

nascom journal

Zeitschrift für Anwender des NASCOM 1 oder NASCOM 2

3. Jahrgang · Mai 1982 · Ausgabe 5

Herausgeber:

MK-SYSTEMTECHNIK Michael Klein · Pater-Mayer-Straße 6 · 6728 Germersheim/Rhein
Telefon (0 72 74) 20 93 · Telex 453500 mks d

MK-SYSTEMTECHNIK Thomas Gräfenacker · Kriegsstraße 164 · 7500 Karlsruhe · Telefon (07 24) 2 92 43
MK-SYSTEMTECHNIK Michael von Keltz · Pfaffenberg 4 · 5650 Solingen 1 · Telefon (0 21 22) 4 72 67

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

INHALT

2	NASCOM Journal intern	
3	LESERBRIEFE	
6	Typenrad-Terminal Teil 1	Günter Kreidl
7	I/O Erweiterung AY-3-891Ø	Günter Kreidl
12	Video-Interface	D.Oberle, A.Zippel
15	Lichtgriffel	Günter Böhm
18	Seite(n) für Einsteiger	W.Mayer-Gürr
19	Programmiersprachen	W.Mayer-Gürr
21	Preiswertes Pascal	Michael Bach
23	Mitarbeiter	
24	Datenverwaltung 2	Günter Böhm
3Ø	Mitarbeiter 3/4-82	
31	NASCOMPL/Impressum	
32	MKS-Angebote	

nascocom journal

intern

Liebe Leser,
nachdem diese Rubrik im letzten Heft etwas zu kurz gekommen ist, hier nun wieder etwas ausführlicher.

Zunächst ein Hilferuf! Wo sind Ihre Einsendungen geblieben? Erst hatte ich etwas Sorge, ob sich mit den vorhandenen Beiträgen überhaupt ein vollständiges Heft zusammenstellen ließe. So arbeitete das Team auf Hochtouren. Die abgedruckten Artikel stammen ja nun fast ausschließlich vom "harten Kern". Da ich es nicht über's Herz bringen konnte, einige Artikel für die nächste Ausgabe zurückzuhalten, sieht unser Vorrat nun etwas Mager aus. Also bitte: Trotz des schönen Wetters - setzen Sie sich doch mal an Ihre Maschine und schreiben Sie, was Sie sicher schon lange schreiben wollten.

Zu den Cassettensendungen ein wichtiger Hinweis (die meisten Einsender schicken ja glücklicherweise schon ihre Beiträge auf Cassette): Bitte verpacken Sie die Cassetten ordentlich (am besten in Jiffy-Umschläge). Neulich erhielt ich eine Cassette, die einschließlich Plastikhülle quer durchgeschnitten war. Einziger Kommentar der Post: "Wir bitten, das Vorkommnis zu entschuldigen." Deshalb auch immer Kopien bereithalten!

Michael Bach hat angeregt, die Assemblerlistings in etwas kürzerem Format abzudrucken. Vergleichen Sie nun bitte die Formate beim Lightpen und Typenrad-Terminal. Letzteres ist inklusive Maschinencode abgedruckt und beansprucht sehr viel Platz (deshalb das Querformat). Beim Lightpen tauchen zwei Formate auf: 1. voll formatiertes Listing mit abgesetzten Labels (Maschinencode abgeschnitten) und 2. einfacher Ausdruck des Sourcecodes (Labels eingerückt). Dieser ist natürlich für uns am bequemsten und zudem platzsparend. Bitte entscheiden Sie, welches

Format für Sie am angenehmsten ist. (Bitte keine Klagen, wenn Sie sich nicht melden).

Immer wieder treffen Karten von Lesern ein, die sich am letzten Rundlauf beteiligen wollen. Bitte stellen Sie die Anfragen nun ein, der Rundlauf ging termingerecht heraus und wird nun wohl einige Zeit unterwegs sein (es gab diesmal eine Unmenge von Teilnehmern). Wenn die Cassetten wieder zurück sind werden wir das bekanntgeben.

Beachten Sie bitte die Antwort auf den Leserbrief von Herrn Schneider. Hier wird dargelegt, wie wir in Zukunft das Journal etwas durchlässiger für Benutzer anderer Z80 Systeme gestalten wollen. Das Lightpen-Listing ist da ein bescheidener Anfang (Bitte Meinung dazu; vor allem von Lesern mit anderen Rechnern!). Dies ist im Augenblick für uns das wichtigste Thema und bei weitem noch nicht ausgegoren.

Zur Video-Karte von D. Oberle und A. Zippel (übrigens wie der Typenradanschluß von G. Kréidl schon seit langem von vielen Lesern gewünscht!): Die Folie zum Lay-Out ist für einen Unkostenbeitrag von DM 5.- und Freiumschlag bei der Redaktion erhältlich.

Fehler im Centronics-Interface von J.C.Lotter (Heft 6-82 Seite 10):

74LS125 Pin4 an SFD

TBRE zu Datenbit 6 ist zu unterbrechen (nicht wie angegeben).

Das TOOLKIT scheint mit NASSYS3 nicht zu funktionieren: die automatischen Zellennummern spielen verrückt. Wer weiß Abhilfe??

Hier die versprochenen Pin Anschlüsse der 2114 zum Programmierbaren Charactergenerator von J.C.Lotter (Heft 3/4-82),

Pin 1	A6	Pin 2	A5	Pin 9	GND	Pin10	\overline{WE}
Pin 3	A4	Pin 4	A3	Pin 11 bis 14	DATEN		
Pin 5	A0	Pin 6	A1	Pin 15	A9	Pin16	A8
Pin 7	A2	Pin 8	\overline{CS}	Pin 17	A7	Pin18	+5V

Last not least - Wir haben ihn: den Gewinner des Grafik Preisausschreibens. Michael Bach mit seinem Doppelwurm (bzw. der Klötzchengrafik in Masch.Code) machte das Rennen. Trost für die anderen Teilnehmer: es war ein harter Kampf in der Redaktion und bei den Lesern.

Und nun viel Spaß mit diesem "Hardware-Heft"

Ihr Günter Böhm

LESERBRIEFE

Mir ist besonders positiv aufgefallen, daß seit einigen Ausgaben die Hex-Programme mit Prüfsumme abgedruckt werden.

Nachdem ich das Programm von Herrn Mombaur eingegeben hatte und zu Beginn erfreut war, wie es lief, mußte ich bereits zur ersten Spielrunde feststellen, daß dies ein Trugschluß war. (Wo liegt der Fehler? Red.)

Trauern Sie nicht um den Strichcode. Ich denke, wenn die Lesestifte erst einmal billiger geworden sind und sich eine einheitliche Form durchgesetzt hat, kommt er wieder.

Th. E. Schreiner, 3000 Hannover

Wenig Erklärungen gibt es über den Programmaufbau von 8 K-Microsoft-BASIC und NASPEN. Nützliche Unterprogramme sollten als Assemblerlistings abgedruckt und erklärt werden.

Wünschenswert wäre auch TOOLKIT, DEBUG und NASSYS 3.

Heinrich Broja, Flensburg

Aus irgend einem Grunde läuft die NASSYS 3 Modifikation auf Seite 11 Heft 11/81 nicht. (Sie bezog sich auf NASSYS 1; siehe Heft 1/82 Red.)

Wer kann den Verify-Befehl so verbessern, daß er auch wirklich "verifiziert" und nicht nur Checksummentest macht (ähnlich dem Read-Befehl)?

Wolfgang Rudolph, 631 Grünberg

Beim Eprommer in Heft 11/72-81 muß angegeben werden, mit welchem Clock das System läuft wegen der 50 ms Programmierzeit. (Es läuft mit 2 MHz Red.)

Manfred Stoll, 2000 Hamburg

Beiträge wie Rechnerprogramm, CONMOD, Formattierprogramm (leider T4) ((Inzwischen NASSYS Red.)) finde ich sehr gut. Besonders auch das Eprommer-Programm, da auch eine vollständige Schaltung dazugehörte. BASIC kommt etwas zu kurz. In jedem Heft eine andere Sortieroutine oder ein Spiel mit seitenlangem Text entspricht nur den bescheidensten Möglichkeiten, die BASIC bietet. Eine Schaltung für PIO-Anpassung wäre vollständiger und zu realisieren, wenn auch die PIO-Bausteine mit eingezeichnet wären. DOKE-Programm war sehr gut. Bin an hochauflösender Grafik sehr interessiert.

Hans Schneider, 7901 Dornstadt

Frage: Ich besitze 8-K-Tape-BASIC. Bei Fehlern im Programm spielt der Computer verrückt. Er gibt beim Listing-Befehl richtig eingegebene Zeilennummern als zufällige Kommazahlen aus. Z.B. 100= 98.254. Da ich einen Schreibschutz für das BASIC-Programm besitze, kann durch einen Kaltstart der Fehler behoben werden. Wie kommt es zu diesem Fehler?

Jochen Muenster, 4438 Heek

Herr Böhm sollte noch eine detailliertere Aussage machen zu dem Motor/Getriebeatz von Fa. Valvo (Schrittmotor?). Mein Vorschlag: bei Speichererweiterungen am Mikrocomputer sind meistens Serienfertigungen nötig (C-MOS RAMs für Elektorplatine). Dafür sollte der Plotter auch als Bohr/Fräßgerät einsetzbar sein.

Verbesserungsvorschlag: In Zukunft nie wieder Barcode.

Gerhard Scholz, 3387 Vienenburg

(Anm. Die Valvo-Synchronmotoren - keine Schrittmotoren aus Preisgründen - waren vom Typ 111 31 431: 48V/3.5 Watt. Die Getriebe Bestellnr. 266 8005. Ich würde aber vorschlagen, die erwartete Leistung bzw. Geschwindigkeit individuell zu durchdenken und ein entsprechendes Angebot von Valvo einzuholen. Meine verwendeten Motoren sind z.B. meines Wissens nicht mehr lieferbar.

Anschrift: Valvo, Postfach 106323, 2000 Hamburg 1, Tel. 040 3296-605. Herr Dreger wird Sie gut beraten. G.B.)

Bitte gestatten Sie mir, auch ein paar Worte zum NASCOM-JOURNAL zu 'verlieren'. Zweckmäßigerweise schön nach Punkten unterteilt. Falls Sie antworten wollen, braucht nicht zitiert werden. Leichter lesbar ist es dann auch.

1. Mein Computer

ist kein NASCOM, sondern der LUXOR ABC-80. Da in beiden Geräten ja doch das gleiche Herz schlägt, bietet mir das NASCOM-JOURNAL eine Menge guter Informationen. CHIP oder MC kommen da nicht mit!

2. Sprache

Mit der Maschinensprache (auch der Assemblersprache) stehe ich leider immer noch auf Kriegsfuß. Obwohl langsamer und Platzraubender, ziehe ich das ganz normale BASIC (ohne PEEK und POKE) vor. Es gibt auch so noch genug Anpassungsschwierigkeiten (d.h.: einfach abschreiben geht praktisch niemals).

3. Programme

In der Dezember-Ausgabe von NASCOM-JOURNAL waren viele brauchbare Programme zu finden. Derartiges noch öfter zu finden, würde mich sehr freuen. Das Programm 'Seeschlacht' in der Ausgabe 1/82 ist gespickt mit POKE und PEEK. Und dazu noch mit dem mir gänzlich unbekanntem DOKE und DEEK (wohl nahe Verwandte von POKE und PEEK?!).

Und wie funktionieren die verschiedenen SCREEN-Befehle (mal 3,4 mal 1,14 mal 29 usw.)? Wird da Zeilenweise gelöscht oder ist das eine Cursorbewegung? Wahrscheinlich letzteres! Wenn ich dann noch den Nachsatz (für NASCOM 3) richtig interpretiert habe, dann wird damit überwiegend der Bereich des Bildschirmspeichers direkt beeinflusst bzw. abgefragt. (Meine Befehle dazu SETDOT, DOT, CLRDOT = setzen, abfragen, löschen.) Um alleine diese Grafik richtig zu übertragen, braucht man nicht nur fast einen 'Dr.' vor dem Namen, sondern auch noch sehr viel Zeit und vor allem eine Übersicht der Adressen. Oder sind es beim NASCOM zufällig auch 80 Spalten und 72 Zeilen? Das wäre zu schön um wahr zu sein. Ein Trost ist es, daß mir in diesem Programm sowieso zuviel geschossen wird. Aber man hätte doch gerne gewußt, wie es funktioniert.

4. Strichcode-Tod

Es ist zum weinen - wirklich!

Wenn es nicht gleich so läuft, wie geschmiert, dann braucht man doch nicht gleich aufzugeben?! Sicher, einem Listing kann man gleich den Wert des Programmes entnehmen (meistens jedenfalls!). Auch kann man fremde Sprach-elemente gleich übersetzen bzw. beim Eintippen berücksichtigen. Das geht bei einem Strichcode-Programm leider nicht, Strichcode-Programme müssen darum entweder sehr gerätespezifisch oder in einem universellem BASIC erstellt werden. Und genau da wird auch für Sie der Pferdefuß an der eigentlich recht guten Sache versteckt gewesen sein. Eigentlich müßte ich Ihnen für Ihre Entscheidung dankbar sein. Denn Listings kann ich (meistens) übertragen, mit NASCOM-Strichcode-Programmen könnte ich absolut nichts anfangen. Trotzdem stimmt mich der zu frühe Tod des Strichcodes traurig.

5. Die zwei Mitten

Wer wird denn diese Sache so verbissen sehen?

Aber etwas ist an dieser Sache doch dran, gemeint ist die schadhafte Abtrennung bestimmter Artikel. Nehmen wir hier zunächst als Beispiel das Programm 'Seeschlacht' der Ausgabe 1/82. Wenn ich das NASCOM-JOURNAL nicht im Ganzen sondern nach Sachthemen getrennt aufbewahren möchte, käme die 'Seeschlacht' unter 'SPIELE' oder aber unter 'BASIC' zu liegen. Aber oh weh! Die Seite 11 geht dabei verloren! Der Rest von 'Irrgarten' gehört auch wohl noch in die Schublade 'SPIELE', aber schon nicht mehr unter 'BASIC'. Aber ganz sicher passt die Sache mit dem ROM nicht in eine dieser beiden Schubladen. Also heißt es verzichten oder aber gleich zwei Kopien machen. Könnte die Seite 11 nicht mit einem kleineren Spielprogramm in BASIC oder auch mit einer beliebigen vergänglichem Mitteilung gefüllt sein? Dann gäbe es keine Abtrennprobleme.

Schön: Seiten 19-22 = Grafik-Erweiterung (+ Füller)

Schön: Seiten 23+24 = Format für MASSYS

Schön: Seiten 25-30 = Seite(n) für Anfänger (+ gleichartiger Zusatz)

Schön: Seiten 31-33 = Sprachsynthese (Seite 34 Anzeigen)

Schön: Seite 18 = Inhaltsverzeichnis (Seite 17 Anzeige).

