NEXT
6470 SCREEN5,10:PRINT"
6480 FORA=1TO100
6490 A1=USR(0):IFAL()0 THEN 6510
6500 NEXT: GOTO 6410
6510 W$=CHR$(A1)
6520 IFW$="w"THEN RESTORE:W$="n":GOTO300
6530 IFW$="n"THEN6600
6540 GOTO 6480
6600 REM Nchw.:Clear,ESC(22,40,55,70,90,110,125

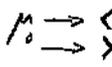
```



# Sterne

von MICHAEL BACH

PROGRAM STERNE; (29.08.81\*)  
(\*Michael Bach, Gundelfingen,  
Tel. .  
Ein unfertiges Programm, das die Illusion eines  
Fluges durch den Weltraum vermitteln soll.

Gesteuert  
wird mit den Cursor-Tasten rechts und links.  
Man  
an einen Stern stoest, blitzt es. 

\*)  
CONST STM=20; ZS=90; VZ=15; ZN1=400; ZN2=250;  
ZN3=180; ZN4=100;  
XFMI=11; XFMA=86; YFMI=8; YFMA=36;

VAR XS,YS,X,Y,Z: ARRAY(1..STM) OF INTEGER;  
STACK: INTEGER;  
AUS: BOOLEAN;  
ESC, KEY: INTEGER;

PROCEDURE LINE(X0,Y0,X1,Y1,Z1: INTEGER);  
(\*Vektor von (X0,Y0) nach (X1,Y1), Z wie in  
PLOT(X,Y,Z)\*)  
VAR I,DX,DY,D,AX,AY,BX,BY: INTEGER;  
BEGIN  
DX:=X1-X0; DY:=Y1-Y0; BX:=0; AY:=0;  
IF DX<0 THEN BEGIN AX:=-1; DX:=-DX END ELSE  
AX:=1;  
IF DY<0 THEN BEGIN BY:=-1; DY:=-DY END ELSE  
BY:=1;  
IF DX<0 AND DY<0 THEN BEGIN  
I:=DX; DX:=DY; DY:=I; BX:=AX; AX:=0; AY:=BY;  
BY:=0;  
END;  
I:=0; D:=DX SHIFT -1 (\*=DIV 2\*);  
REPEAT  
PLOT(X0,Y0,Z1);  
X0:=X0+AX; Y0:=Y0+AY; D:=D+DY; I:=I+1;  
IF D<0 THEN BEGIN  
D:=D-DX; X0:=X0+BX; Y0:=Y0+BY;  
END;  
UNTIL I<0  
END; (\*LINE\*)

PROCEDURE KREIS(X0,Y0,R,Z1: INTEGER);  
VAR I,X,Y,S: INTEGER;  
NEGX,NEGY,SWAP: BOOLEAN;  
PROCEDURE KPLOT(X,Y: INTEGER);  
BEGIN  
IF SWAP THEN BEGIN I:=X; X:=Y; Y:=I END;  
IF NEGY THEN Y:=-Y;  
IF NEGX THEN X:=-X;  
X:=X+X0; Y:=(Y\*6) DIV 8+Y0;  
IF ((X<0 AND X<XFMA) AND (Y<0 AND Y<YFMA)) THEN  
PLOT(X,Y,Z1);  
END; (\*KPLOT\*)  
BEGIN  
FOR NEGX:=TRUE TO FALSE DO  
FOR NEGY:=TRUE TO FALSE DO  
FOR SWAP:=TRUE TO FALSE DO BEGIN  
X:=R; Y:=0; S:=-R;  
REPEAT  
KPLOT(X,Y); S:=S+Y+1; Y:=Y+1;  
IF S<0 THEN BEGIN S:=S-X+2; X:=X-1 END;  
UNTIL Y<0;  
END;  
END;

PROCEDURE FILL(X0,Y0: INTEGER);  
(\*rekursiver Fuell-Algorithmus\*)  
BEGIN  
STACK:=STACK+1;

IF (STACK<300) AND (NOT POINT(X0,Y0)) THEN BEGIN  
PLOT(X0,Y0,1);  
IF (Y0<YFMI) AND (NOT POINT(X0,Y0-1)) THEN  
FILL(X0,Y0-1);  
IF (X0<XFMA) AND (NOT POINT(X0+1,Y0)) THEN  
FILL(X0+1,Y0);  
IF (Y0<YFMA) AND (NOT POINT(X0,Y0+1)) THEN  
FILL(X0,Y0+1);  
IF (X0<XFMI) AND (NOT POINT(X0-1,Y0)) THEN  
FILL(X0-1,Y0);  
END;  
END;

PROCEDURE NEUP(I: INTEGER);  
(\*neuer Punkt\*)  
CONST BEREICH=100;  
BEGIN  
X(I):=RANDOM(BEREICH\*4)-BEREICH\*2;  
Y(I):=RANDOM(BEREICH\*2)-BEREICH;  
Z(I):=ZN1+ZN1+RANDOM(1000);  
XS(I):=0; YS(I):=0;  
END; (\*NEUP\*)

PROCEDURE INI;  
VAR I: INTEGER;  
BEGIN  
FOR I:=1 TO STM DO NEUP(I);  
FOR I:=XFMI TO XFMA DO BEGIN  
PLOT(I,YFMI,1); PLOT(I,YFMA,1);  
END;  
FOR I:=YFMI TO YFMA DO BEGIN  
PLOT(XFMI,I,1); PLOT(XFMA,I,1);  
END;  
END; (\*INI\*)

PROCEDURE FLASH(KODE: INTEGER);  
(\*Besetzen des Bildschirmfensters mit KODE\*)  
CODE \$DD,\$2A,\$92,\$C,\$DD,\$7E,\$FE,\$21,\$90,\$8,  
\$11,\$1B,\$0,\$E,\$9,\$6,\$25,\$77,\$23,\$10,\$FC,  
\$19,\$D,\$20,\$F6;

PROCEDURE DELAY;  
VAR I: INTEGER;  
BEGIN FOR I:=1 TO STM\*10 DO; END;

PROCEDURE PROJEKTION;  
(\*3dimensional-->2dimensional\*)  
VAR I,X1,Y1,Z1: INTEGER;  
BEGIN  
FOR I:=1 TO STM DO BEGIN  
AUS:=TRUE; Z1:=Z(I)-VZ;  
X1:=((ZS\*X(I)) DIV Z1)+48;  
IF X1<XFMI THEN  
IF X1<XFMA THEN BEGIN  
Y1:=((ZS\*Y(I)) DIV Z1)+24;  
IF Y1<YFMI THEN AUS:=Y1<YFMA;  
END;  
XS(I):=X1; YS(I):=Y1;  
IF AUS THEN NEUP(I) ELSE Z(I):=Z1;  
END; (\*FOR\*)  
END; (\*PROJ.\*)

PROCEDURE PLOTTEN;  
VAR I,J,K,X1,Y1,Z1: INTEGER;  
BEGIN  
FOR I:=1 TO STM DO BEGIN  
X1:=XS(I); Y1:=YS(I); Z1:=Z(I);  
PLOT(X1,Y1,1);  
IF Z1<ZN1 THEN BEGIN  
PLOT(X1+1,Y1,1); PLOT(X1,Y1+1,1);  
PLOT(X1+1,Y1+1,1);  
IF Z1<ZN2 THEN BEGIN  
KREIS(X1,Y1,2,1);  
IF Z1<ZN3 THEN BEGIN  
KREIS(X1,Y1,3,1); KREIS(X1,Y1,4,1);  
IF Z1<ZN4 THEN BEGIN  
FOR J:=5 TO 8 DO KREIS(X1,Y1,J,1);  
IF Z1<ZS+1 THEN BEGIN  
FOR J:=1 TO 15 DO BEGIN  
FLASH(255); DELAY; FLASH(32); DELAY;  
END;

```

    NEUP(I);
  END;
END;
END;
END;
END;
END;
END; (*PLOTTE*)

PROCEDURE SORTIEREN;
VAR I,ZMIN,IMIN: INTEGER;
BEGIN
  ZMIN:=MAXINT;
  FOR I:=1 TO STM DO
    IF Z(.I.)<ZMIN THEN IMIN:=I;
    I:=X(.IMIN.); X(.IMIN.):=X(.STM.); X(.STM.):=I;
    I:=Y(.IMIN.); Y(.IMIN.):=Y(.STM.); Y(.STM.):=I;
    I:=Z(.IMIN.); Z(.IMIN.):=Z(.STM.); Z(.STM.):=I;
  END; (*SORTIEREN*)

PROCEDURE MOVE(KEY: INTEGER);
CONST DXY=100; CUL=$11; CUR=$12; CUU=$13; CUD=$14;
VAR I,XV,YV: INTEGER;
BEGIN
  XV:=0; YV:=0;
  CASE KEY OF
    CUR: XV:=-1;
    CUL: XV:=1;
    CUU: YV:=1;
    CUD: YV:=-1;
  END; (*CASE*)
  IF XV#0 THEN
    FOR I:=1 TO STM DO
      X(.I.):=X(.I.)+XV*Z(.I.) DIV DXY;
  IF YV#0 THEN
    FOR I:=1 TO STM DO
      Y(.I.):=Y(.I.)+YV*Z(.I.) DIV DXY;
  END; (*MOVE*)

BEGIN (*HP*)
  WRITE(CHR(12)); ESC:=27;
  INI;
  REPEAT;
  PROJEKTION;
  SORTIEREN;
  FLASH($20);
  PLOTTE;
  KEY:=KEYBOARD;
  IF KEY#0 THEN MOVE(KEY) ELSE DELAY;
  UNTIL KEY=ESC;
END.

```

# Kannibalen

von GERHARD KLEMENT

```

1 REM -- KANNIBALEN FILE T --
2 REM BYTE Sept 80. pp106
3 REM Mod.f.NASCOM 2
4 REM KANNIBALEN "BELL" ZEICHEN
5 REM MISSIONAR MAENNCHEN
6 REM PROBLEM: ES DUERFEN WEDER AM STRAND
7 REM NOCH IM BOOT MEHR KANNIBALEN ALS
8 REM MISSIONARE SEIN.
10 REM LOESUNG DES PROBLEMS MIT HILFE VON
11 REM ** BACKTRACKING **
12 REM ES WIRD DER LOESUNGSWEG, FALLS VORHANDEN
13 REM AM BILDSCHIRM ANGEZEIGT
14 REM VERZOEGERUNG IN ZEILEN 300 UND 570
15 REM EINSTELLEN
100 CLEAR5000:CLS
110 Y$=""
120 X$=""
130 REM - Boot links u Boot rechts --
140 BL$="L="X$:BR$=X$+"L="
150 DIMB(30),C(30),CT(5),D(30),M(30),MT(5)
160 DIMMV$(30)
170 SCREEN1,8:PRINT"Anzahl der Personen ";

```