Wenn NASCOM-JOURNAL von vorne bis hinten so systematisch aufgebaut wäre, wie die Ausgabe 1/82 in der letzten Hälfte, dann bräuhete kein Heft mit zwei Mitten erfunden werden. Auch NOTZ und MAUL, sowie die 'Hand vor den Augen' wäre wohl überflüssig. Es genügt ja einfaches Abtrennen der guten von den schlechten Artikeln (je nach Geschmack oder Wissensstand). Soviel Tolleranz traue ich jedem NASCOM-JOURNAL-Leser zu. Auch den Redaktionsmitgliedern sollte man Tolleranz abverlangen können. Denn es ist zugegebenerweise doch eine Zumutung, wenn der Leser nicht das ganze 'Werk' 'schlucken' will, sondern hier und da ein endgültiges Urteil (Mappe X oder Y oder Papierkorb) sprechen will und man ihn dabei auch noch helfen soll (durch leichtere Trennbarkeit der Artikel). Den Redaktionsmitgliedern sollte es ein Trost sein, zu wissen, daß die meisten Leser sich freuen würden.

Ich sehe die ganze Sache nicht ganz so verbissen. Vielleicht liegt das daran, daß ich mich auch nach 15 aktiven Monaten Computer-Hobby noch als Anfänger einstuft.

6. Nachfrage

Betrifft 'Programme aus der Luft' von Mary Jo Kostya.

Zuerst eine ganz dumme Frage (wegen der Anrede!): Ist M.J.K. eine 'Sie' oder ein 'Er'?

Wegen des Programmes BAER muß ich einmal nachhaken. Als Beringer der Vogelwarte Helgoland liegen mir einige von Groß-Computern ausgewertete Wiederfundblätter vor. Zum Testen des Programmes BAER habe ich die vorliegenden Daten damit nachgerechnet. Solange ich im östlichen Bereich bleibe, ist alles klar. So wie ich aber in den westlichen Bereich komme, ergeben sich kleinere, aber auch sehr große Abweichungen. Die Schuld daran möchte ich zunächst den kleineren Computern mit ihrem kleineren Programm geben. Sollte wirklich der 'Große Bruder' schuld sein, so käme das einer mittleren Katastrophe gleich (Neuberechnung und Ausdruck einiger Tausend Unterlagen).

Hans-Dieter Schneider D-2943 Esens

Antwort der Redaktion:

Lieber Herr Schneider,

Sie sind nicht der einzige Leser des Journals, der keinen NASCOM besitzt und versucht, unsere Programme an sein System anzupassen. Schon lange spekulieren wir mit dem Gedanken, die Programme für andere Z80 Systeme durchsichtiger und somit leichter anpaßbar zu machen. Ihr umfangreicher Leserbrief gibt uns nun Anlaß, einmal dazu Stellung zu nehmen.

Zu Punkt 3: es schwebt uns vor, in Zukunft Tabellen zu erstellen, die die Unterschiede zwischen den einzelnen Rechnersystemen deutlich machen. Sie sollen helfen (ähnlich wie der Vergleich zwischen NASSYS und T4) verschiedene Routinen zu erkennen und umzusetzen. Dazu wäre es aber notwendig, von Benutzern der unterschiedlichen Rechner die entsprechenden Befehle bzw. Adressen zu erfahren. Machen Sie doch einen Anfang für den ABC-80. Josef Zeller ist gerade dabei, eine Anpassung für den TRS 80 zu schreiben. Wir denken dabei an Ein- und Ausgaberroutinen, Ladeformate, Bildschirmspeicher, spezielle BASIC-Befehle, Monitor-Unterprogramme usw. Demnächst veröffentlichen wir eine Liste mit den benötigten Angaben.

Zu Punkt 4: im Augenblick ist der Strichcode für uns begraben. Falls sich die technische Entwicklung oder ein Sprachenstandard entsprechend ergeben, und der Wunsch unter den Lesern entsteht; es nun doch noch mit dem Strichcode zu versuchen, können wir immer noch unsere Maschinerie von neuem anwerfen.

Zu Punkt 5: Ihr Vorschlag zur Lösung des Zwei-Mitten-Problems klingt gut. Das Problem ist aber leider nicht so leicht zu lösen. Die von Ihnen so positiv angesprochenen Seiten von Heft 1/82 sind ein Beispiel für unsere Bemühungen, die Artikel und Programme nach Themen zu trennen. Leider läßt sich das nicht immer durchführen; da ist noch eine halbe Seite übrig, die wir nicht einfach frei lassen können, und so muß sich entweder ein Artikel finden, der nun absolut nicht in die Schublade paßt, dafür aber gerade diese halbe Seite füllt; oder es schließt sich ein Programm an, das wiederum den vorgesehenen Platz nicht ganz ausfüllt, und schon haben wir eine Kettenreaktion, die das ganze Konzept durcheinanderbringt. Durch minutiöse Planung ließe sich so etwas schon vermeiden,

aber da wir alle nur hobbymäßig am Journal arbeiten, müssen wir die Artikel Stück für Stück vorbereiten, damit die Layoutarbeit nicht auf einmal zusammenkommt. So muß wohl jeder Leser seine Abstriche machen, oder wohl doch hie und da einen "MOTZ" oder "MAUL" aufkleben.

Zu Punkt 6: zunächst M.J.K. ist eine "Sie" (übrigens meines Wissens die einzige

Leserin des Journals). Über das Programm BAER können wir leider keine Auskunft geben. Ich verweise an Frau Kostya (Adresse im entsprechenden Heft) und fordere gleichzeitig die Leser auf, über ihre Erfahrungen zu berichten.

Ich gratuliere zu der gelungenen Gestaltung des Journals. Besonders gefällt mir die redaktionelle Unabhängigkeit im Gegensatz zu zumindest einer anderen an einen Rechner gebundenen Zeitschrift, in der in erster Linie Firmenprodukte verkauft werden.

Ich wünsche mir, daß die private Atmosphäre zwischen dem NASCOM Journal und seinen Lesern erhalten bleibt und hoffe, daß die Redaktion nicht den Spaß an der zeitraubenden Arbeit verliert, besonders da dies keine Reichtümer einbringt.

Reno Hinrichs, 6500 Mainz

Hinweis zum Programm "Türme von Hanoi" von E.Horch (Heft 3/4-82 Seite 49): Das Spiel benutzt die NASCOM2-Grafik. Die PRINT-Anweisungen in Zeile 1300 oder 1330 sind entsprechend zu interpretieren.

Aus ungeklärten Gründen hat der Rechner beim Ausdruck des Listings zu "Doppelwurm" von M. Bach (Heft 3/4-82) einige falsche Bytes ausgespuckt. (Wahrscheinlich hat sich die Nachbarin gerade wieder die Haare geföhnt). Hier also nun die richtigen Bytes:

0E18 20 28 F9 18

0E9B 52

0F18 B6 77 E1 D1

Wir danken Herrn Stehlik für den Hinweis und der Sprachplatte für die Hilfe bei der Fehlersuche.

Typenrad-Terminal Teil 1

von Günter Krelidl

Seit einiger Zeit sind preiswerte Typenrad-Schreibmaschinen auf dem Markt. Eine von ihnen ist die PRAXIS 307/35 von Olivetti, die unter anderem Namen auch vom Versandhaus Quelle für knapp 1100,- DM angeboten wird. Findige und geschäftstüchtige Leute haben natürlich schnell gemerkt, daß diese Maschine zwar keinen Computeranschluß besitzt, daß man einen solchen aber leicht bauen kann, und bieten in Kleinanzeigen "Interfaces mit Normschnittstellen" zu Preisen zwischen ca. 400,- und 1200,- DM an. In MC Nr. 3/82 hat Dr. Ingmar Thilo gezeigt, wie man statt eines solchen Interfaces auch ein Kabel und ein paar Widerstände nehmen kann. Das "Interface" besteht dann in einem Programm, das nach den Prinzipien des "Abhörens" und "Täuschens" aufgebaut ist; der in die Schreibmaschine eingebaute Ein-Chip-Microcomputer wird bei seiner Arbeit belauscht und erhält im geeigneten Augenblick ein Signal, daß ihm einen Tastendruck auf der Schreibmaschine vortäuscht: damit wird die Typenradschreibmaschine zum Schönschreibdrucker für jeden Computer. Andererseits kann auch die Tastatur der Schreibmaschine als Eingabestelle für den Computer benutzt werden; dabei kann der Druckvorgang verhindert werden, indem der interne Microcomputer wieder getäuscht wird: Es wird ihm ein Signal übermittelt, das eine gleichzeitige Eingabe von 8 Tasten vorgaukelt, die er dann nicht akzeptiert. Mit anderen Worten: Mit einem Materialaufwand von ca. 5,- DM (Flachkabel, Steckverbinder und 2 Widerstände) wird aus der Schreibmaschine ein Typenrad-Terminal.

Der Anschluß

Zum Anschluß der Typenradmaschine werden 18 I/O-Leitungen benötigt. Man kann schon mit der Grundversion des NASCOM (ohne weitere PIO) auskommen, wenn man Port 0 mitbenutzt. Dies ist allerdings etwas umständlich, da die 4 freien I/O-Leitungen von Port 0 nicht in der Datenrichtung programmierbar sind. Ich habe die beiden I/O-Ports des PSG AY-3-8910 benutzt. Dazu muß man sich das in diesem Heft beschriebene bidirektionale Interface zum PSG bauen. Außerdem werden

noch zwei Bits von Port B der PIO benutzt. Der Anschluß in der Schreibmaschine wird an den Lötstellen der beiden 13-poligen Flachkabel, die die Tastatur mit der Elektronikplatine verbinden, auf der Tastaturplatine vorgenommen. Es müssen 19 Verbindungen nach außen geführt werden: 8 Eingangsleitungen der Tastatur (MATC 0 ... 7), 8 Ausgangsleitungen der Tastatur (MATR 0 ... 7), eine Leitung für die Shift-Taste und eine für die Keyboard-Selektion (SPECOA) sowie der Masseanschluß (siehe Bild 1 + 2). MATC wird mit Port A des PSG verbunden, MATR mit Port B. SHIFT und SPECOA werden jeweils über einen Widerstand von 1K an Bit 0 und 1 von Port B der PIO gelegt. Die in MC 3/82 angegebenen Pull-Up-Widerstände sind in unserem Fall nicht nötig, da die PSG-Ports über interne Pull-Ups verfügen. Wer aber den Anschluß über PIO-Ports vornimmt, der sollte 1K-Widerstände von MATC 0 ... 7 nach +5V legen. Dieser Anschluß ist aus zwei Gründen nicht mit dem in der MC beschriebenen identisch: Der MC-Redaktion lag kein Schaltbild der Maschine vor, so daß dem Verfasser einige Busleitungen durcheinandergerieten. Das stört aber nicht weiter, wenn man von den geänderten Ansteuercodes absieht. Außerdem fehlt dort der Anschluß zum Keyboard-Selektor, durch den der Computer zwischen den beiden Zeichensätzen (9 Tasten sind 3- bzw. 4-fach belegt!) umschalten kann. Das ist die ganze Hardware! Man kann nun alle Funktionen der Schreibmaschine ansteuern (auch die 14 Funktionstasten!) und alle Tasteneingaben in den Computer übernehmen. Nur den Zeilenabstand muß man vor Hand einstellen.

Die Software

Da die Schreibmaschine erst wenige Tage vor Redaktionsschluß des Journals bei mir ankam, war natürlich keine Zeit mehr, das ausführliche Terminal-Programm zu schreiben, das ich mir wünsche. Deshalb wird in diesem Heft nur ein einfaches Programm wiedergegeben, das die Maschine als Drucker ansteuert. Die dabei verwendeten Unterprogramme und die Code-Tabellen sind schon für das ausführlichere Programm vorbereitet. Das Treiberprogramm erwartet ein Zeichen im Akku, maskiert Bit 7, untersucht, ob es sich um ein Steuerzeichen (Hex 0 - 1F) handelt, von denen es nur NEWLINE (Hex 0D)

akzeptiert, und gibt das Zeichen auf der Schreibmaschine aus. Dann kehrt es mit unveränderten Registern ins Hauptprogramm zurück. Bei jedem NEWLINE wird die Zeilenzahl inkrementiert und mit einer vorgegebenen Zahl verglichen. Bei Erreichen dieser Zahl, wird der Druck unterbrochen (Papierwechsel !) bis von der NASCOM-Tastatur NEWLINE eingegeben wird. Durch Drücken einer beliebigen Taste kann man das Programm zwischen zwei Zeilen anhalten und ebenfalls mit NEWLINE fortfahren. Nach dem Laden wird zunächst INIT aufgerufen mit E F35 XX. XX ist die maximale Zeilenzahl, die an einem Stück gedruckt werden soll. Dann ist das Programm einsatzbereit und kann auch mit dem U-Befehl der Bildschirm-Ausgabe zugeschaltet werden. Die Schreibmaschine muß während des Computerbetriebs immer in der Stellung KB II stehen und die SHIFT-LOCK-Funktion darf nicht eingeschaltet sein.

Anwendungen

Die Schreibmaschine ist vorwiegend für Textverarbeitung geeignet. Als Schnelldrucker eignet sie sich wirklich nicht; wenn beide Tastaturselektionen verwendet werden, beträgt die Schreibgeschwindigkeit etwa 5 Zeichen/Sekunde. Ein weiterer Schwachpunkt der ansonsten guten Maschine ist die Papierführung: schlichtweg eine Krankheit!

I/O Erweiterung mit AY-3-8910 von Günter Kreidl

Der PROGRAMMABLE SOUND GENERATOR AY-3-8910 wurde von Herrn Böhm bereits im NASCOM-Journal vorgestellt (6 + 7²81). Über seine Piepserqualitäten hinaus verfügt er noch über zwei 8-Bit-I/O-Ports. Um sie in beiden Richtungen benutzen zu können, muß ein bidirektionales Interface gebaut werden. Dies ist möglich über Port A der PIO im MODE 2. Da der PSG keine Quittungssignale erzeugt, die für den bidirektionalen PIO-Modus nötig sind, müssen diese Signale extern mit ein paar Gattern und Monoflops erzeugt werden. Eine entsprechende Schaltung zeigt Bild 3.