```

180 PRINT"(4-16)";:INPUTN
190 CLS:IFN<4ORN>16THEN170
200 PRINTTAB(15)"Missionare und Kannibalen"
210 REM -- Zeichne Fluss --
220 FORK=4TO40:SET(44,K):SET(67,K):NEXT
230 CI=INT(N/2):MI=N-CI:BP=1:I=0
240 CL=CI:CR=0:ML=MI:MR=0
250 CT(1)=2:CT(2)=1:CT(3)=0:CT(4)=0:CT(5)=1
260 MT(1)=0:MT(2)=0:MT(3)=2:MT(4)=1:MT(5)=1
270 GOSUB700:GOSUB620
280 C(I)=CL:M(I)=ML:B(I)=BP
290 IFML#0ANDCL#0THEN590
300 FORK=1TO5:NEXT:REM VERZOEGERUNG
310 I=I+1:D(I)=0
320 D(I)=D(I)+1:IFD(I)>5THEN530
330 IFBP=-1THEN360
340 IFCL<CT(D(I))ORML<MT(D(I))THEN320
350 GOTO370
360 IFCR<CT(D(I))ORMR<MT(D(I))THEN320
370 CL=CL-BP*CT(D(I)):CR=CI-CL
380 ML=ML-BP*MT(D(I)):MR=MI-ML:BP=-BP
390 IFML#0ANDCL#0MLTHEN510
400 IFMR#0ANDCR#0MRTHEN510
410 K=0
420 IFCL=C(K)ANDML=M(K)ANDBP=B(K)THEN510
430 K=K+1:IFK<ITHEN420
440 A$="":IFCT(D(I))#0THEN460
450 FORIZ=1TOCT(D(I)):A$=A$+"":NEXT
460 B$="":IFMT(D(I))#0THEN480
470 FORIZ=1TOMT(D(I)):B$=B$+"INT":NEXT
480 IFBP=-1THENMV$(I)=A$+B$+"="2":GOTO500
490 MV$(I)="L="A$+B$
500 GOTO270
510 BP=-BP:CL=CL+BP*CT(D(I)):CR=CI-CL
520 ML=ML+BP*MT(D(I)):MR=MI-ML:GOTO320
530 SCREEN9,14:PRINT"Zurueck und nochmals"
540 I=I-1:IFI<1THENSREEN1,14:PRINTY$:GOTO600
550 CL=C(I-1):CR=CI-CL:ML=M(I-1):MR=MI-ML
560 BP=B(I-1):GOSUB700:GOSUB620
570 FORK=1TO5:NEXT:REM VERZOEGERUNG
580 SCREEN1,14:PRINTY$:GOTO320
590 SCREEN1,14:PRINT"Erfolg":SCREEN47,13:INPUTU
595 GOTO100
600 SCREEN1,2:PRINTX$:SCREEN1,14
610 PRINT"Misserfolg":SCREEN47,13:INPUTU:GOTO100
0
620 IFI#0THENRETURN
630 FORK=1TO14:SCREEN1,K:PRINTX$
640 NEXT
650 S=I-13:IFS<1THENS=1
660 FORK=STOI:J=K-S+1
670 SCREEN1,J:PRINT" ",MV$(K):NEXT:RETURN
680 REM - Graphik -
690 CR$="":CL$="":MR$="":ML$=""
700 CR$="":CL$="":MR$="":ML$=""
710 REM - Kannibalen rechts -
720 Z$="":FORIZ=1TO8-CR:Z$=Z$+"":NEXT
730 IFCR#0THEN750
740 FORIZ=1TOCR:CR$=CR$+"":NEXT
750 CR$=CR$+Z$
760 REM - Kannibalen links -
770 Z$="":FORIZ=1TO8-CL:Z$=Z$+"":NEXT
780 CL$=Z$:IFCL#0THEN810
790 FORIZ=1TOCL:CL$=CL$+"":NEXT
800 REM - Missionare rechts -
810 Z$="":FORIZ=1TO8-MR:Z$=Z$+"":NEXT
820 IFMR#0THEN840
830 FORIZ=1TOMR:MR$=MR$+"INT":NEXT
840 MR$=MR$+Z$
850 REM - Missionare links -
860 Z$="":FORIZ=1TO8-ML:Z$=Z$+"":NEXT
870 ML$=Z$:IFML#0THEN890
880 FORIZ=1TOML:ML$=ML$+"INT":NEXT
890 IFBP=-1THENB$=BL$:GOTO910
900 B$=BR$
910 SCREEN14,8:PRINTCL$
920 SCREEN36,8:PRINTCR$
930 SCREEN24,9:PRINTB$
940 SCREEN14,10:PRINTML$
950 SCREEN36,10:PRINTMR$
960 RETURN

```

# Adressverwaltung

von GÜNTER BÖHM

Einige Leser haben das Adreßverwaltungsprogramm auf ihrem Rechner implementiert. Besonders interessant war dabei die Routine, die Adressen zweispaltig ausdrückt. Aber gerade darin befanden sich noch ein paar Käfer, wie sich nachträglich herausstellte. Ich möchte den umgeschriebenen Programmteil hier nochmals abdrucken; das vollständige Listing der Adreßverwaltung finden Sie auf Disk2.

Der Fehler bestand in der Praxis darin, daß bei Adressen, die mehr als drei Zeilen hatten (z.B. Ausland) nicht genügend Zeilenabstände machten. Weiterhin wurden bei manchen Records Informationen bei "Aufkleber" mit ausgedruckt, die nicht gedruckt werden sollten.

Bisher arbeitet das Programm einwandfrei, die Fehler sollten also aller Wahrscheinlichkeit nach ausgemerzt sein.

```

4755 {Unterpr. Druckausgabe (in AUSGAB)
4760 LLEN EQU 32 {Zeilenlänge (pro Spalte)
4765 BLEN EQU 320 {Pufferlänge (18 Zeilen)
4770 DRUCK LD A,(PFLAG)
4775 CP 0
4780 JR NZ DRUCK2
4785 JR KLEB1
4790 DRUCK2 CALL EIP2
4795 XOR A
4800 LD (PFLAG),A
4805 LD HL,(DATEND)
4810 LD BC,BLEN
4815 DRUCK3 LD A,(HL)
4820 CALL PRINT
4825 INC HL
4830 DEC BC
4835 LD A,B
4840 OR C
4845 JR NZ DRUCK3
4850 RET
4855 ;-----
4860 {Unterprog. Ausdruck von Aufklebern
4865 {mit TANDY (in AUSGAB)
4870 KLEBER LD A,(PFLAG)
4875 CP 0
4880 JR NZ KLEB2
4885 KLEB1 CALL INITP
4890 CALL EIP1
4895 LD A,0FF
4900 LD (PFLAG),A
4905 RET
4910 KLEB2 CALL EIP2
4915 XOR A
4920 LD (PFLAG),A
4925 LD HL,(DATEND)
4930 LD BC,BLEN
4935 XOR A
4940 LD (FLAG2),A
4945 LD (FLAG2),A {0=SPALTE DRUCKEN
4950 KLEB3 LD A,(HL)
4955 CP *
4960 JR NZ FLAG1
4965 LD A,0FF {SPALTE NICHT DRUCKEN

```

```

4970 LD (FLAG1),A
4975 FLAG1 PUSH AF
4980 LD A,(FLAG1)
4985 OR A
4990 JR Z KLEB4
4995 POP AF
5000 JR KLEB5
5005 KLEB4 POP AF {DRUCKEN
5010 CALL PRINT
5015 KLEB5 INC HL {NEXT LETTER
5020 DEC BC
5025 LD A,(HL)
5030 CP #18 {SPALTE FERTIG?
5035 JR NZ KLEB3
5040 KLEB6 LD A,(HL)
5045 CP #0
5050 JR Z KLEB8+1 {NL INNER DRUCKEN
5055 CP *
5060 JR NZ KLEB7
5065 LD A,0FF
5070 LD (FLAG2),A
5075 KLEB7 PUSH AF
5080 LD A,(FLAG2)
5085 OR A
5090 JR Z KLEB8
5095 POP AF
5100 JR KLEB9
5105 KLEB8 POP AF {DRUCKEN
5110 CALL PRINT
5115 KLEB9 INC HL
5120 DEC BC
5125 PUSH AF
5130 LD A,B
5135 OR C
5140 JR Z KLEB10
5145 POP AF
5150 CP #0 {RE SPALTE FERTIG?
5155 JR NZ KLEB6 {WEITER
5160 JR KLEB3 {WIEDER LINKS
5165 KLEB10 POP AF
5170 LD A,0 {SENDE
5175 LD B,5 {NL BIS NEXT KLEBER
5180 KLEB11 CALL PRINT

```

```

5185 DJNZ KLEB11
5190 RET
5195 ;-----
5200 {SUB:Puffer u.Flag f.Eintr.1 init.
5205 INITP LD HL,(DATEND)
5210 LD (HL),#20
5215 LD DE,(DATEND)
5220 INC DE
5225 LD BC,BLEN
5230 LDIR
5235 XOR A
5240 LD (PFLAG),A
5245 LD B,5 {ZEILEN
5250 LD A,#0
5255 LD HL,(DATEND)
5256 DEC HL
5260 LD DE,LLEN
5265 LLOOP ADD HL,DE
5270 ADD HL,DE
5275 LD (HL),A
5280 DJNZ LLOOP
5285 RET
5290 ;-----
5295 {SUB:Eintrag in Puffer1 (zweispalt.)
5300 EIP1 LD DE,(DATEND)
5305 LD HL,(ASTFIL)
5310 LD B,5
5315 EIP1L CALL LDLINE
5320 PUSH HL
5325 LD HL,LLEN
5330 ADD HL,DE
5335 EX DE,HL
5340 POP HL
5345 DJNZ EIP1L
5350 RET
5355 ;-----
5360 LDLINE PUSH BC
5365 LD B,LLEN-4
5370 LOOPP1 LD A,(HL)
5375 CP,7
5380 JR Z LRET
5385 CP,#0
5390 JR Z CHSP
5395 CP,#8A
5400 JR NZ NONL
5405 CHSP LD A,*
5410 LD (DE),A
5415 INC HL
5420 LOOPP2 INC DE
5425 DJNZ LOOPP2
5430 {Info für Druckpositionierung
5435 INFO LD A,27
5440 LD (DE),A
5445 INC DE
5450 LD A,16
5455 LD (DE),A
5460 INC DE
5465 LD A,1
5470 LD (DE),A
5475 INC DE
5480 LD A,#50 {variabler Tabulator
5485 LD (DE),A
5490 INC DE
5495 ;-----
5500 POP BC
5505 RET
5510 NONL LD (DE),A
5515 INC DE
5520 INC HL
5525 DJNZ LOOPP1
5530 JR INFO
5535 LRET DEC HL
5540 JP CHSP
5545 ;-----
5550 {SUB:Eintr.in Puffer2 u. Druck (zweispalt.
5555 EIP2 LD DE,(DATEND)
5560 LD HL,LLEN
5565 ADD HL,DE
5570 EX DE,HL
5575 LD HL,(ASTFIL)
5580 LD B,5

```

```

5585 EIP2L CALL LD2LINE
5590 PUSH HL
5595 LD HL,LLEN
5600 ADD HL,DE
5605 EX DE,HL
5610 POP HL
5615 DJNZ EIP2L
5620 RET
5625 ;-----
5630 LD2LINE PUSH BC
5635 LD B,LLEN-1
5640 LOOPL1 LD A,(HL)
5645 CP,7
5650 CALL Z LRET2
5655 CP,#0
5660 JR NZ TSTL2
5665 CALL CHSP2
5670 JR RETLN
5675 TSTL2 CP,#8A
5680 JR NZ NONL2
5685 CALL CHSP2
5690 JR RETLN
5695 NONL2 LD (DE),A
5700 INC DE
5705 INC HL
5710 DJNZ LOOPL1
5715 RETLN LD A,#0
5720 LD (DE),A
5725 INC DE
5730 POP BC
5735 RET
5740 ;-----
5745 CHSP2 LD A,*
5750 LD (DE),A
5755 INC HL
5760 LOOPL2 INC DE
5765 DJNZ LOOPL2
5770 RET
5775 ;-----
5780 LRET2 PUSH DE
5785 POP HL {HL=PUFFER MIT SPACES
5790 LD A,(HL)
5795 RET {ZU KLEB2
5800 ;-----
5805 PFLAG NOP {Flag für Puffer 1 oder 2
5810 {beim zweispalt. Ausdruck
5815 ;-----
5820 ;-----

```

# Spielothek 2

von KASTRUP/SAUERBREY

```
1 CLS:PRINT"WAEHLE ":PRINT
2 PRINT,"HEXAPAWN",1:PRINT,"AUSWEICHEN",2
3 GOSUB366Q
4 GOSUB369Q:IFIN=49GOTO1Q
5 IFIN=50GOTO2Q
6 GOTO4
1Q GOSUB1Q:GOSUB185Q
2Q DIMSI(39,4)
4Q P$=CHR$(19)+CHR$(27)+""
5Q RESTORE126Q
6Q FORK=1TO8:READBL(K,1),BL(K,2),BL(K,3):NEXT
7Q RESTORE135Q
8Q FORX=1TO39:FORY=1TO4:READSI(X,Y):NEXTY,X
9Q GOSUB1Q:GOTO12Q
1Q CLS:SCREEN2Q,16:PRINT"HEXAPAWN":SCREEN1,1
5
11Q RETURN
12Q DOKE22Q2,-2648Q:DOKE22Q4,-26472
13Q DOKE22Q6,-2596Q:DOKE22Q8,-26472
14Q DOKE22Q10,-26472:DOKE22Q12,-2647Q
15Q DOKE22Q14,-26472:DOKE22Q16,-28264
16Q POKE22Q6,148:POKE22Q7,148
17Q POKE22Q8,148:POKE22Q9,148
18Q POKE23Q0,148:POKE23Q5,148
19Q POKE23Q4,148:POKE23Q5,148
2Q FORX=2394TO2586STEP192
21Q DOKEK,-26475:DOKEK+2,-26472
22Q DOKEK+4,-26984:DOKEK+6,-26472
23Q DOKEK+8,-26472:DOKEK+1Q,-26474
24Q DOKEK+12,-26472:DOKEK+14,-26728
25Q POKEK+64,148:POKEK+69,148
26Q POKEK+74,148:POKEK+79,148
27Q POKEK+128,148:POKEK+133,148
28Q POKEK+138,148:POKEK+143,148
29Q NEXT
3Q DOKE2778,-26478:DOKE278Q,-26472
31Q DOKE2782,-26216:DOKE2784,-26472
32Q DOKE2786,-26472:DOKE2788,-26471
33Q DOKE279Q,-26472:DOKE2792,-27752
34Q FORX=2268TO2278STEP5
35Q DOKEK,-31Q97:DOKEK+64,-793
36Q NEXTX
37Q FORX=2652TO2662STEP5
38Q DOKEK,-31Q97:DOKEK+64,-1
39Q NEXTX
4Q RESTORE134Q
41Q FORK=1TO9:READF(K):NEXTK
42Q M(1)=Q:M(2)=Q:M(3)=Q
43Q FOR I=1 TO 7
44Q G(1)=F(7)*1Q+Q+F(8)*1Q+Q+F(9)
45Q G(2)=F(4)*1Q+Q+F(5)*1Q+Q+F(6)
46Q G(3)=F(1)*1Q+Q+F(2)*1Q+Q+F(3)
47Q FOR L=1 TO 8
48Q FOR N=1 TO 3
49Q IFBL(L,N)G(N) THEN62Q
5Q NEXT N
51Q IF (I/2)=INT(I/2) THEN57Q
52Q PRINTP$;"DU KANNST NICHT MEHR ZIEHEN!"
53Q FOR ZS=1 TO 25Q:NEXT ZS
54Q PRINTP$;"DU HAST VERLOREN!"
55Q FOR ZS=1 TO 25Q:NEXTZS
56Q GOTOL1Q
57Q PRINTP$;"ICH KANN NICHT MEHR ZIEHEN!"
58Q FOR ZS=1 TO25Q:NEXT
59Q PRINTP$;"DU HAST GEWONNEN!"
6Q FOR ZS=1 TO 25Q:NEXT
61Q GOTOL16Q
62Q NEXTL
63Q IF I=2 THEN B=1:E=3:GOTO81Q
64Q IF I=4 THEN B=4:E=23:GOTO81Q
65Q IF I=6 THEN B=24:E=39:GOTO81Q
66Q PRINTP$;"DEIN ZUG ?"
67Q GOSUB369Q:A1=IN-48:GOSUB369Q:A2=IN-48
68Q IF A1<1 OR A1>9 THEN79Q
```

```
69Q IF A2<1 OR A2>9 THEN79Q
7Q IF A2=-A1 THEN79Q
71Q P1=INT((A2+2)/3):P2=INT((A1+2)/3)
72Q IF P1<P2+1 THEN79Q
73Q P3=A2-A1
74Q IF P3<2 OR P3>4 THEN79Q
75Q IFF(A1)<1OR(P3=3)+F(A2)<2THEN79Q
76Q LO=A1:NF=A2
77Q GOSUB179Q
78Q PRINTP$;"ICH DENKE NACH ...":GOTO1Q8Q
79Q PRINTP$;"ILLEGALER ZUG!":FORZS=1TO25Q:NEXT
8Q GOTO66Q
81Q FOR L=B TO E
82Q FOR K=LTO3
83Q IF INT(SI(L,K))G(K) THEN86Q
84Q NEXTK
85Q GOTO9Q
86Q NEXTL
87Q PRINTP$;"DU HAST GEWONNEN!"
88Q FORZS=1TO25Q:NEXT
89Q GOTOL16Q
9Q LI=L:NM=SI(LI,4)
91Q IF INT(NM) THEN94Q
92Q PRINTP$;"ICH GEBE AUF!":FORZS=1TO25Q:NEXT
93Q GOTOL16Q
94Q Z=INT(RND(1)*INT(NM)+1)
95Q M=SI(LI,Z)-INT(SI(LI,Z))
96Q M1=INT(M*1Q+.QQ5):M2=INT((M-M1/1Q)*1Q+.QQ5)
97Q M(I/2)=Z/1Q+LI
98Q LO=M1:NF=M2
99Q GOSUB179Q
1Q IF M2<4 THEN1Q5Q
1Q FORK=1TO9:IF F(K)=1 THEN1Q8Q
1Q NEXTK
1Q PRINTP$;"DU HAST KEINEN BAUERN MEHR!"
1Q FORZS=1TO25Q:NEXT
1Q PRINTP$;"DU HAST VERLOREN!"
1Q FORZS=1TO25Q:NEXT
1Q GOTOL1Q
1Q NEXT I
1Q GOTO87Q
11Q PRINTP$;"NOCH EIN SPIEL ?":GOSUB369Q
111Q IF IN$="J"GOTO9Q
112Q IF IN$="N" THEN RUN
113Q PRINTP$;"ILLEGALE ANTWORT!"
114Q FORZS=1TO25Q:NEXT
115Q GOTOL1Q
116Q IF M(3)=QTHEN118Q
117Q LS=INT(M(3)):LZ=INT((M(3)-LS)*1Q+.5):GOTO1
21Q
118Q IF M(2)=QTHEN12Q
119Q LS=INT(M(2)):LZ=INT((M(2)-LS)*1Q+.5):GOTO1
21Q
12Q LS=INT(M(1)):LZ=INT((M(1)-LS)*1Q+.5)
121Q SI(LS,LZ)=INT(SI(LS,LZ)):NM=INT(SI(LS,4))
122Q SI(LS,LZ)=SI(LS,LZ)+SI(LS,NM)-INT(SI(LS,NM))
123Q SI(LS,NM)=INT(SI(LS,NM))
124Q SI(LS,4)=SI(LS,4)-1
125Q GOTOL1Q
126Q DATA232,121,313
127Q DATA323,313,333
128Q DATA233,132,331
129Q DATA233,123,313
13Q DATA332,321,313
131Q DATA323,312,331
132Q DATA323,213,133
133Q DATA332,231,133
134Q DATA1,1,1,3,3,3,2,2,2
135Q DATA22.84,133.85,311.96,3.0Q
136Q DATA22.74,313.75,131.95,4.96
137Q DATA22.74,331.85,113.86,3.0Q
138Q DATA232.96,133.0Q,331.0Q,1.0Q
139Q DATA232.41,213.75,331.95,4.96
14Q DATA232.41,231.42,313.0Q,2.0Q
141Q DATA232.75,113.95,313.96,3.0Q
142Q DATA23.75,112.84,331.0Q,2.0Q
143Q DATA23.84,131.85,331.86,3.0Q
144Q DATA322.86,211.95,133.0Q,2.0Q
145Q DATA322.84,123.95,331.52,4.53
```