Die zugehörige Software soll hier noch kurz

besprochen werden. READ, WRITE und SETADR sind die Ansteuerrouinen für den PSG. PINIT initialisiert die PIO und die beiden Leitungen von Port O. Mit dem PSG-Testprogramm sind alle Register des PSG von der Tastatur ansteuerbar. Nach dem Programmaufruf kann man bis zu 10 3-stellige (!) Hexzahlen eingeben (bis NEWLINE). Die erste Stelle gibt die Adresse des jeweiligen PSG-Registers an (0 - F), die beiden nächsten Stellen sind der Wert, der in das entsprechende Register geladen wird. Das Ergebnis sollte man sich anhören! PSGTST ist ein Programm für diejenigen die die Schaltung nachbauen und vor (!) dem Einsetzen des PSG testen wollen. Nach dem Aufruf wird zunächst eine ein- (Adresse) oder zweistellige Hexzahl eingegeben, dann NEWLINE, dann A (Adressieren) oder W (Schreiben) oder R (Lesen). Das Programm gerät dann in eine endlose Schleife, in der es ständig die zum Adressieren, Lesen oder Schreiben von PSG-Registern notwendigen Signale erzeugt. Folgende Signale müssen anliegen (AL = Active Low, AH = Active High):

READ: BDIR = LOW, BC1 = 1000 ns AH, BSTB = 500 ns AL; WRITE: BDIR = 500ns AH, BC1 = LOW, ASTB = 1000 ns AL; ADRESS³ BDIR = 500 ns AH, BC1 = 500 ns AH, ASTB = 1000 ns. Am Monoflop 3 sollte ein Impuls von ca. 300 ns erscheinen. Die Zeiten sind nicht kritisch, nur die Verhältnisse sollten stimmen.

IC's: 2 x LS 08, 1 x LS 00, 1 x LS 32, 2 x LS 123 R1, R2, R4 = 10k, R3 = 5k, C1, C2, C3 = 100pf, C4 = 200pf.

BIETE Lohn- und Einkommensteuer-Berechnungsprogramm auf Cassette. Tausche gegen NAS-DIS, NAS-DEBUG, ZEAP oder andere Programme in EPROM.

K.Körner, [REDACTED]

Tel. [REDACTED]

Suche INMC Newsletter 7 (1981) zum Kopieren. Suche ständig interessante Software zum Tausch gegen NASPEN, DEBUG, CLD-BASIC, TOOLKIT, Schach, NASSYS 3 u. v.a.

Johannes C. Lotter, [REDACTED]

Tel. [REDACTED]

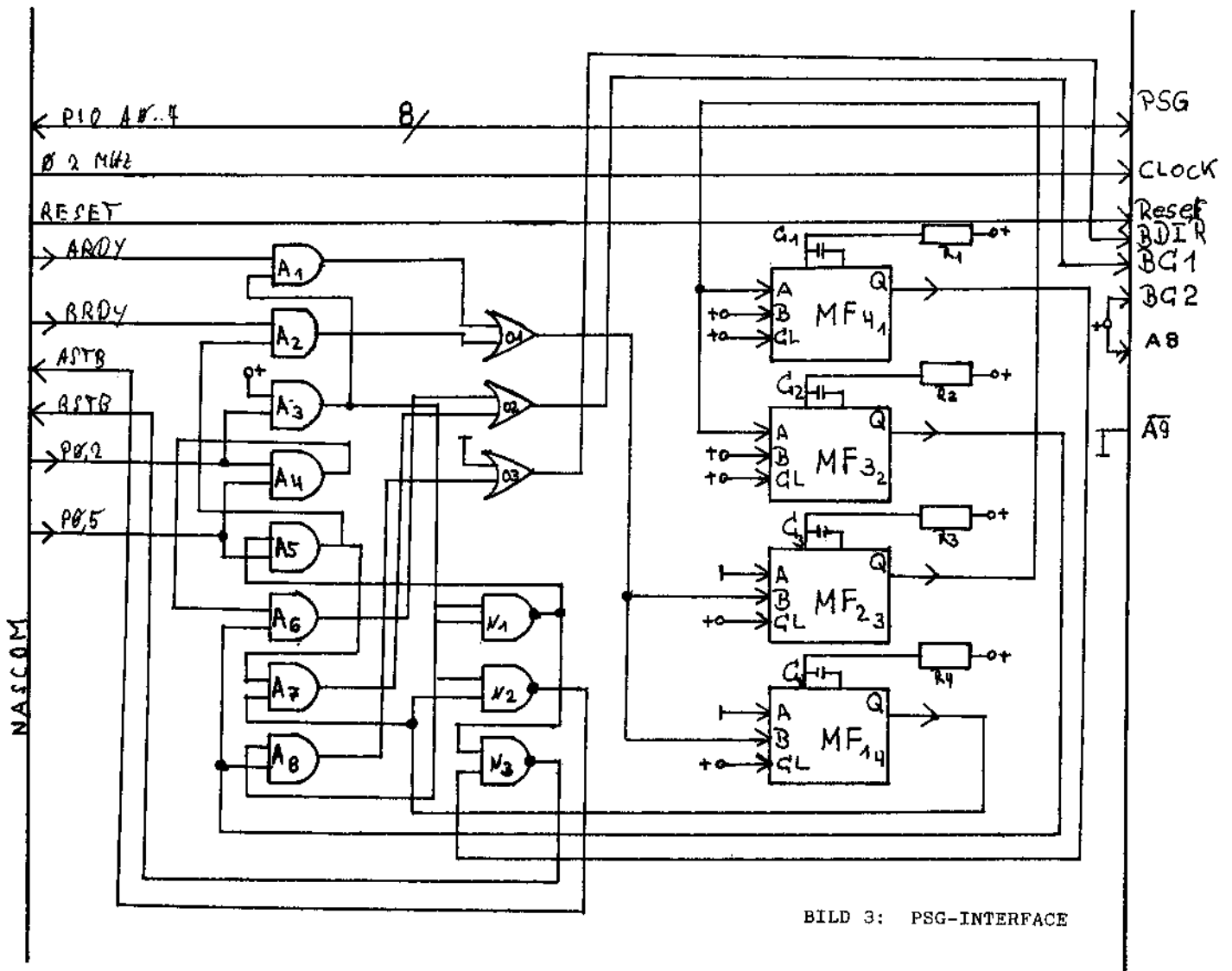


BILD 3: PSG-INTERFACE

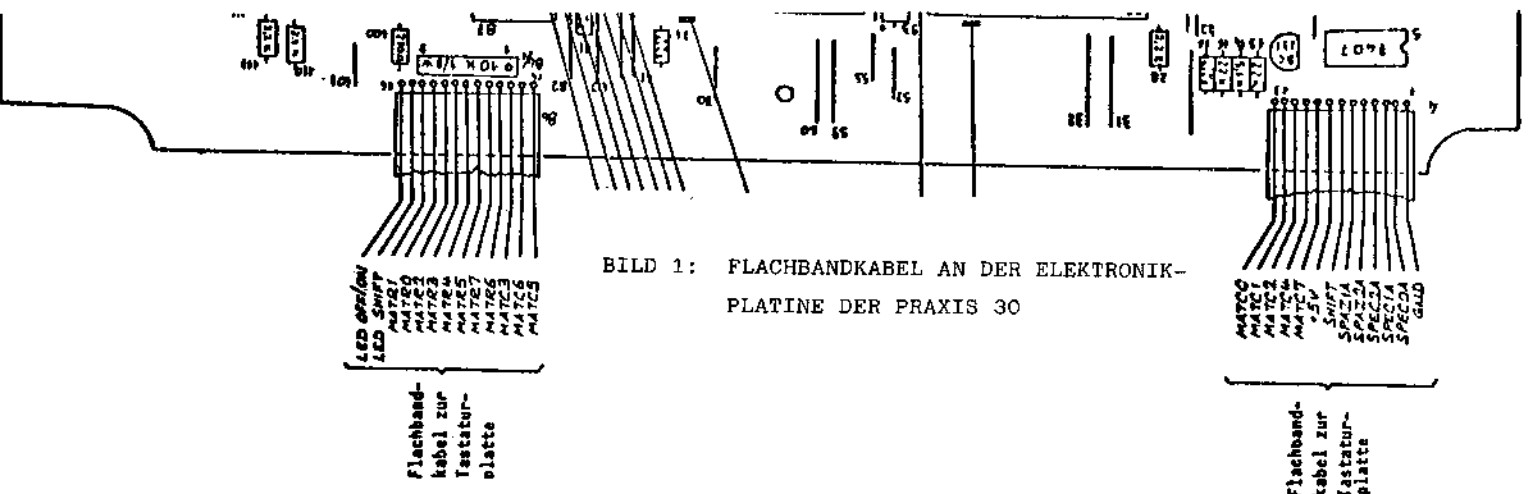
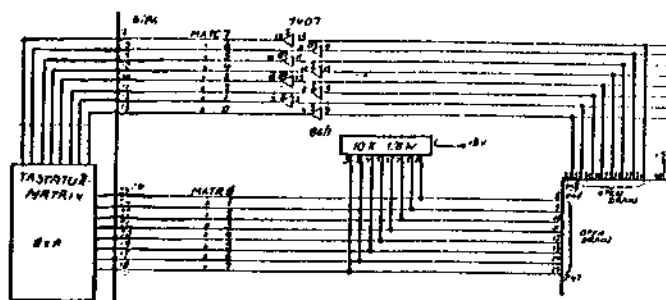


BILD 1: FLACHBANDKABEL AN DER ELEKTRONIK- PLATINE DER PRAXIS 30

BILD 2: PRAXIS 30 TASTATURANSTEUERUNG DURCH F8-PROZESSOR



ODCA	CD370E	1200	CALL	SETADR	OE47	3E20	1800	READ	LD	A, READEN
ODCD	CD3F0E	1210	CALL	WRITE	OE49	D300	1810	OUT	(0), A	
ODDO	21FF0F	1220	LD	HL, EFFF	OE4B	DB04	1820	IN	A, (4)	
ODD3	CD370E	1230	CALL	SETADR	OE4D	C9	1830	RET		
ODD6	CD3F0E	1240	CALL	WRITE			1840	Code-Tabellen		
ODD9	OE0A	1250	LD	C, DELAY4			1850	1) Steuerworte		
ODDB	78	1260	LD	A, B	OE4E	FE	1860	TABSTW	DEFB	EF, EFD, EFB, EF7
ODDC	E602	1270	AND	2	OE52	EF	1870	DEFB	EF, EDF, EBF, EF7	
ODDE	FD8E01	1280	CP	(IY+1)			1880	2) Druckzeichen Kurz-Code		
ODE1	2806	1290	JR	Z, COUNT2			1890	Reihenfolge entspricht ASCII		
ODE3	AF	1300	XOR	A			1900	Codierung: BIT 7 LOW = SHIFT		
ODE4	060C	1310	LD	B, DELAY3			1910	BIT 3 LOW = KB1		
ODE6	FF	1320	RST	E38			1920	BIT 4-6 = POSITION EINGANGS-		
ODE7	10FD	1330	DJNZ	DL3			1930	STEUERWORT IN TABSTW		
ODE9	78	1340	COUNT2	A, B			1940	BIT 0-2 = POSITION AUSGANGS-		
ODEA	E602	1350	AND	2			1950	STEUERWORT IN TABSTW		
ODEC	FD7701	1360	LD	(IY+1), A	OE56	E7	1960	TABDR	DEFB	EF7, E22, E20, E38
ODEF	CD30D	1370	CLOOP	CALL	WAIT1	OE5A	48	DEFB	E48, E40, E60, E12, E50, E63	
ODF2	260F	1380	LD	H, EF	OE60	4E	1980	DEFB	E4E, E36, E32, E2, E36	
ODF4	6B	1390	LD	L, E	OE66	BE	1990	DEFB	E3E, E90, EAO, EBO, ECO, EEO	
ODF5	CD370E	1400	CALL	SETADR	OE6C	DO	2000	DEFB	E30, E32, E32	
ODF8	CD3F0E	1410	CALL	WRITE	OE72	92	2010	DEFB	E46, E14, E62, E37, E34, E33	
ODFB	CD3F0E	1420	CALL	WAIT2	OE76	46	2020	DEFB	E44, E57, E61, E45, E51, E41	
ODFE	21FF0F	1430	LD	HL, EFFF	OE82	31	2040	DEFB	E31, E42, E52, E35	
OE01	CD370E	1440	CALL	SETADR	OE86	25	2050	DEFB	E25, E13, E43, E24, E53, E55	
OE04	CD3F0E	1450	CALL	WRITE	OE8C	47	2060	DEFB	E47, E23, E27, E17, E54, E11	
OE07	0D	1460	DEC	C	OE92	21	2070	DEFB	E21, E15, E68, E28, E18, E83	
OE08	20E5	1470	JR	NZ	CLOOP	OE96	56	DEFB	E56, E94, E2, E7, E84, E83	
OE0A	213F07	1480	LD	HL, E73F	OE9C	C4	2090	DEFB	E34, E32, E2, E5, E31, E31	
OE0D	CD370E	1490	CALL	SETADR	OEAE	B1	2100	DEFB	E31, E32, E2, E5	
OE10	CD3F0E	1500	CALL	WRITE	OEAF	A5	2110	DEFB	E45, E93, E33, E44, E33, E35	
OE13	3EFF	1510	LD	A, EFFF	OEAC	C7	2120	DEFB	E37, E43, E47, E97, E34, E91	
OE15	D307	1520	OUT	(7), A	OEAE	A1	2130	DEFB	E41, E95, E46, E2A;	
OE17	D307	1530	OUT	(7), A			2140	Funktions-Code-Tabelle		
OE19	C9	1540	RET				2150	2-Byte-Code 1. Byte =		
OE1A	3EFF	1550	PI-INITIALISIERUNG				2160	Funktionsklasse		
OE1C	D307	1560	PINIT	LD	A, EFFF		2170	2. Byte = Ausgabecode		
OE1E	D307	1570	OUT	(7), A; PORT B MODE 3			2180	RES	EQU	0
OE20	3E03	1580	OUT	(7), A; ALLE BITS INPUT			2190	DIR	EQU	E10
OE22	D307	1590	LD	A, 3	0000		2200	SPEC1	EQU	1
OE24	D306	1600	OUT	(7), A; KEIN INTERRUPT PORT B	0001		2210	SPEC2	EQU	2
OE26	3E8F	1610	OUT	(6), A; KEIN INTERRUPT PORT A	0002		2220	SPEC3	EQU	3
OE28	D306	1620	LD	A, E8F	0003		2230	SPEC4	EQU	4
OE2A	213F07	1630	OUT	(6), A; PORT A MODE 2	0004		2240	SPEC5	EQU	5
OE2D	CD370E	1640	LD	HL, E73F	0005		2250	SPEC6	EQU	6
OE30	CD3F0E	1650	CALL	SETADR	0006		2260	SPEC7	EQU	7
OE33	AF	1660	CALL	WRITE	0007		2270	SPEC8	EQU	8
OE34	D300	1670	XOR	A	0008		2280	SPEC9	EQU	9
OE36	C9	1680	OUT	(0), A	0009		2290	EINF	EQU	0
OE37	3E24	1690	RET		000A		2300	SPECA	EQU	EA
OE39	D300	1700	SETADR	A, ADREN	000B		2310	SPECB	EQU	EB
OE3B	7C	1710	OUT	(0), A	000C		2320	SPECC	EQU	EC
OE3C	D304	1720	LD	A, H	000D		2330	SPECD	EQU	ED
OE3E	C9	1730	OUT	(4), A	000E		2340	SPECE	EQU	EE
OE3F	3E04	1740	RET		OE66	11	2350	TABF	DEFB	DIR+SPEC1, EF1
OE41	D300	1750	LD	A, WRITEN	OE68	12	2360	DEFB	DIR+SPEC2, EAE	
OE43	7D	1760	OUT	(0), A	OEBA	12	2370	DEFB	DIR+SPEC2, E96	
OE44	D304	1770	LD	A, L	OEBC	12	2380	DEFB	DIR+SPEC2, E16	
OE46	C9	1780	OUT	(4), A	OE46	10	2390	DEFB	DIR+EINF, E3E	
OE46	C9	1790	RET							