```

1460 DATA322.95,313.96,331.00,2.00
1470 DATA322.95,313.96,133.00,2.00
1480 DATA322.86,321.51,133.52,3.00
1490 DATA223.74,313.75,331.00,2.00
1500 DATA223.74,313.75,133.00,2.00
1510 DATA223.74,321.86,133.51,4.52
1520 DATA223.52,123.53,331.00,2.00
1530 DATA322.84,131.85,133.86,3.00
1540 DATA232.74,311.75,313.95,3.00
1550 DATA232.62,132.63,313.00,2.00
1560 DATA232.74,312.75,133.95,4.63
1570 DATA232.74,331.00,133.00,1.00
1580 DATA332.41,213.95,333.96,3.00
1590 DATA332.41,221.52,333.00,2.00
1600 DATA233.75,213.41,333.00,2.00
1610 DATA233.41,221.52,333.00,2.00
1620 DATA233.75,111.00,333.00,1.00
1630 DATA323.84,112.63,333.00,2.00
1640 DATA323.86,321.52,333.00,2.00
1650 DATA323.84,123.52,333.00,2.00
1660 DATA332.52,122.63,333.00,2.00
1670 DATA332.95,312.63,333.00,2.00
1680 DATA233.52,122.63,333.00,2.00
1690 DATA233.75,312.63,333.00,2.00
1700 DATA332.85,111.00,333.00,1.00
1710 DATA323.86,211.41,333.00,2.00
1720 DATA323.84,113.00,333.00,1.00
1730 DATA323.86,311.00,333.00,1.00
1790 F(NF)=F(LO):F(LO)=3
1800 LO=(LO-INT((LO-1)/3)*3)*5+2268+(2-INT((LO-1)/3))*192
1810 NF=(NF-INT((NF-1)/3)*3)*5+2268+(2-INT((NF-1)/3))*192
1820 DOKENF-5,DEEK(LO-5):DOKENF+59,DEEK(LO+59)
1830 DOKELO-5,8224:DOKELO+59,8224
1840 RETURN
1850 SCREEN1,2:PRINT"Brauchst du eine Anleitung?"
1860 GOSUB3690:IFIN=78THENRETURN
1870 IFIN=74GOTO1860
1880 GOSUB1000:SCREEN1,2
1890 PRINT"HEXAPAWN EDUCABLE ROBOT":PRINT
1900 PRINT"Bei diesem Spiel treten zwei Spieler"
1910 PRINT"mit je drei Bauern auf einem 3x3-Brett"
1920 PRINT"gegeneinander an. Das Brett wird so"
1930 PRINT"          bezeichnet:":PRINT
1940 PRINT"          7 8 9"
1950 PRINT"          4 5 6"
1960 PRINT"          1 2 3"
1970 PRINT:PRINT"Es hat der gewonnen, der mit einem"
1980 PRINT"Bauern die Grundlinie des Gegners"
1990 PRINT"erreicht oder ihn am Ziehen hindert."
2000 PRINT"Klar?":GOSUB3690:GOSUB1000:SCREEN1,2
2010 PRINT"Das besondere am Spiel ist es,"
2020 PRINT"daß ich aus meinen Fehlern lerne."
2030 PRINT:PRINT"Du gibst deinen Zug an, indem du"
2040 PRINT"die Nummer des Feldes, von dem,und"
2050 PRINT"die des Feldes, wohin du ziehst"
2060 PRINT"willst, nacheinander eintippst.":PRINT
2070 PRINT"Du faengst an.          Fertigt?"
2080 GOTO3690
2090 GOSUB2100:SCREEN1,2:PRINT:GOTO2120
2100 CLS:SCREEN1,2:PRINT"A U S W E I C H E N"
2110 SCREEN1,1:PRINT:RETURN
2120 PRINT"Brauchst du eine Anleitung?"
2130 GOSUB3690:IFIN=78THENGOSUB3220:GOTO2150
2140 IFIN=74GOTO2130
2150 PRINT"Groesse des Spielfelds (3-8)?"
2160 GOSUB3690
2170 A=IN-48
2180 IFA=8ORA=3GOTO2160

```

```

2190 P(1,0)=A-1:P(2,0)=A-1
2200 FORJ=1TOA-1:P(1,J)=10*J+1:P(2,J)=10*A+J+1:
NEXT
2210 F=1:CS(1)="Ziffern":CS(2)="Buchstaben"
2220 Z$(1)="T":Z$(2)="→":Z$(3)="↓"
2230 M(1,1)=0:M(1,2)=2:M(1,3)=1:M(1,4)=3
2240 M(2,1)=4:M(2,2)=2:M(2,3)=1:M(2,4)=0
2250 FORJ=1TO9:A(1,J)=48+J:A(2,J)=64+J:NEXT
2260 GOSUB2100:AD=2400-A-INT((A-1)/2)*64
2270 FORJ=1TOA-1:D(J,1)=1:POKEAD+J*64+2,J+48
2280 FORK=2TOA:D(J,K)=0:POKEAD+J*64+2*K,46:NEXT
K,J
2290 D(A,1)=0:POKEAD+A*64+2,46
2300 FORK=2TOA:D(A,K)=1:POKEAD+A*64+K*2,63+K:NEXT
2310 FORJ=0TOA-1:D(0,J)=1:D(J,0)=1:D(J,A+1)=1
2320 D(A+1,J)=1:NEXT
2330 SCREEN1,INT(A/2)+9:P$=CHR$(19)+CHR$(27)
2340 FORJ=1TO24-A:P$=P$+" ":NEXT:PRINT
2350 PRINTP$;"Wieviele Spieler?"
2360 GOSUB3690:B=IN-48:IFB=2GOTO2410
2370 IFB=1GOTO2360
2380 PRINTP$;"Ich ziehe die Ziffern."
2390 FORJ=1TO2500:NEXT:PRINTP$;"Faengst du an?"
2400 GOSUB3690:F=2+(IN=78):IFB=2ANDIN=74GOTO2410
2410 PRINTP$;"Einen Moment ..."
2420 FORJ=FTO3-FSTEP3-2*F:FORJ1=1TOA-1
2430 R=INT(P(J,J1)/10):C=P(J,J1)-10*F
2440 IFJ=2GOTO2480
2450 IFC=AGOTO2630
2460 IFC=ATHENNEXT:GOTO2550
2470 GOTO2490
2480 ONR+1GOTO2540,2630
2490 IFD(R-1,C)=0ORD(R,C+1)=0GOTO2630
2500 IFJ=2THEN2530
2510 IFD(R+1,C)=0GOTO2630
2520 NEXT:GOTO2550
2530 IFD(R,C-1)=0GOTO2630
2540 NEXT
2550 PRINTCHR$(19);CHR$(27);"Die ";CS(3-J);" haben ";
2560 PRINT"keinen Zug fuer die "
2570 PRINTCS(J);" gelassen !"
2580 PRINT"Die ";CS(J);" haben gewonnen !"
2590 PRINT"Noch ein Spiel?"
2600 GOSUB3690:IFIN=74THENCLEAR:GOTO2150
2610 IFIN=78GOTO2600
2620 PRINT"Bis zum naechsten Mal !":RUN
2630 IFB=2ORJ=2GOTO2820
2640 L1=2:FORL0=1TO3:ONL0GOTO2690,2650,2680
2650 IFRND(1)=.5GOTO2670
2660 L1=1:GOTO2690
2670 L1=3:GOTO2690
2680 L1=4-L1
2690 P1=INT(RND(1)*A):FORL2=1TOA-1:P1=P1+1
2700 IFP1=A-1THENP1=P1-A+1
2710 IN=P1+48
2720 R=INT(P(J,P1)/10):C=P(J,P1)-10*F
2730 IFC=ATHENNEXTL2,L0:GOTO2550
2740 ONL1-1GOTO2770,2800
2750 IFD(R-1,C)=0GOTO3030
2760 NEXTL2,L0:GOTO2550
2770 IFD(R,C+1)=0GOTO3060
2780 IFC=ATHENNEXTL2,L0:GOTO2550
2790 P(J,0)=P(J,0)-1:GOTO3060
2800 IFD(R+1,C)=0GOTO3090
2810 NEXTL2,L0:GOTO2550
2820 PRINTP$;CS(J);"-Zug ?"
2830 GOSUB3690
2840 IFIN=64GOTO3200
2850 FORB0=1TOA-1:IFA(J,B0)=IN THENP1=B0:GOTO2870
2860 NEXT:GOTO2830
2870 R=INT(P(J,P1)/10):C=P(J,P1)-10*F
2880 IFR=0GOTO2830
2890 IFC=ATHEN2830
2900 POKEAD+R*64+C*2,46:FORB0=1TO20:P2=USR(0)
2910 IFP2 THENPOKEAD+R*64+C*2,IN:GOTO2940
2920 NEXT:POKEAD+R*64+C*2,IN:FORB0=1TO20
2930 P2=USR(0):IFP2=0THENNEXT:GOTO2900

```