OE00 10	DEFB DIR+EINF,£2E	OD1B C9	RET	
OE02 10	DEFB DIR+EINF,£9E	OD1C 3E04	LD	A,WRITEN
OE04 10	DEFB DIR+EINF,£1E	OD1E D300	OUT	(O),A
OE06 0000	DEFW RES	OD20 7D	LD	A,L
OE08 13	DEFB DIR+SPEC3,£FO	OD21 D304	OUT	(4),A
OE0A 00	DEFB RES	OD23 C9	RET	
OE0B 14	DEFB DIR+SPEC4,£F2	OD24 3E20	LD	A,READEN
OE0D 0000	DEFW RES	OD26 D300	OUT	(O),A
OE0F 15	DEFB DIR+SPEC5,£E6	OD28 DB04	IN	A,(4)
OE1 16	DEFB DIR+SPEC6,£FF;DUMMY	OD2A C9	RET	
OE3 17	DEFB DIR+SPEC7,£85	OD2B CDO00D	CALL	PINIT;PSG-TESTPROGRAMM
OE5 18	DEFB DIR+SPEC8,£81	OD2E DF	RST	£18
OE7 19	DEFB DIR+SPEC9,£E5	OD30 DF	DEFB	£63;INLIN
OE9 10	DEFW RES	OD31 79	RST	£18
OE1B 0000	DEFW RES	OD32 3004	DEFB	£79;RLIN
OE1D 0000	DEFW RES	OD34 DF	JR	NC,PSGLP
OE1F 0000	DEFW RES	OD35 6B	RST	£18
OE3 1A	DEFB DIR+SPECA,£F6	OD36 18F6	DEFB	£6B;ERRM
OE5 0000	DEFW RES	OD38 DD210B0C	JR	MAINLP
OE7 1E	DEFB DIR+SPECB,£86	OD3C DD4600	LD	IX,£COB
OE9 1C	DEFB DIR+SPECC,£82	OD3F DD23	LD	B,(IX)
OE1D 1D	DEFW RES	OD41 DD6E00	INC	IX
OE1F 1E	DEFB DIR+SPECD,£F3	OD44 DD23	LD	L,(IX)
OE3 1E	DEFB DIR+SPECE,£F5	OD46 DD6600	INC	IX
OE40	DEFB DIR+SPECE,£F7	OD49 DD23	LD	H,(IX)
OF35 CD1A0E	WORKSP	OD4B CD140D	INC	IX
OF38 21000D	INIT	OD4E CD1C0D	CALL	SETADR
OF3E FD£1F50E	CALL	OD51 10EE	CALL	WRITE
OF42 3A0E0C	LD	OD53 18D9	DJNZ	PSGLP1
OF45 FD7703	LD	OD55 CDO00D	JR	MAINLP
OF48 AF	XOR	OD58 DF	CALL	PINIT;SIGNALTEST VOR
OF49 FD7702	LD	OD59 63	RST	£18;EINSETZEN DES PSG
OF4C DF	LD	OD5A DF	DEFB	£63;MIT OSZILLOSKOP
OF4D 5B	RST	OD5B 79	RST	£18
	DEFB	OD5C DF	DEFB	£79
		OD5D 7B	RST	£18
		OD5E F7	DEFB	£7B
		OD5F DF	RST	£30
		OD60 60	RST	£18
		OD61 FE57	DEFB	£60
		OD63 2811	CP	"W
		OD65 FE52	JR	Z,WLOOP
		OD67 2816	CP	"R
		OD69 FE41	JR	Z,RLOOP
		OD6B 20E8	CP	"A
		OD6D 3E24	JR	NZ,PSGTST
		OD6F D300	LD	A,ADREN
		OD71 7D	OUT	(O),A
		OD72 D304	LD	A,L
		OD74 18FC	OUT	(4),A
		OD76 3E04	JR	ALP
		OD78 D300	LD	A,WRITEN
		OD7A 7D	OUT	(O),A
		OD7B D304	LD	A,L
		OD7D 18FC	OUT	(4),A
		OD7F 3E20	JR	WLP
		OD81 D300	LD	A,READEN
		OD83 DB04	OUT	(O),A
		OD85 18FC	IN	A,(4)
			JR	ILP

0010	;PSG-PROGRAMME 25.4.82			
0020	INAKT EQU 0;PSG-CONTROL PORT 0			
0030	ADREN EQU £24			
0040	READEN EQU £20			
0050	WRITEN EQU £04			
0060	ORG £D00;1) PIO-INITIALISIERUNG			
0070	PINIT XOR A			
0080	OUT (O),A;PSG DISABLE			
0090	LD A,£FF			
0100	OUT (7),A;PORT B MODE 3			
0110	OUT (7),A;ALLE BITS INPUT			
0120	LD A,3			
0130	OUT (7),A;KEIN INTERRUPT PORT B			
0140	OUT (6),A;KEIN INTERRUPT PORT A			
0150	LD A,£8F			
0160	OUT (6),A;PORT A MODE 2			
0170	RET			
0180	SETADR LD A,ADREN			
0190	OUT (O),A			
0200	LD A,H			
0210	OUT (4),A			

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 seinen 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 Cursosteuerung, das Bildschirmformat und die Behandlung von CTRL und ESC Charaktern nur von der Programmierung abhängig sind.

Ebenso ist es per Software möglich die Anpassung an verschiedene Videomonitor 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 x800h 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 21h 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 IC7404, IC163 und ICS174 (Latch) erzeugt und über Logikgatter entsprechend verteilt. (ICS174 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 Bildschirmzeilen (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 Ramspeicher 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(8) --> 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 Freiumschlag 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 am NASBUS Pin 45 und IC7485 Pin 9 muss durch einen dünnen Wrap-Draht nachträglich im Layout noch eingefügt werden.

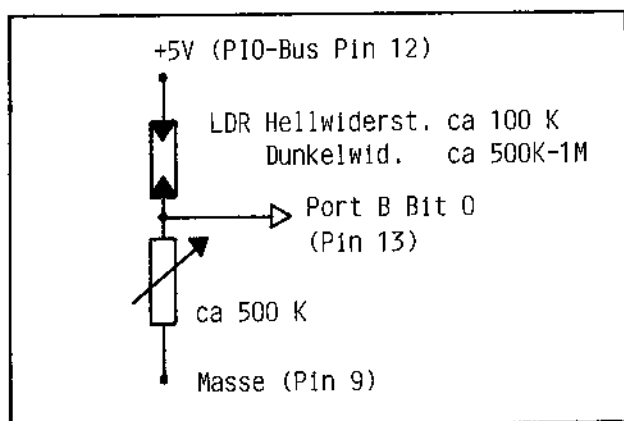
LITERATURANGABEN

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

Der billigste Lichtgriffel der Welt

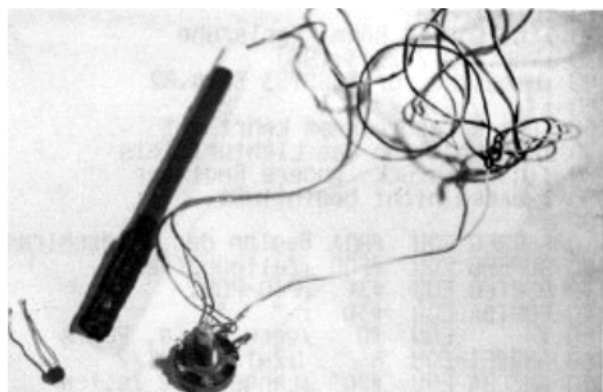
von Günter Böhm

Die Überschrift übertreibt sicher nicht. Die Bauteile zum Lichtgriffel finden sich sicher in jeder Bastelkiste. Benötigt werden nur ein lichtempfindlicher Widerstand (LDR), ein Potentiometer (oder nach Ausmessen auch Festwiderstände) und das "Gehäuse" eines ausgedienten Filzschreibers.



Die Schaltung ist so primitiv, daß sich Josef Zeller strikt weigert, sie ins reine zu zeichnen. Aber gerade der geringe Aufwand fordert dazu heraus, mit solch einem Instrument wie dem Lichtgriffel einmal zu experimentieren. Über Aufbau und "Abgleich" braucht wohl beim Betrachten der Schaltung nichts mehr gesagt werden; wichtiger als die Einstellung des Potis scheint mir sowieso der Abgleich des Bildschirms auf Kontrast und Helligkeit. Auch sollte man zu helles Streulicht im Raum vermeiden.

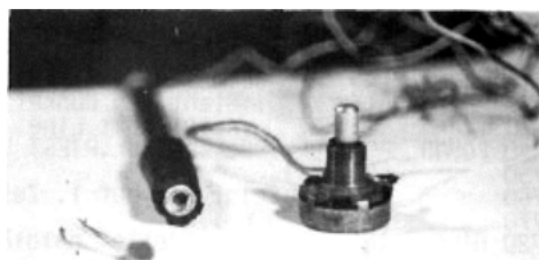
Das erste Unterprogramm tastet den Bildschirm ab, um die Position des Lichtgriffels auf dem Schirm zu finden. Um nicht laufend jede Adresse abzuklappern, werden jeweils zwei Zeilen hell und dann dunkel gezeichnet. Wird der Pen in einer Zeile entdeckt, wird seine Position durch laufendes Halbieren der "Zelle" eingeengt und schließlich in HL übergeben. Diese Methode spart Zeit, zeigt aber doch die gewaltige Unterlegenheit gegenüber einem "professionellen" Lightpen, der z.B. an eine Grafikkarte angeschlossen ist. Der Schirm wird laufend mit "Sternchen"



überzogen, und man bemerkt, wie der Griffel häufig verzweifelt versucht, die richtige Position zu finden, zumal, wenn man ihn nicht exakt auf einer Zeile positioniert hat. Sie können Ihre Erfahrungen damit selbst mit den beiden Hauptprogrammen KEYB (Start #D24) und DRAW (Start #D2D) machen. Sehr viel verlässlicher (und wahrscheinlich den Einsatz in manchen Programmen wert) ist die andere Subroutine HLPEN, die nur vorher definierte Bildschirmadressen untersucht. Die Zuverlässigkeit läßt sich durch die Anzahl der Prüfschleifen steigern (Zeile 1590); bei mir lief das Test-Menu-Programm allerdings fehlerlos mit drei Wiederholungen, wie im Listing angegeben.

Das Assemblerlisting ist für die Anwendung auf anderen Systemen sehr offen gehalten. Im Hexdump wird Bit 0 von Port B zum Anschluß des Griffels verwendet. (Siehe Schaltbild). Die Unterprogramme sind im Prinzip verschieblich, benutzen allerdings Unterprogrammaufrufe, die man beim Verschieben beachten muß. Ebenso die Sprünge in Zeile 1140 und 2260.

Falls Sie den Lichtgriffel in einem Ihrer Programme einsetzen, lassen Sie uns bitte davon hören. Wer weiß, welche Einsatzmöglichkeiten es da geben könnte. (Günter Kreidl spielt mit dem Gedanken, Noten für den Soundgenerator vom Bildschirm abzulesen).



```

0010 ;LIGHT-PEN
0020 ;(C) GÜNTER Böhm, Karlsruhe
0030 ;Ver.1.0 / 23.3.1980
0040 ;umgeschr.für MASSYS3 31.4.82
0050 ;
0060 ;Das Unterprogramm kehrt mit
0070 ;der Position des Lichtgriffels
0080 ;in HL zurück. Andere Register
0090 ;werden nicht beeinflusst.
0100 ;
0110 SCRBEQ EQU #80A Beginn des Bildschirmsp
0120 BUFBEQ EQU #FOO ;Zeilenpuffer
0130 PORTCO EQU #3F ;PIO-PORT
0140 PORTDA EQU #3D ;" "
0150 X EQU #0 ;verw.Bit d. Ports
0160 ANZZEI EQU 8 ;Zeilenzahl/2
0170 ANZCHA EQU #70 ;Länge v. 2 Zeilen
0180 ABST EQU #80 ;vert.Abstand zw.3 Zeilen
0190 LIGHT EQU #2A ;Char.zum Hellzeichnen
0200 ORG #C80
0210 LTPEN PUSH AF
0220 PUSH BC
0230 PUSH DE
0240 START1 LD A,#4F
0250 OUT (PORTCO),A ; Input Mode
0260 START2 LD B,ANZZEI
0270 LD HL,SCRBEQ
0280 BUFFER LD DE,BUFBEQ
0290 PUSH HL
0300 PUSH BC
0310 PUSH HL
0320 LD BC,ANZCHA
0330 LDIR ;2 Zeilen in Puffer
0340 POP HL
0350 LD B,ANZCHA
0360 CALL PTEST
0370 RSTORE PUSH HL ;Bildsch.restaur.,wenn
0380 PUSH HL ;Pen nicht gefunden
0390 LD BC,ANZCHA
0400 LD HL,BUFBEQ
0410 POP DE
0420 LDIR
0430 POP HL ;Zeilenanfang vor RSTORE
0440 ZEILE LD BC,ABST
0450 ADD HL,BC
0460 POP BC
0470 POP DE
0480 DJNZ BUFFER ;Test auf Bildsch.Ende
0490 JR START2 ;Schleife bis Pen gefund
0500 ;
0510 PTEST PUSH BC
0520 PUSH HL
0530 HELL LD (HL),LIGHT
0531 INC HL
0532 DJNZ HELL
0540 PEN1 XOR A
0550 OUT (PORTDA),A ;Reset Port
0560 CALL DELAY
0570 IN A,(PORTDA)
0580 BIT X,A ;"Sieht Pen Licht?"
0590 POP HL
0600 POP BC
0610 RET Z ;nein: Ret. (next Line)
0620 PUSH HL ;ja : prüfen
0621 PUSH BC
0630 DUNKEL LD (HL),"
0640 INC HL
0650 DJNZ DUNKEL ;Zeile dunkel zeichnen
0660 POP BC
0670 POP HL
0680 PEN2 XOR A
0690 OUT (PORTDA),A ;Reset Port
0700 CALL DELAY
0710 IN A,(PORTDA)
0720 BIT X,A ;"Sieht Pen dunkel?"
0730 RET NZ ;nein: next Line
0740 FOUND POP DE ;Rücksor.Adr.PTEST löschen
0750 LD A,1
0760 CP B ;zeigt Pen auf 1. Zeichen?
0770 JR Z ENDE ;ja
0780 HALB INC B ;nein; "Zeile" halbieren
0790 SRL B
0800 PUSH HL
0810 POP DE
0820 LD A,B
0830 ADD A,E
0840 LD E,A ;neuer "Zeilenanfang"
0850 CALL PTEST
0860 HALDU PUSH HL ;1.Hälfte falsch:dunkel
0870 PUSH BC
0880 LLOOP LD (HL),"
0890 INC HL
0900 DJNZ LLOOP
0910 POP BC
0920 POP HL
;
0930 SCDHLF EX DEHL ; zweite Hälfte
0940 CALL PTEST
;
0950 JR NOTFND ;beide Hälften falsch
0960 ;
0970 ENDE POP BC
0980 POP DE
0990 PUSH HL
1000 LD HL,BUFBEQ
1010 LD BC,ANZCHA
1020 LDIR ;Restore Screen
1030 POP HL ;POSITION LIGHTPEN
1040 POP DE ;Register Hauptorogramm
1050 POP BC
1060 POP AF
1070 RET ;zum Hauptorog.
1080 ;
1090 NOTFND POP BC
1100 POP DE
1110 LD HL,BUFBEQ
1120 LD BC,ANZCHA
1130 LDIR ;Restore Screen
1140 JP START1 ;Schleife bis Pos. gefunden
1150 ;
1160 DELAY LD A,#10 ;ca.20 ms Verzögerung, bis
1170 ELOOP EX AF,AF' ;Bildsch. voll beschrieben
1180 XOR A
1190 DLOOP DEC A
1200 JR NZ DLOOP
1210 EX AF,AF'
1220 DEC A
1230 JR NZ ELOOP
1240 RET
1250 ;
1260 ;
1270 ;HAUPTPROGRAMM KEYB./LIGHTP.
1280 ;schreibt den Charakter einer gedrückten
1290 ;Taste an die Position des Lichtgriffels
1300 ;
1310 KEYB DEFW #7BDF ;Keyboardabfrage
1320 NOP
1330 CALL LTPEN
1340 LD (HL),A
1350 JR KEYB
1360 ;
1370 ;
1380 ;HAUPTPROGRAMM DRAW/LIGHTP.
1390 ;zeichnet Sternchen mit dem Griffel
1400 ;
1410 DRAW RST #28 ;Clear Screen
1420 DEFW #C
1430 LOOP1 CALL LTPEN
1440 LD (HL),"*
1450 JR LOOP1
1460 ;