```

2940 P2=P2-16:IFP2<1ORP2<4GOTO2830
2950 P2=M(J,P2):IFP2=0GOTO2830
2960 IFJ=2GOTO2990
2970 IFC=AORP2<2GOTO3010
2980 P(1,0)=P(1,0)-1:GOTO3060
2990 IFR=1ORP2<1GOTO3010
3000 P(2,0)=P(2,0)-1:GOTO3030
3010 ONP2-1GOTO3050,3080,3110
3020 IFD(R-1,C)GOTO2830
3030 IFR=1THENPOKEAD+R*64+C*2-64,IN
3040 D(R-1,C)=1:P(J,PI)=P(J,PI)-1:GOTO3140
3050 IFD(R,C+1)GOTO2830
3060 IFC=ATHENPOKEAD+R*64+C*2+2,IN
3070 D(R,C+1)=1:P(J,PI)=P(J,PI)+1:GOTO3140
3080 IFD(R+1,C)GOTO2830
3090 IFR=1THENPOKEAD+R*64+C*2+64,IN
3100 D(R+1,C)=1:P(J,PI)=P(J,PI)+1:GOTO3140
3110 IFD(R,C-1)GOTO2830
3120 IFC=1THENPOKEAD+R*64+C*2-2,IN
3130 D(R,C-1)=1:P(J,PI)=P(J,PI)-1
3140 D(R,C)=0:POKEAD+R*64+C*2,46
3150 IFB=2ORJ=2GOTO3180
3160 PRINTP$;"Ich ziehe:";STR$(PI);Z$(LI)
3170 FORKO=LTO4000:NEXT
3180 IFP(J,0)THENNEXTJ:GOTO2410
3190 PRINTP$;"Die ";C$(J);" gewinnen!":GOTO2590
3200 PRINTCHR$(19);CHR$(27);"Die ";C$(J);
3210 PRINT" geben auf !!" :GOTO2590
3220 PRINT
3230 PRINT"Hier ist ein Beispiel fuer ein Spiel
feld:"
3240 PRINT
3250 PRINT"      1 . . . ."
3260 PRINT"      2 . . . ."
3270 PRINT"      3 . . . ."
3280 PRINT"      4 . . . ."
3290 PRINT"      . A B C D"
3300 PRINT
3310 PRINT"Zwei Steinesorten(Buchstaben und Zif
fern)"
3320 PRINT"werden von zwei Spielern abwechsel
nd in"
3330 PRINT"rechten Winkeln auf dem quadrati
schen"
3340 PRINT"Spiefeld bewegt. Die Groesse des S
piel-"
3350 PRINT"feldes wird von dir festgelegt (Das
oben"
3360 PRINT"abgebildete hat Groesse 5). we
iter?"
3370 GOSUB3690
3380 PRINT"Ziel des Spieles ist es, seine S
teine"
3390 PRINT"ueber das Feld und an der gegenue
ber-"
3400 PRINT"liegenden Seite wieder hinauszubew
egen."
3410 PRINT"Ziffern verlassen das Feld nur a
n der"
3420 PRINT"rechten Seite, Buchstaben an der ob
eren."
3430 PRINT"Der Spieler, der zuerst keine Steine
mehr"
3440 PRINT"auf dem Spiefeld hat, hat gewonnen.
"
3450 PRINT
3460 PRINT"Es wird abwechselnd gezogen, inde
m man"
3470 PRINT"einen Stein zu einer benachbarten Po
siti-"
3480 PRINT"on zieht, die entweder ausserhalb
des"
3490 PRINT"Spiefeldes oder auf einem leeren
Feld"
3500 PRINT"liegt. Es gibt keine Spruenge und d
iego-"
3510 PRINT"nale Zuege. Ziffern koennen sich
nicht"
3520 PRINT"links bewegen,Buchstaben nicht nach
unten";
3530 PRINT"n. Ok?";:GOSUB3690:PRINT

```

```

3540 PRINT"Um einen Stein zu bewegen, gib s
einen"
3550 PRINT"taste) an.
3560 PRINT
3570 PRINT"Manoevrierst du deinen Gegner in
eine"
3580 PRINT"Situation, in der er sich nicht
mehr"
3590 PRINT"bewegen kann, so hast du verloren."
3600 PRINT
3610 PRINT
3620 PRINT"Schliesslich kannst du, wenn du auf
geben"
3630 PRINT"willst, SHIFT und 'a)' druecken.
3640 PRINT:PRINT:PRINT:PRINTTAB(36);"Alles klar
?"
3650 PRINT:PRINT:PRINT:GOTO3690
3660 DOKE3300,25311:DOKE3302,312:DOKE3304,18351
3670 DOKE3306,10927:DOKE3308,-8179:POKE3310,233
3680 DOKE4100,3300:RETURN
3690 IN=USR(0):IFIN=0GOTO3690
3700 IN$=CHR$(IN):RETURN
Ok

```

Zeile 3670 : für die "Minderheiten-BASIC-Version: aus -8179 muß -8182 gemacht werden.

Zeile 2220 : die Grafikzeichen erreicht man folgendermaßen:

→ Control I  
↓ Control K

Red.

## Hex-Dez

von GÜNTER BÖHM

Schon im Programm "ZAHLEN" von Eberhard Horch haben Sie die Möglichkeit, Hex-Zahlen in Dezimalzahlen umzuwandeln.

Das folgende kleine Programm ist aber auf die Programmierpraxis zugeschnitten: es gibt nicht nur die POKE- (bis max. 255dez) sondern auch die DOKE-Werte an, die beim NASCOM-BASIC die Programmierung (besonders mit eingebauten Maschinenprogrammen) verkürzen.

```

10 CLS:PRINT"HEX-DEZIMAL WANDLUNG
20 PRINT:PRINT
30 INPUT"HEX-ZAHL" :H$
35 IF H$="" THEN END
40 T=0:D=1
50 FORP=LEN(H$)-1TOSTEP-1
60 C=ASC(MID$(H$,D,1))
70 D=D+1
80 IF(C>=48)*(C<=57) THENC=C-48:GOTO110
90 IF(C>=65)*(C<=70) THENC=C-55:GOTO110
100 PRINT"FALSCH EINGABE !":GOTO20
110 T=T+C*16↑P:NEXT
120 PRINT:PRINT"HEX " :H$
130 PRINT"DEZ " :T
140 IFT>32768THENT=INT(T-65536)
150 PRINT"DOKE " :T
160 GOTO20

```

# Hardware Tips

von JÖRG WITTICH

## SCHNELLE CASSETTE

Nachdem man beim Laden von Programmen vom NASCOM oftmals auf eine Geduldsprobe gestellt wird, habe ich einmal das schnelle Cassetteninterface aus Elektor 10/82 nachgebaut. Es funktioniert bei mir nun seit einiger Zeit mit über 5000 Baud völlig problemlos. Zum Abspeichern verwende ich einen preiswerten Stereo-Radiorecorder. Wenn man auch zur Sicherheit bei der Aufnahme beide Stereokanäle zusammenschalten kann, so sollte man jedoch bei der Wiedergabe auf Grund von möglichen Phasenverschiebungen immer nur einen Kanal verwenden.

Wer schon einmal Versuche mit höheren Baudraten gemacht hat, wird vielleicht festgestellt haben, daß ab einer bestimmten Grenze nicht mehr alle Blöcke fehlerfrei einzulesen sind. Dies muß nicht unbedingt am Übertragungsverfahren liegen, sondern an NASSYS1. Beim READ-Befehl wird nämlich neben dem UART auch noch die Tastatur abgefragt, und ab etwa 8000 Baud, bei 2 MHz-Systemen entsprechend bei etwa 4000 Baud, können dadurch einzelne Bytes verlorengehen.

Und noch ein Tip für alle NASCOM1-Besitzer, die auf möglichst einfache Weise auch NASCOM2-Programme lesen können möchten: In der Funkschau 16/83 war ein RTTY-Demodulator mit 4 Transistoren angegeben, welcher auch noch FSK-Signale nach dem Kansas-City-Standard mit 1200 Baud verarbeiten kann (es steht zwar nicht dabei, aber ich habe es ausprobiert.)

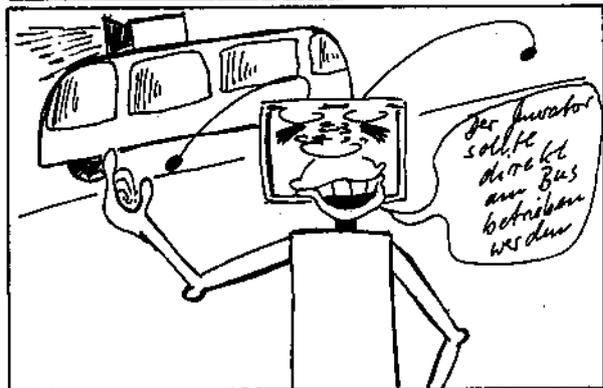
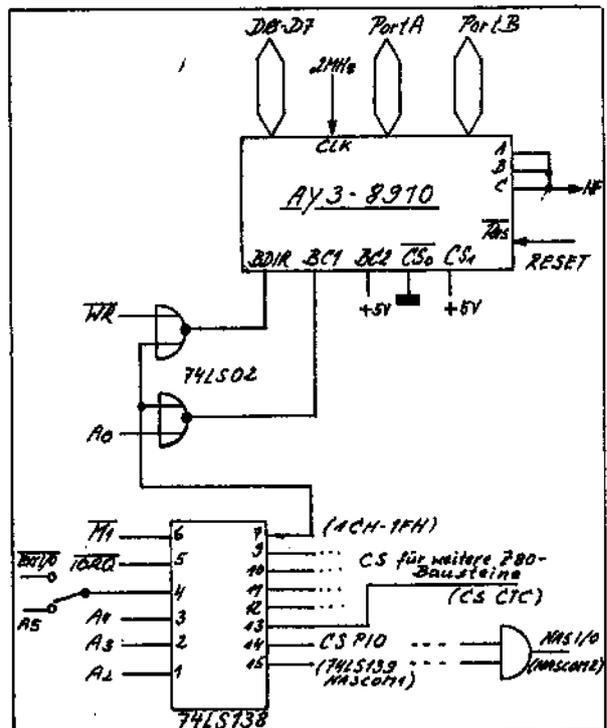
## SOUNDGENERATOR

Der eine oder andere wird inzwischen sicher versucht haben, den Soundgenerator AY-3-8910 an seinem NASCOM zu betreiben. Störend bei den bisher vorgestellten Interface-Schaltungen war, daß dadurch eine ganze PIO belegt wurde. Folgende einfache Schaltung zeigt, wie der PSG direkt am Bus betrieben werden kann. Es ist sowohl Schreiben wie auch Lesen der Register möglich, wodurch sich auch die beiden I/O-Ports voll nutzen lassen.

Hardwarebedingt muß zum Lesen der Register die Adresse verwendet werden, an die man sonst die Registernummer des PSG schreibt. Der PSG läuft bei mir mit 2 MHz Taktfrequenz, die CPU mit 4 MHz.

;Schreiben und Lesen eines Registers

```
LD A,Regnr
OUT (Regnr),A
LD A,Data
OUT (Datadr),A
;
IN A,(Regadr)
```



Wer kann mir zu einer Übersetzung der Handbücher von ZEAP und NASDIS verhelfen?  
Klaus Peter Schmidt

# 64 Zeichen

von UWE TEPP

Um auch mit dem NASCOM1 64 Zeichen pro Zeile darstellen zu können, müssen folgende Änderungen vorgenommen werden:

Quarz austauschen gegen 20,465 MHz

C3 gegen 47 pF austauschen.

Ersetzen der ICs 1, 2, 3, 4, 7, 10 durch Steckanschlüsse.

Dann kann die nachfolgende Schaltung eingesetzt werden.

Die abgedruckten Schaubilder geben Aufschluß über das Timing und die Bestückung zur günstigen Leitungsführung.

Hier die entsprechenden neuen Frequenzen:

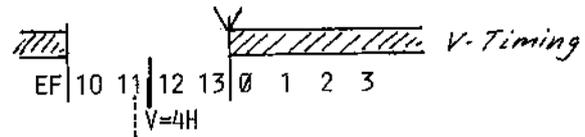
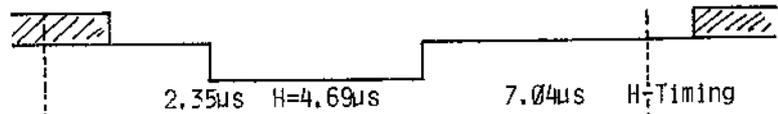
20,465 MHz

Ck Shift 10,232 MHz

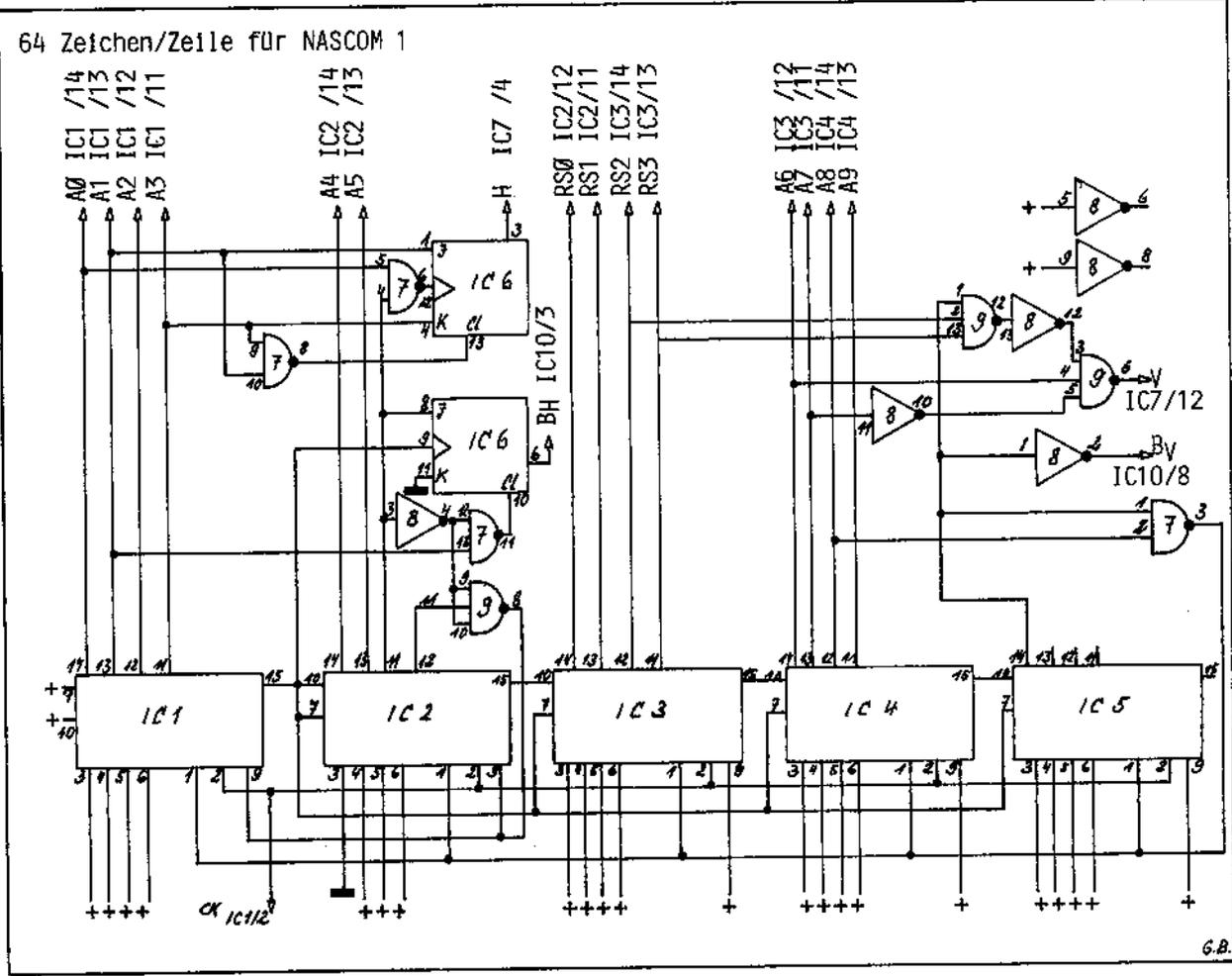
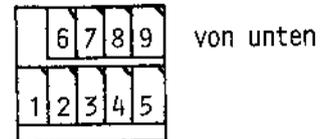
Ck Char 1,279 MHz = 781,80 usec

T64 50,04 usec

TH 18 Char



- IC 1-5 74 LS 161
- IC 6 74 LS 107 A
- IC 7 74 LS 00
- IC 8 74 LS 04
- IC 9 74 LS 10



# VIDEOINTERFACE

von Dr. Oberle und A. Zippel

Welcher NASCOM Anwender hat sich nicht schon über das zu anderen Rechnern inkompatible Bildschirmformat von 16\*48 Zeichen geärgert. Besonders dann wenn er Software, z.B. BASIC-Programme, auf seinem NASCOM übernehmen wollte oder seinen NASCOM als Terminal für Grossrechner, und andere Mikrorechner verwenden wollte, wie z.B. CP/M-Systeme?

Diese Tatsache hat mich dazu veranlasst eine programmierbare Videointerfacekarte für NASCOM Computer zu entwickeln die am NASBUS als "Memory mapped" Videointerface betrieben werden kann. "Memory mapped" bedeutet: der Bildspeicher, in diesem Fall 2KB, ist Teil des Adressbereiches der CPU und kann auch direkt über Speicherbefehle angesprochen werden. Dies hat den Vorteil, dass eine sehr schnelle Ausgabe auf den Bildschirm erfolgen kann und auch eine Semigrafik (bei entsprechendem Charaktergenerator) leicht implementiert werden kann mit einer höheren Auflösung als beim NASCOM Videointerface. Allerdings müsste die Software dazu noch geschrieben werden.

Die Interfacekarte dient bei mir in Verbindung mit einem Billig-Monitor (Fernseher sind wegen der zu geringen Bandbreite nicht geeignet) als 24\*80 Zeichen Display für einen CP/M-Rechner. Sie kann auch für ein beliebiges anderes Bildformat bis zu 128\*x (je nach Bildspeichergrösse) Zeichen unprogrammiert werden. Sollten allerdings mehr als 2K Zeichen dargestellt werden so ist der Bildspeicher und der MUX auf der Karte entsprechend zu erweitern.

### Funktionsbeschreibung

Die Karte ist mit dem weitverbreiteten Videocontrollerchip MC6845 aufgebaut. Dieses IC übernimmt die gesamte Bildschirmsteuerung und die Generierung des Videosignals, sowie der Synchronisationssignale. Es ist sogar leicht möglich einen Lichtgriffel für interaktiven Bildschirmbetrieb anzuschliessen, wozu jedoch ebenfalls erst die Software erstellt werden müsste.

Der Controllerbaustein ist per I/O-Befehl über den NASBUS voll programmierbar, sodass die Cursorsteuerung, das Bildschirmformat und die Behandlung von CTRL und ESC Charakteren nur von der Programmierung abhängig sind.

Ebenso ist es per Software möglich die Anpassung an verschiedene Videomonitore vorzunehmen die mindestens eine Bandbreite von 16Mhz haben müssen. Die im nächsten Heft veröffentlichten Programme sind für einen ZENITH Monitor und einen CP/M Rechner ausgelegt.

### Adressdecoder und NASBUS-Anpassung

Die Anfangsadresse für den Bildspeicher kann mit 4 DIL-Schaltern in Blöcken zu 4KB verschoben werden, wobei die Anfangsadresse immer nur  $\times 800h$  sein kann weil die Adressleitung A11 als Freigabesignal für den Adresscomperator IC7485 verwendet wird. Dies ist z.B. in CP/M Systemen sinnvoll weil damit die letzten 2KB Ram als Bildspeicher verwendet werden können. Bei 64K Ram wäre die Adresse F800h zu verwenden.

Die I/O Adressdecodierung erfolgt mit dem IC138 in Verbindung mit dem 8-fach DIL-Schalter ICDIL8. Der Controller MC6845 arbeitet mit 2 I/O Portadressen und 16 internen Steuerregistern. Zuerst muss über einen I/O Befehl im Control Reg. das gewünschte Steuerregister angewählt werden und anschliessend über einen zweiten I/O Befehl das Datenbyte gelesen oder geschrieben werden. Die I/O Portadresse ist wählbar und setzt sich aus folgenden Adressbits zusammen:

A7 0 A5 0 A3 0 X X

z.B. 0 0 1 0 0 0 0 0 => 20h

als Adresse des Control Reg. und damit ist automatisch 2fh die Adresse des Steuerregisterdatenports. Für jede mögliche I/O Adresse darf nur ein DIL-Schalter geschlossen sein womit sich genau 8 verschiedene Adressen einstellen lassen die aus A7,A5,A3 gebildet werden. Die RD, WR Signale vom NASBUS und der Adressdecoder erzeugen auch die speziellen Bussignale MEMEXT, I/OEXT u. DBDR mit Hilfe von zusätzlichen Logikgattern. Dabei sind diese Anschlüsse NASBUS seitig wie folgt zu ergänzen:

I/OEXT u. MEMEXT sind jeweils über einen Widerstand von 500 Ohm an Masse zu legen. Die Signale werden dann im Betrieb über die Ge-Dioden (Ge wegen Störabstand wichtig) auf

"aktiv high" gelegt. Diese Methode ist zwar unkonventionell und entspricht in keiner Weise den TTL-Spezifikationen, aber sie ist einfach und funktioniert bei mir auf mehreren NASBUS Karten problemlos. (u.U. L-Pegel kontrollieren und gegebenenfalls Widerstand verkleinern). DBDR wird über die Ge-Diode wie eine normale "wired OR" Schaltung angesteuert und ist auch so vorgesehen.

### Timing

Die Taktfrequenz für den Videoteil und den Controllerchip wird mit IC144, IC163 und IC8174 (Latch) erzeugt und über Logikgatter entsprechend verteilt. (IC8174 muss unbedingt ein Shottky-Typ sein)

### Memory und Adress-MUX

Der Bildspeicher ist mit einem 2K\*8 Statik Ram 6116 aufgebaut. Er kann auf zwei Arten adressiert werden. Zum einen kann der CRT-Controller MC6845 die Adressierung vornehmen und den Inhalt des Bildspeichers verändern und zum anderen kann auch direkt von der CPU aus auf den Bildspeicher zugegriffen werden. Die Umschaltung der Adressierungsart wird über den Adressmultiplexer, der mit 3 TTL-Chips 74157 aufgebaut ist vorgenommen. Mit diesem MUX wird auch die Freigabe des Datenbusses auf das Video-Ram über den Baustein IC2452 vorgenommen.

### CRT-Controller und Videosignalerzeugung

Der CRT-Controllerchip erzeugt über 4-Adressbits (RA0-RA3) die Adressen für die Bildschirmzellen (max. 16/Charakter) die direkt auf das Charakter-ROM IC2716 gegeben werden. Dazu kommen 7-Bits aus dem Video-Ram die über ein 8-Bit Latch IC273 auf den Charaktergenerator gelangen. Dieser bildet daraus mit Hilfe des Videoschieberegisters IC165 das serielle Videonutzsignal welches dann über zusätzliche Steuerlogik auf den Vid-overstärker gelangt. Dort wird es mit den Horizontal-SYNC und Vertikal-SYNC Impulsen aus dem CRT-Controllerchip gemischt und als Composit-Video-Signal auf den angeschlossenen Monitor mit Videoeingang ausgegeben. Der Bildschirm zeigt dann den Inhalt des Bildspeichers in der durch den Charaktergenerator und die Treibersoftware

für den MC6845 festgelegten Form an.

Das Video-Ram auf der CRT-Karte ist in der hier gezeigten Version und auch im Layout als "write only" geschaltet, d.h. es kann nur beschrieben aber nicht gelesen werden. Dies hat den Vorteil, dass die CRT-Karte jederzeit an eine Ramspeicheradresse gelegt werden kann an der sich bereits Rampeicher befindet. Dann wird praktisch parallel in den Bildspeicher 6116 und das dort befindliche Ram geschrieben aber nur aus dem CPU-Ramspeicher gelesen. Viele Speicherkarten lassen sich nämlich nur in Blöcken zu 16K verschieben, sodass diese Lösung durchaus sinnvoll sein kann. Will man diese "WR only" Methode nicht anwenden so müssen folgende Änderungen im Layout und im Schaltplan vorgenommen werden:

IC7400(8) --> IC7411(11) auftrennen!  
 IC7402(2) --> IC7411(11) einfügen!

Als Charaktergenerator wird hier ein einfaches EPROM 2716 verwendet welches z.B. mit dem in Heft 1/82 von mir beschriebenen BASIC-Programm erstellt und programmiert werden kann.

### Aufbau der Hardware

Eine Layoutkopie und Bestückungsplan der Interfacekarte ist gegen Freiuschlag und Kopierkostenersatz bei der Redaktion erhältlich. Beim Aufbau geht man am besten so vor wie es von mir bereits in Heft 12/81 beschrieben wurde.

\*\*\*\*\*

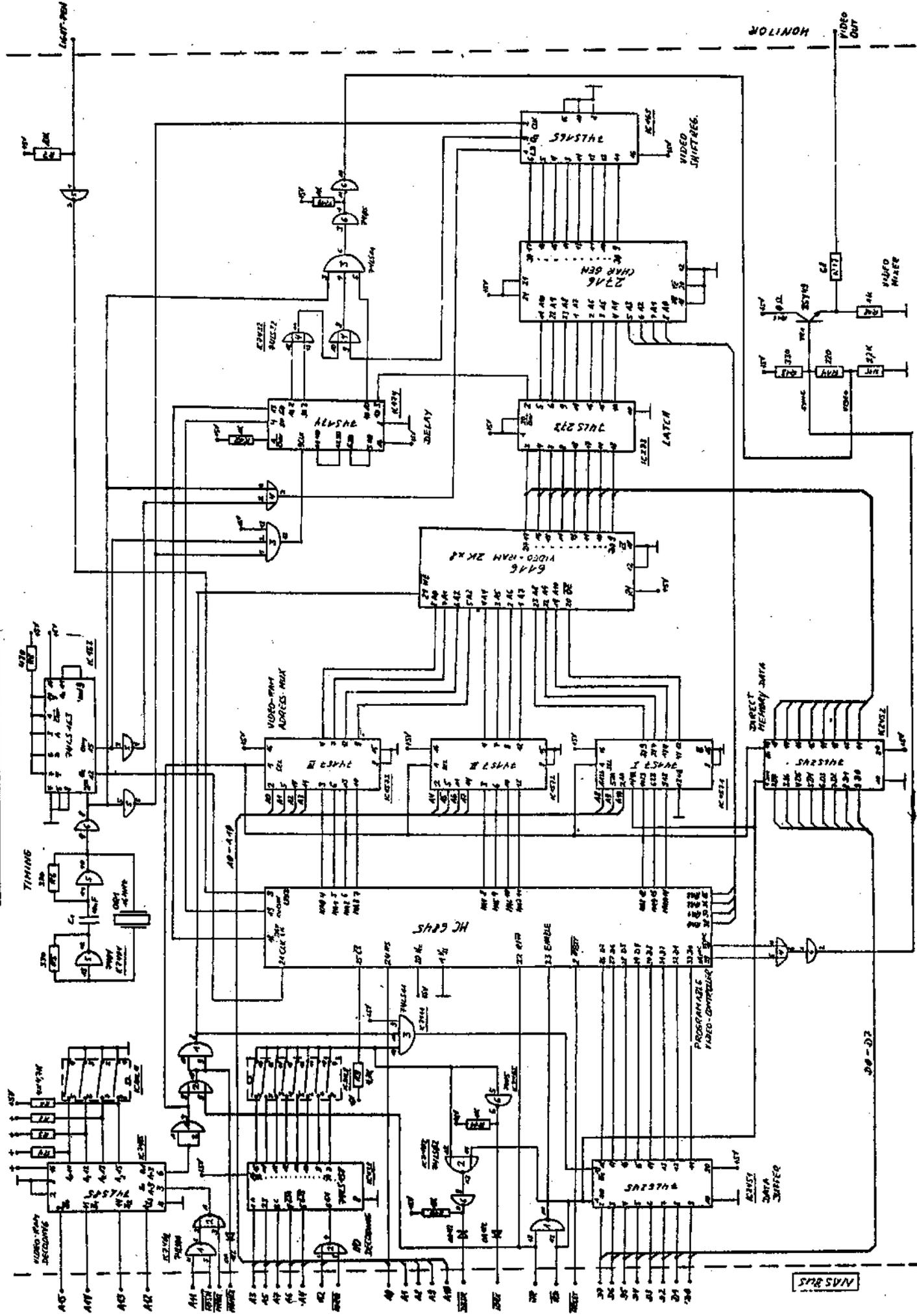
Die Verbindung der Adressleitung A15 an NASBUS Pin 45 und IC7485 Pin 9 muss durch einen dünnen Wrap-Draht nachträglich in Layout noch eingefügt werden.

\*\*\*\*\*

### LITERATURANGABEN

- NASBUS Manual
- NASCOM I und NASCOM II Manuals
- Datenblätter und Applikation Notes MC6845
- NASCOM-JOURNAL 12/81, 1/82

PROGRAMMIERBARE VIDEO-INTERFACE-KARTE



# Hisoft Tips

von CONSTANTIN OLBRICH

## Hisoft Pascal Tips und Tricks:

Das Hisoft Pascal 4T wird in einem eigenen Cassettenformat geliefert, und benoetigt einen eigenen Loader um es zu laden. Dieser Loader hat die unangenehme Eigenschaft einen Autostart durchzufuehren. Dies waere nicht erwaehnenenswert, wenn nicht das Pascal gleich nach dem Start, nach einigen Fragen zur Speicheraufteilung, sich selbst an eine neue Stelle im Speicher legt, bevor man mit NASSYS Back-up Kopien erstellen kann. Nachtraegliche Kopien entsprechen stets der gewaehlten Speicheraufteilung.

Um also eine original Kopie des HP4T einschliesslich der Relokatorstabelle mit NASSYS machen zu koennen, ist folgende Prozedur noetig:

In der NASSYS Workspace wird eine 80H in 0C75H (bei NASSYS 3) eingetragen. Dies ist die einfachste Moeglichkeit einen Autostart von Objektprogrammen im 'B' Format zu verhindern. Hiermit wird die serielle Schnittstelle in der Eingabe abgeschaltet, und das E xxxx auf dem Band bleibt ohne Wirkung. Nun kann man nach Laden des HP4T Loader mit 'R' die Cassette stoppen und folgendes eingeben:

B 0CCD            Breakpoint nach letztem Block  
E 0C80            Start des Loaders

Start der Cassette

Nach einigen Minuten meldet sich NASSYS mit der Breakpointanzeige und man gibt ein:

B 0CD3            Break auf RET; hier =JP 1000  
E                ausfuehren bis Break

und nun das File retten mit

W 1000 6300        aber nun bitte nicht mit 300

Baud !! Der Kaltstart einschliesslich der Reloziierung ist nun

1000 H. Nach

dem Start bei 1000 H ist die Adresse 1000 H die Startadresse fuer

Pascal Objektprogramme, ein weiterer Start bei 1000 H ist also

unmoeglich. Kalt- und Warmstart sind dann 101F und 1021 H.

Wer im Hisoft Pascal 4T das Control 'C' als ungewohnt empfindet und

Lieber ESCAPE als Break Funktion mag, kann im original File vor dem

Reloziieren folgendes aendern: Die Speicherstellen 133A,1513,192F,2A65,

3FEC, 4AD7, 4C5E und 544D sind von 03 auf 1B zu aendern.

# READPROM

von GÜNTER BÖHM

Sowohl bei der Grafikkarte, als auch bei der FDC- Karte werden PROMs verwendet. Wenn beim Testen Fehler auftreten, ist man sich nicht sicher, ob nicht etwa die PROMs falsch programmiert oder einfach kaputt sind.

Deshalb habe ich mir ein einfaches Programm geschrieben, das den Inhalt der PROMs ausliest.

Die Hardware ist extrem einfach: sie besteht nur aus einem Flachbandkabel, das einen PROM-Sockel mit den entsprechenden Pins eines PIO-Ausgangs verbindet.

Wenn Sie auf eine genauere Beschreibung der Hardware-Anschlüsse scharf sind, drucke ich gerne eine Verbindung der entsprechenden Stecker ab.

Ansonsten dürfte das Listing selbst genügend Auskunft geben.

## TEAP 280 Assembler - Source Listing