```

ZEAP Z80 Assembler - Symbol Table

0080H	0180	ABST	0070H	0170	ANZCHA
0008H	0160	ANZZEI	0F00H	0120	BUFBEQ
0C8CH	0280	BUFFER	0D18H	1160	DELAY
0D1CH	1190	DLOOP	0D2DH	1410	DRAW
0CC9H	0630	DUNKEL	0D1AH	1170	ELOOP
0CFBH	0970	ENDE	0CDBH	0740	FOUND
0CE1H	0780	HALB	0CECH	0860	HALDU
0CB5H	0530	HELL	0D24H	1310	KEYB


```

002AH 0190 LIGHT      OCEEH 0880 LLOOP
0030H 1430 LOOP1      OC80H 0210 LTPEN
000BH 1090 NOTFND     OCB AH 0540 PEN1
0CDOH 0680 PEN2       003FH 0130 PORTCO
003DH 0140 PORTDA     OCB3H 0510 PTEST
0C9DH 0370 RSTORE     0CF5H 0930 SCDHLF
080AH 0110 SCRBEQ     OC83H 0240 START1
0C87H 0260 START2     0000H 0150 X
0CA9H 0440 ZEILE

```

```

0C80 F5 C5 D5 3E 4F D3 07 06 88
0C88 08 21 0A 08 11 00 0F E5 D4
0C90 C5 E5 01 70 00 ED B0 E1 35
0C98 06 70 CD B3 0C E5 E5 01 71
OCA0 70 00 21 00 0F D1 ED B0 BA
OCA8 E1 01 80 00 09 C1 D1 10 C1
OCB0 DB 18 D4 C5 E5 36 2A 23 B0
OCB8 10 FB AF D3 05 CD 18 0D 48
QCC0 DB 05 CB 47 E1 C1 C8 E5 0D
QCC8 C5 36 20 23 10 FB C1 E1 BF
QCD0 AF D3 05 CD 18 0D DB 05 35
QCD8 CB 47 00 D1 3E 01 B8 28 A6
QCE0 1A 04 CB 38 E5 D1 78 83 BE
QCE8 5F CD B3 0C E5 C5 36 20 DF
QCF0 23 10 FB C1 E1 EB CD B3 37
QCF8 0C 18 10 C1 D1 E5 21 00 D0
QD00 0F 01 70 00 ED B0 E1 D1 DC
QD08 C1 F1 C9 C1 D1 21 00 0F 52
QD10 01 70 00 ED B0 C3 83 0C 7D
QD18 3E 10 08 AF 3D 20 FD 08 8C
QD20 3D 20 F7 C9 DF 7B 00 CD 71
QD28 80 0C 77 18 F7 EF 0C 00 42
QD30 CD 80 0C 36 2A 18 F9 00 07

```

```

1940 DEFM "Programm 2  "
1950 DEFB #D
1960 DEFM "Programm 3  "
1970 DEFW #D
1980 ;Abfrage mit Lightoen
1990 ABFR LD HL #818
2000 CALL HLPEN
2010 OR A
2020 JR NZ PROG1
2030 LD HL #858
2040 CALL HLPEN
2050 OR A
2060 JR NZ PROG2
2070 LD HL #898
2080 CALL HLPEN
2090 OR A
2100 JR NZ PROG3
2110 JR ABFR
2120 ;
2130 ;Programme zum Test
2140 ;
2150 PROG1 RST #28
2160 DEFM "erstes Programm"
2170 DEFW #D
2180 JP TEXT
2190 PROG2 RST #28
2200 DEFM "zweites Programm"
2210 DEFW #D
2220 JR TEXT
2230 PROG3 RST #28
2240 DEFM "drittes Programm"
2250 DEFW #D
2260 JP TEXT
2270 ;

```

```

1500 ;SUBROUTINE HLPEN
1510 ;testet, ob Pen auf HL gerichtet.
1520 ;Kehrt zurück mit A=0, wenn
1530 ;Pen nicht vorhanden, sonst anderer Wert
1540 ORG #D40
1550 HLPEN PUSH BC
1560 PUSH DE
1570 LD A, (HL)
1580 PUSH AF ;(HL) retten
1590 LD B, 3 ;Sicherheitsschleife
1600 SLOOP LD (HL), "*"
1610 CALL DELAY
1620 IN A, (PORTDA)
1630 BIT X, A ;Licht?
1640 JR Z NOLITE
1650 XOR A
1660 OUT (PORTDA), A ;Reset Port
1670 LD (HL), "
1680 CALL DELAY
1690 IN A, (PORTDA)
1700 BIT X, A ;dunkel ?
1710 JR NZ NOLITE
1720 DJNZ SLOOP ;prüfen
1730 POP AF
1740 LD (HL), A
1750 JR RETURN
1760 NOLITE POP AF
1770 LD (HL), A
1780 XOR A
1790 RETURN POP DE
1800 POP BC
1810 RET
1820 ;
1830 ;BEISPIEL: MENU MIT LIGHTPEN
1840 ;
1850 MENU LD A, #4F
1860 OUT (PORTCO), A ;Input Mode
1870 RST #28
1880 DEFW #C ;Clear Screen
1890 TEXT LD HL, #80A
1900 LD (#C29), HL ;Cursor
1910 RST #28
1920 DEFM "Programm 1  "
1930 DEFB #D

```

```

0D40 C5 D5 7E F5 06 03 36 2A C3
0D48 CD 18 0D DB 05 CB 47 28 61
0D50 14 AF D3 05 36 20 CD 18 33
0D58 0D DB 05 CB 47 20 06 10 9A
0D60 E5 F1 77 18 03 F1 77 AF EC
0D68 D1 C1 C9 3E 4F D3 07 EF 26
0D70 0C 00 21 0A 08 22 29 0C 13
0D78 EF 50 72 6F 67 72 61 6D 4C
0D80 6D 20 31 20 20 20 20 0D D8
0D88 50 72 6F 67 72 61 6D 6D DA
0D90 20 32 20 20 20 20 0D 50 CC
0D98 72 6F 67 72 61 6D 6D 20 BA
0DA0 33 20 20 20 20 0D 00 21 8E
0DA8 18 08 CD 40 0D B7 20 14 DA
0DB0 21 58 08 CD 40 0D B7 20 2F
0DB8 20 21 98 08 CD 40 0D B7 77
0DC0 20 2C 18 E3 EF 65 72 73 4D
0DC8 74 65 73 20 50 72 6F 67 09
0DD0 72 61 6D 6D 0D 00 C3 72 CC
0DD8 0D EF 7A 77 65 69 74 65 79
0DE0 73 20 50 72 6F 67 72 61 EB
0DE8 6D 6D 0D 00 18 84 EF 64 CB
0DF0 72 69 74 74 65 73 20 50 08
0DF8 72 6F 67 72 61 6D 6D 0D 07
0E00 00 C3 72 0D 00 00 00 50

```

TAUSCHE original 8K-Tape-BASIC gegen
Cassette mit original ROM-BASIC. Suche
Schreibmaschineninterface aus Funkschau
4/80 (Hofer) zu kaufen;
Karl-H. Czerulla, Tel. [REDACTED]

Seite(n) für Einsteiger

Sprünge

Heute darf gesprungen werden, von einem Byte zum nächsten oder wohin auch immer. Bewegung soll anregen, also lassen wir die Speicherzellen rauchen. Die "Intelligenz" eines Rechners wird bekanntlich durch zwei Dinge ermöglicht: 1. Er kann sehr schnell fehlerfrei etwas immer wieder tun und 2. kann er einfache Entscheidungen treffen. Allerdings muß er dabei immer wissen, wo's lang geht.

Die Sprünge, die der Z80 Prozessor kennt, lassen sich nach mehreren Gesichtspunkten einteilen, nämlich nach der Art der Bedingung und der Form, wie die Bedingung beschrieben wird.

Der einfachste Sprung ist derjenige ohne Bedingung. Wenn z.B. der Zielort die Hausnummer 4C3F hat, sieht bei einer direkten Adressierung der entsprechende Befehlscode so aus: C3 3F 4C . C3 kennzeichnet die Sprungart, die beiden Bytes der Adresse folgen in umgekehrter Reihenfolge. Soll der Zielort erst innerhalb eines Programmablaufs errechnet werden, legt man das Ergebnis im HL- oder eines der Indexregisterpaare ab. Geben Sie hier unter NAS-SYS einmal folgendes komplexe Programm ein:

```

OD00 DF 63 lese eine Zeile ein
OD02 DF 64 wandle sie in einen
          Binärwert um, speichere Er-
          gebnis ab 0C21
OD04 2A 21 0C Lade diese Adresse in das
          HL Register
OD07 E9 arbeite ab der Stelle weiter,
          auf die der Inhalt von 0C21
          und 0C22 zeigt.
    
```

Nach dem Start ED00 wird jetzt eine Adresseingabe erwartet. Das Programm springt dann dorthin und arbeitet dann von dieser Stelle weiter. Probieren Sie folgende Werte: 0051, 02EB, 0450.

Die relative Adressierung hat zwei Vorteile: 1. Programme lassen sich ohne Probleme im Arbeitsspeicher verschieben und 2. besteht der OP-Code nur noch aus 2 Bytes. Leider aber muß man, wenn keinen Assembler zur Verfügung hat, doch etwas umständlich die Sprungweite berechnen. Das 2. Byte gibt die Sprungweite bezogen auf den aktuellen Stand des Programmzählers (PC) an. Soll der Sprung rückwärts erfolgen, wird das 8. Bit gesetzt und die restlichen sieben werden in ihrem Wert rückwärts gesetzt (2-Komplement).

Beispiele:

```

OD00 18 04 JR LABEL
OD02 26 02 LD H,2
OD06 00 LABEL NOP
    
```

Die erste Programmzeile wird in den Prozessor eingelesen und der PC um die entsprechende Zahl an Bytes erhöht. Um jetzt zum Label zu gelangen, müssen 4 Bytes übersprungen werden. Deshalb muß das 2. Byte in der 1. Zeile 04 sein. Das 1. Byte steht für einen relativen Sprung.

```

OD00 06 04 LABEL LD B,4
OD02 0E 07 LD C,7
OD04 18 FA JR LABEL
    
```

Um zum LABEL zu gelangen, müssen ab der 3. Zeile insgesamt 6 Bytes übersprungen werden. Will man bei einem Programm mit relativen Sprüngen später noch etwas einfügen, müssen die Sprungweiten natürlich neu berechnet werden. Relative Sprünge können vorwärts maximal 129 Bytes, rückwärts 126 Bytes weit reichen. Mit dem folgenden Programm darf man mit ED00 ein Gefühl für die Weite erhalten!

```

OD00 3E 0C F7 EF 45 73 20 69
OD08 73 74 20 61 6C 6C 65 73
OD10 20 72 65 6C 61 74 69 76
OD18 21 0D 0D 52 65 6C 61 74
OD20 69 76 65 72 20 53 70 72
OD28 75 6E 67 20 28 7A 2E 42
OD30 2E 20 33 46 29 3F 0D 00
OD38 DF 63 DF 64 38 C2 21 21
OD40 0C 46 21 E1 09 22 29 0C
OD48 3E 53 F7 CB 78 28 0C 78
OD50 2F 47 EF 11 11 2A 00 10
OD58 F9 18 05 3E 2A F7 10 FB
OD60 21 4A 0B 22 29 0C EF 4E
OD68 6F 63 6B 20 65 69 6E 6D
OD70 61 6C 3F 20 28 4A 2F 4E
OD78 29 00 CF FE 4A 28 B1 C7
    
```

Wolfgang Mayer-Gür

Hinweis zum Programm SEESCHLACHT von K. Mombaur (Heft 1-82):

H. Auge hat entdeckt, daß das Programm nicht wegen NASSYS3 auf manchen Systemen nicht läuft, sondern wegen der obskuren zweiten BASIC-Version. Entgegen der Hinweise der Redaktion ist das Programm so einfacher anzupassen:

Durch DOKE 4175,-6670 wird der Zeilenvorschub unterdrückt und die Eingabe an der erwarteten Stelle gemacht (Manual Seite 25). Durch DOKE 4175,-25 kann wieder auf die normale Betriebsart umgeschaltet werden (im Spiel nicht nötig).

Mehrere Leser hatten Schwierigkeiten mit der Grafikerweiterung aus Heft 1-82. Peter Deege aus Aachen schreibt dazu:

Ich habe mir diese einfache Grafikerweiterung nachgebaut, aber leider funktionierte sie nicht. Nach Durchdenken des Schaltbilds stellte ich fest, daß die Freigabe der Schieberegister genau umgekehrt erfolgte. Nach den Änderungen IC 15 Pin 7 nach Pin 13 (LS 00) und IC 265 Pin 7 nach Pin 4 (LS 00) tat sich schon 'was, aber noch unsaubere Character und Doppeldarstellungen. Ich tauschte das LS 74 mit einem Standard 74, und die Karte lief einwandfrei.