```

0010 ;READPROM Vers. 0.0
0020 ;Programm zum Einlesen von
0030 ;FL-Proms 256x4Bit oder 512x4Bit
0040 ;
0050 ;G.Boehm, Karlsruhe 7.5.84
0060 ;
0070        ORG    #C80
0080 00F4    0080    PIOAD    EQU   #F4    ;ENABLE/DATEN
0090 00F5    0090    PIOAC    EQU   #F3
00A0 00F6    0100    PIOBD    EQU   #F6    ;ADRESSEN
00B0 00F7    0110    PIOBC    EQU   #F7    ;mc-PIO/SIO-Karte
00C0 ;
00D0 ;
00E0 3E0F    0140    INIT    LD    A,#F    ;OUTPUT
00F0 03F7    0150            OUT    ;(PIOBC),A
0100 3E0F    0160            LD    A,#CF    ;CONTROL
0110 03F5    0170            OUT    ;(PIOAC),A
0120 3E9F    0180            LD    A,#9F    ;MASKE
0130 03F5    0190            OUT    ;(PIOAC),A
0140 ;MASKE    P10A : BIT0 bis BIT3 - DATEN
0150 ;            BIT4            undefin.
0160 ;            BIT5    Adr.8 ad.    ENABLE2
0170 ;            BIT6            ENABLE1
0180 ;
0190 ;
01A0 0C0F    0250        DEFB   #F    ;PRG
01B0 0C0C    0260        DEFB   #C    ;CLS
01C0 32353620 0270        DEFB   '256    ;
01D0        20202020
01E0        31
01F0 0C97    0280        DEFB   #D
0200 35313220 0290        DEFB   '512    ;
0210        20202020
0220        32

```

```

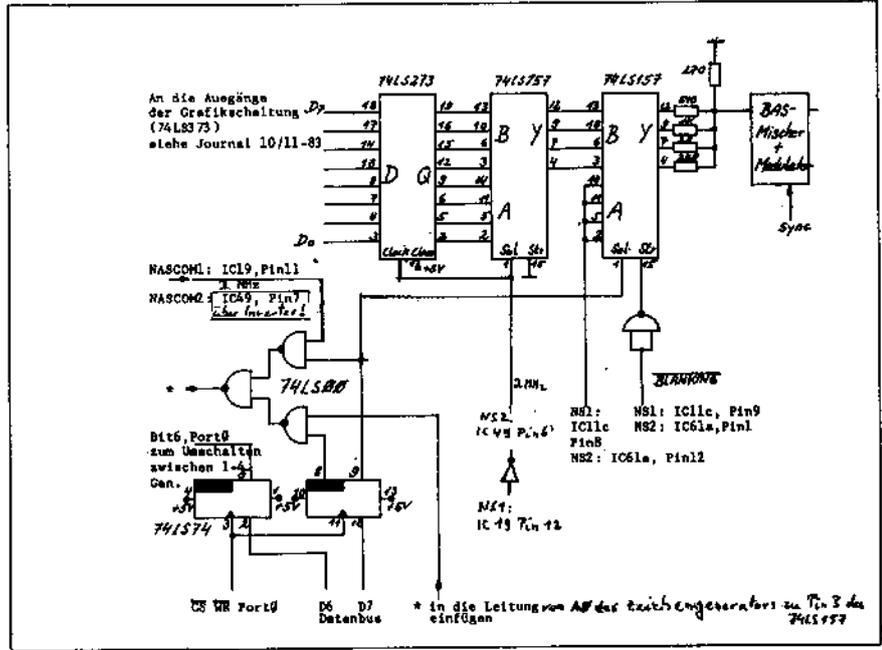
BCA1 0000 0300 DEFN #0
BCA3 DF7B 0310 CHOICE DEFN #72DF ;BLINK
BCA5 0601 0320 LD B,1
BCA7 FE31 0330 CP #1
BCA9 2086 0340 JR Z LOOP1
BCAB FE32 0350 CP #2
BCAD 20F4 0360 JR NZ CHOICE
BCAF 0602 0370 LD B,2
BCB1 210010 0380 LOOP1 LD HL,#1000 ;BUFFER
BCB4 AF 0381 XOR A ;output 0
BCB5 D3F4 0382 OUT (PIOD),A ;Enable 256 Bit
BCB7 0E00 0390 LD C,0 ;COUNTER
BCB9 79 0400 LD A,C
BCBA D3F6 0410 LOOP2 OUT (PIOD),A ;PROM-ADRESSE
BCBC DBF4 0420 IN A,(PIOD) ;DATEN
BCBE E60F 0430 AND #F ;nur 4 Bit
BCD0 77 0440 LD (HL),A
BCD1 BC 0450 INC C
BCD2 79 0460 LD A,C
BCD3 B7 0470 OR A ;schon 256 Adressen?
BCD4 23 0480 INC HL
BCD5 20F3 0490 JR NZ LOOP2
BCD7 3E20 0500 LD A,#20 ;Bit5
BCD9 D3F4 0510 OUT (PIOD),A ;Enable Adresse0
BCDB 10ED 0520 DJNZ LOOP2 ;nocheal falls 512 Bit
0530 ;
BCDD EF 0540 DEFB #EF
BCDE 0000 0541 DEFM #000
BCD0 4349AE47 0550 DEFM 'EINGELESEN BEI #1000
454CA553
454E2042
45492023
31300030
BCE4 0000 0560 DEFM #0
BCE6 DF58 0570 DEFM #5BDF ;HRET
0580 ;

```

Bei Uwe Brockmüller läuft die Floppy inzwischen in Double Density bei 2MHz !!!!

# Nachtraege

Jörg Wittich hat uns eine Korrektur zur Schaltung "Grauwerte" in Heft 12/83 geschickt. Das IC, das im Kästchen zum Nachschalten an den Original-Generator erwähnt wird, soll natürlich ein 74LS244 sein.



Zum Programm TEXTCASS aus Heft 1/84 erreicht uns gerade eine Modifikation von Peter Brendel, die es ermöglicht, auf einfachste Art Texte auf Diskette abzuspeichern. (Sehr gut geeignet für Ihre Beiträge zum 80-Bus Journal).  
 Folgende Zeilen sind zu ändern bzw. einzufügen:  
 490 RST PRS  
 500 DEFM "S TEXT.TXT  
 505 DEFB 0  
 590 JP #A100 ;JP EMO08

**TIP**  
 Wenn Sie BASIC im RAM laufen lassen, müssen Sie auf jeden Fall einen Wert für MEMSIZE eingeben, sonst spielt das BASIC verrückt. Achten Sie darauf, daß EMO08 im gesicherten Speicherbereich bleibt.

## Mitarbeiter

Dieter Oberle  
 Vollmersweiler

Eberhard Horch  
 Hannover

Karl Schulmeister  
 A-Klagenfurt  
 Österreich

Wolfgang Mayer-Gürr  
 Recklinghausen

Michael Bach  
 Gundelfingen

Dieter Metzler  
 Gundelfingen

Andreas Zippel  
 Karlsruhe

Gerhard Klement  
 A-Wien

David Kastrup  
 Aachen

Wolfgang Sauerbrey  
 Aachen

Uwe Brockmüller  
 Oldenburg

Klaus Mombaur  
 Wendelstein

Constantin Olbrich  
 Berlin

Jörg Wittich  
 Donaueschingen

Gerhard Aßmann  
 Herzhausen

Uwe Tepp  
 Hamburg

# nascocom

Hast du NASCOM-C im Haus, so kennst du dich mit Rechnern aus !!

- \* NASCOM-C Bausatz mit 64KB RAM, ANSI-Videoterminal-Logik, 2\*V24 und 1 CENTRONICS-Schnittstelle, MMU, DMA, Controller für 4 Minifloppy-Laufwerke, Netzteil, Tastatur, ein Laufwerk mit 400KB Kapazität, NUCLEOSYS und CP/M komp. Betriebssysteme...DM 2.998,-
- \* NASCOM-C Bausatz mit 64KB RAM ohne Controller....DM 998,-
- \* NASCOM-C Leerplatine mit allen EPROMs, PALs und Handbuchsatz, Schaltplänen, Quellistings, usw.....DM 398,-

Ein NASCOM ohne AVC ist wie ein Winter ohne Schnee !!

- \* NASCOM-AVC Farbgrafik mit 792\*256 Punkte Auflösung, 8 Farben, mit BASIC,-ASSEMBLER-und CP/M-Softwareschnittstellen für die meisten Compiler..DM 848,-

Wer nicht nur BASIC spielen mag, der NASCOM-2 hackt jeden Tag !!

- \* NASCOM-2a, NASCOM-2 mit 8KB CMOS-RAM ,ZEAP-Editor/Assembler und Microsoft-Basic in ROM als Bausatz.....DM 1.098,-

Wer CP/M erleben will, dem ist ein Softcontroller nicht zuviel !!

- \* CLD-Softcontroller mit WD2793-Controller, bringt Ihren NASCOM-2 auf NASCOM-C-Niveau !.....DM 898,-
- \* Softcontrollerbausatz ohne DMA und CTC.....DM 698,-
- \* Softcontroller als Leerplatine + Firmware.....DM 198,-
- \* Minidiskettenlaufwerk TEAC FD55E, 400KB.....DM 740,-
- \* Minidiskettenlaufwerk TEAC FD55F, 800KB.....DM 898,-
- \* Minidiskettenlaufwerk TEAC FD55G, 1600KB.....DM 998,-
- \* CLD-BANKED-Epromkarte für 16 Stück 2708 /16 /32, 2532 sowie 8KB ROMs in vier Banks, Bausatz.....DM 248,-
- \* CLD-Epromkarte als Leerplatine mit Dokumentation..DM 148,-

Wem NASCOM-256KB RAM gehört, den auch kein Speicherfehler stört !!

- \* CLD-256KB-Ramkarte, mit Paritätsprüfung, 64KB RAM, macht CPM+ und MP/M für NASCOM-2 Systeme mit Softcontroller und AVC-Karte möglich.....DM 698,-
- \* CLD-256KB-Ramkarte, Bausatz ohne DMA und Parity-logik mit 64KB RAM.....DM 598,-
- \* CLD-256KB-Ramkarte als Leerplatine mit Dok.....DM 148,-

Die meiste Software dieser Welt man für CP/M 2.2 erhält !!

- \* CP/M 2.2 Betriebssystem mit ADM-31 Terminal emulator fuer AVC-Board, Screen-Editing auf CP/M Kommandoebene, Interface für Centronics-Drucker..DM 498,-
  - \* BIOS-Distribution mit CP/M komp. Betriebssystem...DM 148,-
  - \* CP/M+ Betriebssystem, 256KB-Karte und AVC erforderlich, Komfort und Leistung wie auf Micros bisher nicht bekannt, bis zu 8 mal schnellerer Disk-Zugriff als bei CP/M 2.2, Datum und Uhrzeit, Passwortschutz, MACRO-Assembler, Linker und symb. Debugger im Lieferumfang.....DM 998,-
- Preise inklusiv MwSt., exklusiv Versandkosten

**LAMPSON & ZERBE GMBH**  
Micro-Computer-Vertrieb

Odenwaldstraße 23  
6087 Büttelborn 1  
Telefon (06152) 56730