Programmiersprachen

von W. Mayer-Gürr

Höhere Programmiersprachen und das Acht Damen Problem

Am Beispiel des Acht Damen Problems soll einmal auf die verschiedenen Konzepte einiger Programmiersprachen, die auf dem Nascom laufen, eingegangen werden.

Die älteste problemorientierte Sprache ist Fortran (FORmula TRANslation). Sie wurde 1954 von J. Backus entworfen und dann in mehreren Versionen weiter entwickelt. Zu diesem Zeitpunkt war die Kommunikation zwischen dem Bediener und der Maschine noch recht mühsam. Die Eingabe erfolgte über Lochstreifen oder Lochkarten, die Ausgabe über eine Schreibmaschine oder einen Drucker. Für jede Programmzeile mußte eine Lochkarte gestanzt werden ('Backspace' war d. nur ein schöner Traum). Der Bezug auf die Lochkarte äußert sich immer noch in dem zwingenden Aufbau einer Zeile. Sie umfaßt 80 Stellen. Warum haben Terminals und Drucker wohl eine Zeilenlänge von 80 Zeichen? Die letzten acht Stellen dienen dabei nur zur Information des Programmierers, z.B. fortlaufende Numerierung der Karten. Ein INSERT oder DELETE machte man rein manuell. Die ersten 5 Stellen können ein Label enthalten, eine Zeilennummerierung ist nicht erforderlich. Ist der eigentliche Befehlssteil länger als 66 Zeichen, wird auf der nächsten Karte in der 6. Stelle ein Fortsetzungszeichen gesetzt.

Die Sprache war hauptsächlich für die Lösung mathematischer Probleme gedacht. Daraus erklärt sich die stiefmütterliche Behandlung bei der Bearbeitung von Texten und die die umständliche Formatierung bei der Ausgabe. Flexible ist dagegen die Möglichkeit über eine externe Bibliothek mathematische Routinen, die bereits früher einmal geschrieben wurden, in ein Programm einzubinden.

Die Fortranversion von Microsoft, die auf dem Nascom mit OLD-DUS läuft, ist ein echter Compiler, d.h. aus dem Quellcode wird über 2 Stufen reiner Maschinencode erzeugt. Dieser kann ausgeführt werden, ohne daß dabei im Rechner andere Hilfsprogramme - außer dem DOS - im Rechner präsent sein müssen.

In der 1. Stufe wird durch den Compiler ein relocierbarer Code erzeugt. In der 2. Stufe werden dann durch den Linker Routinen, die in der Library enthalten sind, angebunden und ausführbare Z80 (bzw. 8080) Codes generiert. Allerdings ist die Länge des fertigen Programms relativ groß. Im Nascom Journal 3/80 hat Herr L. Bayer eine Lösung des Acht Damen Problems in Maschinensprache vorgestellt, die nur 1/4 K Speicher belegt. Die Fortran Version dagegen belegt fast 8 K.

Obwohl Basic als reine Interpretersprache konzipiert ist, wird bei den Microsoft Dialekten ein komprimierter Zwischencode erzeugt, also teilweise kompiliert. Dabei wird jeweils ein Befehl zu einem 1 Byte "Token" zusammengefaßt. Kennzeichen eines "Tokens" ist das gesetzte 8. Bit.

Die Verwandtschaft zum Fortran zeigt sich in den GOTO Befehlen, die Sprünge auf ein Label (Zeilennummer) zulassen. Da bei Fortran definitionsgemäß alle Variablen, die mit den Buchstaben I bis N beginnen, als Integer behandelt werden, hat sich als Schleifenzähler die Variable I eingebürgert. Schaut man sich fertige Basic Programme an, so stellt man fest, daß als Schleifenzähler meist auch hier I benutzt wird. Der Mensch ist halt ein Gewohnheitstier.

Über die Vor- und Nachteile von Basic ist schon genug geschrieben worden. Die Sprache ist leicht zu erlernen, komplexe Programme dagegen sind schwer zu lesen und zu warten. Als Interpretersprache ist Basic natürlich relativ langsam.

Das Acht Damen Problem läßt sich bei dem Beispielprogramm auch für größere Felder lösen. Es ist dabei lediglich Zeile 120 zu ändern.

Das Pascalsystem besteht aus zwei Teilen. Durch den Compiler wird aus dem Quelltext ein kompakter Zwischencode, der P-Code, erzeugt. Dieser Code gilt für einen hypothetischen Stackrechner. Das Prinzip dazu ist ähnlich wie bei der Programmiersprache FORTH. Die Microengine von Western Digital kann einen solchen Code (aber auch nur diesen) direkt ausführen. Microcomputer wie der Nascom müssen den P-Code über einen Interpreter umsetzen. Für andere Rechner gibt es auch Pascal Compiler (z.B. Pascal/M), die Maschinencode erzeugen. Allerdings ist hier der Speicherbedarf viel größer, was die Programmlänge sehr einschränkt.

Bei meinem Pascal besteht jeder P-Code aus 4 Bytes. Es gibt auch andere Konzepte, die Interpretation ist dann aber aufwendiger. Das 1. Byte enthält den OP-Code, die restlichen 3 entweder Bedingungen, Zeiger auf Adressen oder auch Konstante.

Die Laufzeiten für die Beispielprogramme lassen deutlich werden, wie die Sprachen aufgebaut sind. Der Maschinencode des Fortranprogramms wird in 60 Sekunden ausgeführt, die Interpretation des P-Codes beim Pascal dauert nur 15 Sekunden länger. Die zeilenweise Bearbeitung durch den Basicinterpreter beansprucht dagegen fast 20 Minuten.

```
100 REM ::: Acht Damen Problem
110 REM
120 Z=B:REM Anzahl der Damen
130 DIM A(Z),B(2*Z),C(2*Z-1),X(Z)
140 I=1
150 GOSUB 170
160 END
170 J=1
180 IF A(J) OR B(I+J) OR C(I-J+Z) THEN 320
190 X(I)=J
200 A(J)=1
210 B(I+J)=1
220 C(I-J+Z)=1
230 IF I<Z THEN I=I+1:GOSUB 170:I=I-1:GOTO 280
240 FOR P=1 TO Z
250 PRINT CHR$(64+P);X(P);" ";
260 NEXT P
270 PRINT
280 J=X(I)
290 A(J)=0
300 B(I+J)=0
310 C(I-J+Z)=0
320 J=J+1
330 IF J<=Z THEN 180
340 RETURN
```

```

C***** Acht Damen Problem in FORTRAN-90 ***
PROGRAM DAME
INTEGER*2 DAME(8)
INTEGER*2 CHD(8)
INTEGER*2 BU(8)

C
DATA CHD/' A', ' B', ' C', ' D', ' E',
1 ' F', ' G', ' H' /
IZ=0

C
DO 1800 I1=1,8
K=1
DAME(K) = 11

C
DO 1700 I2 = 1,8
K=2
IF (I2.EQ.DAME(1) .OR. I2 .EQ. (DAME(1)-1) .OR.
1 I2.EQ.(DAME(1)+1)) GOTO 1700
DAME(K) = 12

C
DO 1600 I3 = 1,8
K=3
L=K-1
DO 1210 J=1,L
IF (I3.EQ.DAME(J) .OR. I3.EQ. (DAME(J)+K-J) .OR.
1 I3.EQ.DAME(J)-K+J) GOTO 1600
1210 CONTINUE
DAME(K)=I3

C
DO 1500 I4 = 1,8
K=4
L=K-1
DO 1310 J=1,L
IF (I4.EQ.DAME(J) .OR. I4.EQ. (DAME(J)+K-J) .OR.
1 I4.EQ.DAME(J)-K+J) GOTO 1500
1310 CONTINUE
DAME(K)=I4

C
DO 1400 I5 = 1,8
K=5
L=K-1
DO 1410 J=1,L
IF (I5.EQ.DAME(J) .OR. I5.EQ. (DAME(J)+K-J) .OR.
1 I5.EQ.DAME(J)-K+J) GOTO 1400
1410 CONTINUE
DAME(K)=I5

C
DO 1300 I6 = 1,8
K=6
L=K-1
DO 1510 J=1,L
IF (I6.EQ.DAME(J) .OR. I6.EQ. (DAME(J)+K-J) .OR.
1 I6.EQ.DAME(J)-K+J) GOTO 1300
1510 CONTINUE
DAME(K)=I6

C
DO 1200 I7 = 1,8
K=7
L=K-1
DO 1610 J=1,L
IF (I7.EQ.DAME(J) .OR. I7.EQ. (DAME(J)+K-J) .OR.
1 I7.EQ.DAME(J)-K+J) GOTO 1200
1610 CONTINUE
DAME(K)=I7

C
DO 1100 I8 = 1,8
K=8
L=K-1
DO 1710 J = 1,L
IF (I8.EQ.DAME(J) .OR. I8.EQ. (DAME(J)+K-J) .OR.
1 I8.EQ.DAME(J)-K+J) GOTO 1100
1710 CONTINUE

C
IZ=IZ+1
IF (IZ .GT. 150) GOTO 990

C
DAME(K)=I8
DO 2000 M=1,8
M1=DAME(M)
BU(M)=CHD(M1)
2000 CONTINUE

```

```

C
WRITE(I,22) (BU(I1),DAME(I1),I1=1,8), IZ
22 FORMAT(' DAME:',8(A2,' ',I1,IX),' IZ =',I6)
C
1100 CONTINUE
1200 CONTINUE
1300 CONTINUE
1400 CONTINUE
1500 CONTINUE
1600 CONTINUE
1700 CONTINUE
1800 CONTINUE
C
STOP
C
990 STOP 'ABBR'
END

```

```

PROGRAM ACHTDAMEN ; PASCAL
VAR
I,N : INTEGER ;
SPALTE : ARRAY [ 1.. 8 ] OF BOOLEAN ;
RUNTER : ARRAY [ 2..16 ] OF BOOLEAN ;
HOCH : ARRAY [-7..+7 ] OF BOOLEAN ;
X : ARRAY [ 1.. 8 ] OF INTEGER ;
PROCEDURE REPORT ;
VAR
K : INTEGER ;
BEGIN
N:=N+1; WRITE(N:3,' ':3);
FOR K:= 1 TO 8 DO WRITE(X(K):5);
WRITELN;
END;
PROCEDURE VERSUCH ( I : INTEGER ) ;
VAR
J : INTEGER ;
BEGIN
FOR J:= 1 TO 8 DO
IF SPALTE [ J ] THEN
IF RUNTER [I+J] THEN
IF HOCH [I-J] THEN
BEGIN
X(I):=J;
SPALTE [ J]:=FALSE;
RUNTER [I+J]:=FALSE;
HOCH [I-J]:=FALSE;
IF I<8 THEN VERSUCH(I+1) ELSE REPORT;
SPALTE [ J]:=TRUE;
RUNTER [I+J]:=TRUE;
HOCH [I-J]:=TRUE;
END;
END (* VERSUCH *) ;
BEGIN
N:=0;
FOR I:= 1 TO 8 DO SPALTE [I]:=TRUE;
FOR I:= 2 TO 16 DO RUNTER [I]:=TRUE;
FOR I:=-7 TO 7 DO HOCH [I]:=TRUE;
VERSUCH(1);
WRITELN('Suche beendet');
END.

```

Wer schreibt eine Lösung des Acht Damen Problems in FORTH?

SUCHE NASSYS, NASPEN, Netzteil N301, Tastaturerweiterung oder Tasten einzeln, Buffer-Bus-Board, RAM-Karte 16, 32 oder 48K, Zeichengenerator für französ. Zeichen MCM 6578, Zeichengenerator mit allen Zeichen v.Skand.b.Span. MCM 6579
Rene Claus, [redacted]
[redacted] Luxemburg
Tel. [redacted]

Preiswertes PASCAL

von Michael Bach

Angeregt durch einen Review im letzten INMC-News habe ich mir in England das "Blue Label Software Pascal" bestellt, da ich es recht preisgünstig fand (57 Pfund) und mal Pascal lernen wollte. Da ich einen Blanko-Verrechnungsscheck beilegte, ging die Lieferung sehr schnell (1 Woche). Ich hatte die Kassetten-Version (1000H-4000H) bestellt, sie wird in 1200Bd im Kansas-City Format geliefert. Es gibt auch eine Eprom-Version von 0000H-FFFFH. Tatsächlich gelang es mir nach einigen Versuchen, die Daten fehlerfrei einzulesen.

Das Pascal-System besteht aus 4 Teilen: Ein "Runtime-Package", ein "Betriebssystem", ein Editor, und der Compiler. Vom Betriebssystem aus können Programme (Quellcode oder Objekt) mit Namen geladen und abgespeichert und die weiteren Funktionen aufgerufen werden: Ein guter cursororientierter Editor, der auch Zeilen mit mehr als 48 Zeichen zuläßt (dann rutscht der ganze Bildschirminhalt nach links); er ist zwar nicht so komfortabel wie der vom UCSD-Pascal aber besser als Naspensowieso. Der Compiler ist sehr schnell und verarbeitet Standard-Pascal mit Ausnahme von Set's Type's und Record's. Dafür bietet er einige Erweiterungen in Richtung Grafik, Stringverarbeitung und direkten Speicherzugriff, d.h. er kann alles, was man von Basic kennt und mehr. Insbesondere ist auch der Datentyp REAL vorhanden mit 11 sicheren Stellen. Wenn beim Compilieren ein Syntaxfehler gefunden wird, springt das System in den Editor und der Cursor blinkt (meistens) genau an der fehlerhaften Stelle (z.B. Semikolon vergessen). Der Compiler erzeugt Z80-Kode (also keinen P-Code), und die Programme laufen daher sehr schnell. Das ist einer der Gründe, warum dies Pascal für mich Basic und Assembler vollständig ersetzt: Es ist so schnell wie Assembler, und es ist komfortabler als Basic. Für Spiele mit Grafik z.B. ist Basic immer zu langsam, und in Assembler dauert das Programmieren viel länger. Pascal-Objektprogramme können für jeden Adressbereich kompiliert werden und sind auch Epromfähig, wenn das Runtime-Package ebenfalls im Speicher ist. Wenn bei der

Programmausführung ein Fehler auftritt (z.B. Division durch 0, Index-Bereichsüberschreitung), kann man in den Editor springen, und der Cursor blinkt wieder an der Fehlerstelle. Nachteile:

1. Die Tastatur-Routine des Pascal funktionierte bei meinem Nascom1 nicht einwandfrei, was sich aber beheben ließ, indem ich die vom Pascal vorgenommene Änderung der Input-Tabelle verhinderte. Beim Nascom2 gibt's diese Probleme nicht.

2. Bei Pascal braucht man die rechteckigen Klammern (für indizierte Zahlenfelder). Das sind aber bei mir die Buchstaben "A" und "O". Das führt zu sehr unleserlichen Programmen. Aber beim erneuten Lesen der Handbücher (2 Hefte Din A5 zusammen 60 Seiten) fand ich, daß diese auch durch die Kombination "(," und ".)" ersetzbar sind. Für Kommentare kann ja sogar in Standard-Pascal statt der geschweiften Klammern die Kombination "(*" eingesetzt werden. Dieses Pascal ist in Dänemark implementiert worden, wahrscheinlich sind dort auch die eckigen Klammern durch nationale Sonderbuchstaben belegt.

Insgesamt kann ich sagen, daß ich von dem Pascal-System regelrecht begeistert bin und wahrscheinlich nächstens Basic und Assembler durch Pascal im Eprom ersetzen (oder zumindest umschalten) werde. Die Sprache Pascal selbst empfinde ich als sehr viel angenehmer als Assembler oder Basic oder Fortran oder Algol oder Forth. Genug des langen Redens, am meisten sagt vielleicht das Beispiel-Programm: Ein Spiel bei dem auf dem Schirm eine Rennbahn gemalt wird (Grafik erforderlich), durch die 2 Spieler ihren Punkt steuern können durch Vorgabe von Beschleunigung und Fahrtrichtung (die Prozedur GETMOVE liest die Stellung eines Tippknüppels ab, dies ließe sich auch über die Tastatur machen (wie bei dem Spiel "Doppelwurm"). Das Spiel ist sehr spannend, weil man leicht aus der Kurve rausfliegen kann und für größte Geschwindigkeit wirklich die Idealkurve fahren muß. Für die trigonometrischen Berechnungen war selbst Pascal zu langsam (beim reinen "Zahlen zerkleinern" sind Interpreter und Compiler ja gleich schnell), deshalb werden für SIN/COS in simpler Weise vorher Tabellen angelegt. Ich hoffe, daß damit noch einige Leute mehr

den Sprung zu Pascal mit dem Nascom wagen und in Zukunft viele Pascal-Programme an dieser Stelle veröffentlicht werden (ich habe Jedenfalls noch einige auf Lager).

PROGRAMM RENNEN2 5.5.82;
 (*1-2 SPIELER KÖNNEN AUF EINER BAHN KREISEN,
 WER RAUSFLIEGT, MUSS DIE RUNDE NEU BEGINNEN.*)

```
CONST XMAX=95; YMAX=47; BREI=12; XSTART=43;
DV=0.015; DPHI=3;
VAR S, I, J, PTX, PTY, X, Y, XA, YA, ZUGZ, ZUGZ1: INTEGER;
PHI, FEHL, RUNDZ, IX, IY: ARRAY(.0..1.) OF INTEGER;
R, R1, R2: REAL;
RX, RY, V: ARRAY(.0..1.) OF REAL;
SINT, COST: ARRAY(.0..359.) OF REAL;
ENDE: BOOLEAN;
```

```
FUNCTION TBREAK: BOOLEAN; (*TESTE AUF ESC*)
BEGIN
TBREAK:=KEYBOARD=$1B
END;
```

```
(*INITIALISIERE PORT A*)
PROCEDURE INIPORT; BEGIN; OUT(6,$CF); OUT(6,$FF);
END;
```

```
PROCEDURE BAHN; (*BAHN MALEN*)
VAR I, J: INTEGER;
BEGIN
FOR I:=0 TO YMAX DO BEGIN (*AUSSENRAHMEN*)
PLOT(0,I,1); PLOT(XMAX,I,1);
END;
FOR I:=1 TO XMAX DO BEGIN
PLOT(I,0,1); PLOT(I,YMAX,1);
END;
FOR I:=BREI TO YMAX-BREI DO BEGIN (*INNENRAHMEN*)
PLOT(BREI,I,1); PLOT(XMAX-BREI,I,1);
END;
FOR I:=BREI TO XMAX-BREI DO BEGIN
PLOT(I,BREI,1); PLOT(I,YMAX-BREI,1);
END;
(*STARTLINIE*)
FOR I:=0 TO BREI DIV 2 DO PLOT(XSTART,I*2,1);
(*SPIELER*)
FOR I:=0 TO 1 DO PLOT(IX(I),IY(I),1);
END; (*BAHN*)
```

```
PROCEDURE BESCHR; (*BESCHRIFTUNG*)
BEGIN
PTX:=BREI DIV 2+3; PTY:=BREI DIV 3+2;
SCREEN(PTX,PTY); WRITE('* RENNEN *');
SCREEN(PTX,PTY+2); WRITE('Zeit:');
```

```
SCREEN(PTX,PTY+3); WRITE('Runden:');
SCREEN(PTX,PTY+4); WRITE('Fehler:');
END; (*BESCHR*)
```

```
PROCEDURE GETMOVE; (*ZUGEINGABE*)
VAR I, J, K: INTEGER;
BEGIN
I:=INP(4); K:=1; IF S=0 THEN I:=I SHIFT -4;
FOR J:=0 TO 3 DO BEGIN
IF (NOT(I) AND K) THEN CASE J OF
0: PHI(.S.):=PHI(.S.)+DPHI;
1: PHI(.S.):=PHI(.S.)-DPHI;
2: V(.S.):=V(.S.)+DV;
3: V(.S.):=V(.S.)-DV;
END (*CASE*);
K:=K*2;
END;
IF PHI(.S.)<0 THEN PHI(.S.):=PHI(.S.)+360;
IF PHI(.S.)>359 THEN PHI(.S.):=PHI(.S.)-360;
END; (*GETMOVE*)
```

```
PROCEDURE NEUPOS; (*BESTIMME NEUE POSITION*)
BEGIN;
RX(.S.):=RX(.S.)+V(.S.)*COST(.PHI(.S.));
IX(.S.):=ROUND(RX(.S.)); X:=IX(.S.);
RY(.S.):=RY(.S.)+V(.S.)*SINT(.PHI(.S.));
IY(.S.):=ROUND(RY(.S.)); Y:=IY(.S.);
END; (*NEUPOS*)
```

```
FUNCTION TRAU: BOOLEAN; (*TEST OB RAUS AUS DER BAHN*)
)
BEGIN
TRAU:=FALSE;
IF (X<1) OR (X=XMAX) OR (Y<1) OR (Y=YMAX) THEN TRAU
:=TRUE;
IF (X>BREI) AND (X=XMAX-BREI) AND (Y>BREI) AND (Y=YMAX-BREI) THEN TRAU:=TRUE;
END; (*TRAU*)
```

```
PROCEDURE STARTPOS;
BEGIN
X:=XSTART+1; IX(.S.):=X; RX(.S.):=X;
Y:=S*2-1+BREI DIV 2; IY(.S.):=Y; RY(.S.):=Y;
PHI(.S.):=0; V(.S.):=0;
END;
```

```
BEGIN (*HAUPTPROGRAMM*)
WRITELN('Kleinen Augenblick, ich muß noch eine SIN/COS-');
WRITELN('Tabelle anlegen.');
```

(*SIN/COS-TABELLE ANLEGEN*)

```
R1:=0; R2:=1; J:=0;
FOR I:=0 TO 359 DO BEGIN
SINT(I.):=R1; COST(I.):=R2; J:=J+1;
```

```

IF J<5 THEN BEGIN
J:=0; R:=I*PI/180.0;
R1:=SIN(R); SINT(.I.):=R1; R2:=COS(R); COST(.I.):=R2;
END;
END;
WRITELN(CHR(12),CHR(13),'          *** RENNEN ***'); WRITELN;
(*INITIALISIERUNGEN*)
WRITE(CHR(12)); INIPOS; ZUGZ:=0; ZUGZ:=0; ENDE:=FALSE;
INIT RUNDZ TO 0,0; INIT FEHL TO 0,0;
(*STARTPOSITION*)
FOR S:=0 TO 1 DO STARTPOS;
BAHN; BESCHR;

(*HAUPSCHLEIFE*)
REPEAT
(*ZUG ZÄHLEN UND AUSGEBEN*)
ZUGZ1:=ZUGZ1+1; IF ZUGZ1<20 THEN BEGIN
ZUGZ1:=0; ZUGZ:=ZUGZ+1; SCREEN (PTX+6,PTY+2); WRITE(ZUGZ);
END;

(*SCHLEIFE FÜR 2 SPIELER*)
FOR S:=0 TO 1 DO BEGIN
(*ZUG HOLEN & AUSFÜHREN*)
GETMOVE; XA:=IX(.S.); YA:=IY(.S.); NEUPOS;
(*NICHT RÜCKWÄRTS ÜBER DIE STARTPOSITION*)
IF ABS(XA-XSTART)≠10 THEN
IF V(.S.)≠0 THEN
IF YA≠BREI THEN STARTPOS;

(*NEUE POSITION ÜEBERPRÜFEN*)
IF TRAUS THEN BEGIN (*RAUSGEFLOGEN!*)
STARTPOS; FEHL(.S.):=FEHL(.S.)+1;
SCREEN(PTX+8+5*S,PTY+4); WRITE(FEHL(.S.):2);
END ELSE BEGIN (*NICHT RAUSGEFLOGEN*)
IF Y≠BREI THEN
IF X≠XSTART THEN
IF XA≠XSTART THEN
BEGIN (*RUNDE FERTIG*) RUNDZ(.S.):=RUNDZ(.S.)+1;
SCREEN(PTX+8+5*S,PTY+3); WRITE(RUNDZ(.S.):2);
END;
END;
(*PUNKT NEU ZEICHNEN FALLS BEWEGT*)
IF (XA≠X) OR (YA≠Y) THEN BEGIN
PLOT(XA,YA,2); PLOT(X,Y,2);
END;
IX(.S.):=X; IY(.S.):=Y;
END; (*S*)
UNTIL TBREAK;
END.

```



Die Mitarbeiter dieser Ausgabe:

Günter Böhm

Karlsruhe

Günter Kreidl

Straelen

Wolfgang Mayer-Gürr

Recklinghausen

Michael Bach

Stegen

Dieter Oberle

Vollmersweiler

Peter Deege

Aachen

Tel.?

Andreas Zippel

Karlsruhe

R.Maurer hat einen neuen Befehl für die Sprache BIRNE entwickelt: "MATSCH" entspricht dem NEW in BASIC, ist aber viel komfortabler, da existierende Programme wirklich unleserlich gemacht werden und eine Regenerierung garantiert unmöglich ist. Die BIRNE-Eproms werden ab sofort mit der Erweiterung MATSCH geliefert, der Bleistift wird dafür aus dem Lieferprogramm gestrichen.

Datenverwaltung

Teil 2 von Günter Böhm

Zunächst einige Anmerkungen zu Teil 1. In die Routine REDUZ hat sich ein Fehler eingeschlichen. Das Verkürzen der Files funktionierte einwandfrei, wenn man nicht ausge-rechnet das letzte File verändern wollte. Hierbei mußte das ganze Programm zusammenstürzen, da sich für den Rest der Datei die Länge 0 ergab und der LDIR Befehl nun den ganzen Speicher verschob. Zu beheben ist der Fehler durch Eintragen von INC BC vor dem LDIR in Zeile 1590.

Nicht so tragisch, aber ärgerlich ist die Tatsache, daß beim Aufruf von INPUT auch bei Bearbeitung von Formularen RAMTOP (Speicher-ende für Files) in die Formularparameter geladen wurde. Diese Manipulation muß vor INPUT bereits in INIT eingetragen werden.

```

2100 ;DATEI INITIIEREN
2110 INIT LD HL, FORBEG ;FORMULARE
2120 LD (FORMPA),HL
2130 LD (FORMPA+2),HL
2140 LD (FORMPA+6),HL
2150 LD HL, 1
2160 LD (FORMPA+4),HL
2170 LD HL, RAMBOT-1
2180 LD (FORMPA+8),HL
2190 ;
2200 LD HL, RAMTOP ;FILES
2210 LD (RAMEND),HL
2220 LD HL, RAMBOT
2230 LD (DATEI),HL
2240 LD (ASTFIL),HL
2250 INPUT LD DE, 1 ;min.File Länge
2260 LD (ALEFIL),DE
2270 INC HL ;Dateiende=Fileende+1
2280 LD (DATEND),HL

```

Beim Ausdrucken von Files in bestimmtem Format hat sich als vorteilhaft erwiesen, vor dem Ausdruck des Formulars ein New Line einzufügen, damit das nächste Formular nicht mitten in der Zeile beginnt oder (da ja vor dem BELL Zeichen immer ein Space steht) eingerückt wird.

In der Routine EINGAB wurde beim Drücken von New Line noch die automatische Ausgabe eines Space eingefügt. Dies erleichtert das Auf-finden von Daten in maskierten Files (Teil 2).

Die Teilnehmer am Rundlauf müssen zur Kor-
rektur das Hexdump von 10BE-10EC dem Listing
in diesem Heft anpassen.

Wie bereits angekündigt, soll das Programm laufend erweitert werden. Hier nun die erste

Erweiterung; die Erstellung und Verarbeitung maskierter Files.

Im Menu sind zusätzlich die Möglichkeiten "Berechnungen B" und "Maskierte Files M" vorgesehen worden. Das Drücken von B erzeugt vorläufig einen Sorung nach #1C00, wo möglicherweise zukünftige Berechnungsroutinen beginnen könnten. Mit M springt man nach #1922, wo folgende Programme zur Wahl stehen:

```

"Maskierte Datei initiieren 1"
"Maskierte Files ausgeben 2"
"Eingabe in maskierte Files 3"

```

Bei der Auswahl von "1" erscheint in der Titelzeile die Aufforderung "Data Input". Nun kann man mit frei beweglichem Cursor ein Formular erstellen (in unserem Beispiel eine Schülerkartei), wobei der Doppelpunkt, gefolgt von Space, angibt, daß nun ein Datenwort folgt, welches im File abgespeichert werden soll. Dieses wird durch Space abgeschlossen. Das nächste Datenwort muß wieder durch Doppelpunkt eingeleitet werden. Die Anzahl der Zeichen nach dem Doppelpunkt ist nun das für alle Files verbindliche Format und ermöglicht Berechnungen und auch formatierten Ausdruck.

```

FILE No 01
Klasse 8b 1981/82 ENGLISCH 2.Halbj.
Nr.: 00
Name: "Maske"
5.Dict./Grammar 78 3.3.82 Note: 0.0→
6.RCT "Slowly Killing..." 31.3.82 Note: 0.0→
7.Dict. 5.5.82 Note: 0.0→
8.RCT ? ? ? Note: 0.0→
Test Voc./Gramm. 16.3.82 Note: 0.0→
Test ? ? ? Note: 0.0→
mündliche Beiträge ----- Note: 0.0→
mündl. Vocabulary ? ----- Note: 0.0→
mündl. Conversation ----- Note: 0.0→
mündl. Grammar ----- Note: 0.0→
Durchschnitte schrl.: 0.0 mün.: 0.0→
1.Halbj.: 0.0 ges.: 0.0

```

Es empfiehlt sich, bei der Eingabe von z.B. Namen zunächst eine entsprechende Anzahl von A0 einzugeben (Graph+ Spacer), da diese zwar Platz für das formatierte Datenwort freihalten (Space würde als Beginn des nächsten Wortes interpretiert), von den meisten Druckern aber einfach als Space ausgegeben werden. Aus dem gleichen Grund wird vom Programm bei der Eingabe in

formatierte Files Space als A0 interpretiert. (So werden zwar Name und Vorname vom Drucker getrennt, vom Programm aber als ein Datenwort aufgefaßt).

Klasse 8b 1981/82 ENGLISCH 2.Halbj.			
Nr. :	001	<i>← 1 Datenwort 0</i>	
Name:	Nascomp1,		<i>Datenwort 3</i>
5.Dict./Grammar	3.3.82	Note:	Franz-Josef
6.RCT "Slowly Killing..."	31.3.82	Note:	<i>← 4</i>
7.Dict.	5.5.82	Note:	4.1
8.RCT	???	Note:	2.8
Test Voc./Gramm.	16.3.82	Note:	0.0
Test	???	Note:	0.0
mündliche Beiträge	-----	Note:	2.3

Als Zwischenraum wurde #20 eingegeben. Nachname und nachfolgende Spaces werden nun jeweils als selbständige Datenwörter interpretiert.

Klasse 8b 1981/82 ENGLISCH 2.Halbj.			
Nr. :	001	(automatisch als A0 1)	
Name:	Nascomp1,	<i>← Franz-Josef</i>	
5.Dict./Grammar	3.3.82	Note:	4.1
6.RCT "Slowly Killing..."	31.3.82	Note:	2.8
7.Dict.	5.5.82	Note:	0.0
8.RCT	???	Note:	0.0
Test Voc./Gramm.	16.3.82	Note:	2.3
Test	???	Note:	0.0
mündliche Beiträge	-----	Note:	0.0
mündl. Vocabulary	-----	Note:	0.0
mündl. Conversation	-----	Note:	0.0
mündl. Grammar	-----	Note:	0.0
Durchschnitt	schr1.: 0.0	mün.:	0.0
	1.Halbj.: 3.2	ges.:	0.0

Zwischenräume und freier Platz nach dem Namen sind als #A0 eingegeben. Sie werden zusammen als ein Datenwort aufgefaßt.

Nach Betätigung der ESCAPE Taste fragt der Rechner "Number of Files". Nun gibt man (max. vierstellig) die benötigte Anzahl ein. Falsche Eingabe (Buchstaben) wird durch "Wrong Input" quittiert. Reicht der Speicherplatz für die benötigte Anzahl von Files, so erscheint in der Titelzeile "All Files Loaded", andernfalls "Number of Files too high". -

Nach dem Laden der Files in den Speicher kehrt das Programm zum "Masken Menu" zurück. Das Drücken einer beliebigen Taste läßt zum Start des Dateiprogramms springen. Dort können die Files in gewohnter Weise (wie in Teil 1 besprochen) verarbeitet werden. Durch Drücken von "2" werden die Files in der Maske angezeigt. Man muß sie durch "File Number" anwählen. (Das hat sich im Falle der Schülerkartel als vorteilhaft erwiesen). Will man Maskierte Files durch Stichwörter aufsuchen, muß man den Umweg über das normale Suchprogramm machen. Hier wären Ihre

Erfahrungen mit speziellen Anwendungen interessant, ob man nicht vielleicht auch eine gesonderte Suche für Maskenfiles einbauen sollte. Vor der Ausgabe kann Printer oder Screen gewählt werden, der jeweilige Modus wird in der Titelzeile angezeigt.

Durch "3" wird die Eingabe angewählt. Sie fragt nach "Word Number". Mit einer bis zu vierstelligen Zahl können Sie nun bestimmen, welches Datenwort Sie eingeben oder ändern wollen. (In unserem Beispiel wäre die Nummer 7 der Test vom 16.3.82). Nach der Wortnummer wird die Filenummer eingegeben. Daraufhin erscheint das File in der Maske auf dem Schirm, und der Cursor blinkt an der entsprechenden Stelle. Wird bei der Eingabe die vorgesehene Länge des Wortes überschritten, so springt der Cursor immer wieder auf die erste Stelle; es kann also nie ein falsches Wort überschrieben werden. Die auf dem Schirm sichtbare Eingabe wird durch New Line im File abgespeichert. Durch ESCAPE springt man ins Menu zurück, ansonsten kann die nächste Filenummer eingegeben werden. Für Leser, die den Assembler nicht im Rundlauf erhalten haben, folgt hier das vollständige Hexlisting. Gestartet wird bei #151A.

Ich würde mich über Erfahrungsberichte, Verbesserungsvorschläge und mögliche Erweiterungen freuen.

```

7730 ;RECHENROUTINEN (Ergebn.Jew.In HL)
7740 ;MULTIPLIZIEREN
7750 MULT PUSH AF ;HL mal B
7760 PUSH DE
7770 PUSH BC
7780 XOR A
7790 CP,B
7800 EX DE,HL
7810 LD HL,0
7820 JR Z,END1
7830 ADDIER ADD HL,DE
7840 JR C,ERROR1
7850 DJNZ ADDIER
7860 END1 POP BC
7870 POP DE
7880 POP AF
7890 RET
7900 ;
7910 ;DIVIDIEREN HL durch BC
7920 DIVID PUSH AF
7930 PUSH DE
7940 LD DE,0
7950 XOR A
7960 SUBTRA SBC HL,BC
7970 JR C,END2
7980 INC DE
7990 JR SUBTRA
8000 END2 EX DE,HL
8010 POP DE
8020 POP AF
8030 RET
8040 ;

```

```

8050 ERROR1 RST #28
8060 DEFB #D
8070 DEFM "ILLEGAL CARRY ERROR"
8080 DEFW #D
8090 RET
8100 ;
8110 ;DEZ,TASTATUREINGABE IN HEX (4-STELLIG)
8120 KEYHEX DEFW #63DF ;INLIN
8130 DEFW #79DF ;RLIN
8140 JR C ERROR2
8150 LD HL,(#COC) ;ARG1
8160 XOR A ; Test auf ungült. Eing.
8170 LD A,H
8180 AND #F
8190 ADD A,#F6
8200 JR C ERROR2
8210 LD A,H
8220 ADD A,#60
8230 JR C ERROR2
8240 LD A,L
8250 AND #F
8260 ADD A,#F6
8270 JR C ERROR2
8280 LD A,L
8290 ADD A,#60
8300 JR C ERROR2
8310 WANDL PUSH HL
8320 SRL,H
8330 SRL,H
8340 SRL,H
8350 SRL,H
8360 LD B,H
8370 LD HL,#3E8
8380 CALL MULT
8390 EX DE,HL
8400 POP HL
8410 PUSH HL
8420 LD A,H
8430 AND #F
8440 LD B,A
8450 LD HL,#64
8460 CALL MULT
8470 ADD HL,DE
8480 EX DE,HL
8490 POP HL
8500 PUSH HL
8510 SRL,L
8520 SRL,L
8530 SRL,L
8540 SRL,L
8550 LD B,L
8560 LD HL,#A
8570 CALL MULT
8580 ADD HL,DE
8590 EX DE,HL
8600 POP HL
8610 LD A,L
8620 AND #F
8630 LD L,A
8640 LD H,0
8650 ADD HL,DE
8660 RET ;Ergebn.in HL
8670 ;

```

```

8680 ERROR2 RST #28
8690 DEFB #D
8700 DEFM "Wrong Inout"
8710 DEFW #D
8720 JR KEYHEX
8730 ;
8740 ;DATENWÖRTER AUS MASKE IN BUFFER
8750 DATBUF LD HL,#FOO ;clear Buffer
8760 LD DE,BUFWD1
8770 SBC HL,DE ;Länge
8780 SLOOP LD A,#FF
8790 LD (DE),A
8800 INC DE
8810 DEC HL
8820 LD A,H
8830 OR L

```

```

8840 JR NZ SLOOP
8850 LD A,7 ;BELL
8860 LD (#FOO),A ;BUFFERENDE
8870 LD DE,BUFWD1 ;START BUFFER
8880 LD HL,(ASTFIL) ;START MASKE
8890 COLON LD A,(HL) ;Wortbeg.wird durch "; und
8900 INC HL ;Space,Ende durch Sp. oder
8910 CP,7 ;"09 gekennz.
8920 JR Z ENDMAS
8930 CP ";
8940 JR NZ COLON
8950 GEFU INC HL
8960 LD A,(DE)
8970 CP,7
8980 JR Z ENDBUF
8990 LD A,(HL)
9000 LD (DE),A
9010 INC DE
9020 CP,"
9030 JR Z COLON ;nicht Wortende
9040 CP,9
9050 JR NZ GEFU
9060 INC HL
9070 JR COLON
9080 ENDBUF RST #28
9090 DEFW #C
9100 LD HL TOPLIN
9110 LD (CURSOR),HL
9120 RST #28
9130 DEFM "ERROR: Data too long"
9140 DEFW #D
9150 CALL NORMAL
9160 POP HL ;RETADR.
9170 JP START
9180 ;
9190 ENDMAS LD (DE),A
9200 RET
9210 ;
9220 ;FILEMENGE,ANZ.ZEICHEN LADEN
9230 LDNUM RST #28
9240 DEFB #D
9250 DEFM "Number of Files?"
9260 DEFW #D
9270 CALL KEYHEX
9280 LD (BUFFER),HL ;Fileanzahl
9290 LD HL,BUFWD1
9300 LD BC,0
9310 MALE LD A,(HL) ;Länge d.Daten incl.Bell
9320 CP,7
9330 INC HL
9340 INC BC
9350 JR NZ MALE
9360 LD (BUFFER+2),BC ;Zeichenanzahl
9370 ; weiter mit LOADMA
9380 ;
9390 ;FILES IN MASKENFORMAT LADEN
9400 LOADMA LD HL,(DATEI)
9410 LD (ASTFIL),HL ;Start Datei
9420 LOOPLD LD HL,(ASTFIL)
9430 ADD HL,BC
9440 LD DE,RAMTOP
9450 XOR A
9460 SBC HL,DE
9470 JR Z OK
9480 JR C OK
9490 RST #28
9500 DEFB #D
9510 DEFM "Number of Files too high"
9520 DEFW #D
9530 JR LDNUM
9540 OK LD DE,(ASTFIL)
9550 LD HL,BUFWD1
9560 LDIR
9570 LD (ASTFIL),DE
9580 LD (DATEND),DE
9590 LD BC,(BUFFER) ;dec.Fileanz.
9600 DEC BC
9610 LD A,B
9620 OR C
9630 JR Z FINI

```

```

9640 LD (BUFFER),BC
9650 LD BC,(BUFFER+2)
9660 JR LOOPLD
9670 FINI RST #28
9680 DEFW #C
9690 LD HL,TOPLIN
9700 LD (CURSOR),HL
9710 RST #28
9720 DEFM "All Files loaded"
9730 DEFB,0
9740 RET

0010 ;Datenverwaltung Teil 2
0020 ;Ver.1.2/25.4.82
0030 ;(c)Günter Böhm, Karlsruhe
0040 ASTFIL EQU #2002
0050 AUSGAB EQU #200B
0060 BLINK EQU #7BDF
0070 CURSOR EQU #0C29
0080 DATBUF EQU #1822
0090 DATEI EQU #2000
0100 DATEND EQU #2006
0110 EXTEND EQU #14CA
0120 FORMPA EQU #200D
0130 INIT EQU #1152
0140 KEYHEX EQU #17B4
0150 LDNUM EQU #1885
0160 NORMAL EQU #1750
0170 PRINT EQU #1767
0180 RETTEN EQU #1739
0190 SCREEN EQU #10F1
0200 START EQU #151A
0210 TOPLIN EQU #0BCA
0220 ;
0230 ORG #1922
0240 MASKE RST #28
0250 DEFB #C
0260 DEFM "Maskierte Datei INITIIEREN 1"

0270 DEFB #D
0280 DEFM "Maskierte Files ausgeben 2"

0290 DEFB #D
0300 DEFM "EINGABE in mask. Files 3"

0310 DEFW #DOD
0320 DEFW #DOD
0330 DEFW #DOD
0340 DEFW #DOD
0350 DEFW #DOD
0360 DEFW #D
0370 M1 DEFW BLINK
0380 CP "1
0390 JP NZ M2
0400 CALL INIT

0410 CALL RETTEN
0420 CALL EXTEND
0430 CALL DATBUF
0440 CALL NORMAL
0450 CALL LDNUM
0460 JP MASKE
0470 ;
0480 M2 CP "2
0490 JR NZ M3
0500 CALL HMAFIO
0510 JP MASKE
0520 M3 CP "3
0530 JP NZ START
0540 CALL MAFIEI
0550 JP MASKE
0560 ;
0570 ;maskierte Files ausgeben
0580 ;Ausgabemod. zuvor in "AUSGAB"
0590 MAFIAU LD DE,(FORMPA) ;Maske=Form1
0600 LD HL #80A
0610 LD (CURSOR),HL
0620 MASK LD A,(DE)

0630 CP ";
0640 JR Z FILE
0650 CP Z
0660 RET Z
0670 CALL AUSGAB-1
0680 INC DE
0690 JR MASK
0700 FILE CALL AUSGAB-1 ;";
0710 LD A,"
0720 CALL AUSGAB-1 ;Space
0730 LD HL,(ASTFIL)
0740 FILEOUT LD A,(HL)
0750 PUSH AF
0760 CALL AUSGAB-1
0770 POP AF
0780 CP "
0790 INC HL
0800 JR NZ FILEOUT
0810 LD (ASTFIL),HL ;next Data
0820 INC DE
0830 INC DE
0840 LOOPF LD A,(DE) ;next Maskenwort
0850 CP "
0860 INC DE
0870 JR NZ LOOPF
0880 LD A,(DE)
0890 CP Z
0900 RET Z ;Maskenende
0910 JR MASK ;nächstes Maskenwort
0920 ;
0930 ;FILE NACH NUMMER FINDEN
0940 WHATFI LD HL,#B8A ;CLRS,TOPLINE retten
0950 LD (CURSOR),HL
0960 RST #28
0970 DEFW #DOD
0980 DEFW #DOD
0990 DEFW #DOD
1000 DEFW #DOD
1010 DEFW #DOD
1020 DEFW #DOD
1030 DEFW #DOD
1031 DEFW #D
1040 TITEL LD HL,#84A
1050 LD (CURSOR),HL
1060 RST #28
1070 DEFM "File Number ?"

1080 DEFW #D
1090 CALL KEYHEX
1100 PUSH HL
1110 POP BC ;Nr in BC
1120 LD HL,(DATEI)
1130 ISNUM DEC BC
1140 LD A,B
1150 OR C
1160 JR Z RETURN
1170 LOOPNR LD A,(HL)
1180 INC HL
1190 CP Z
1200 JR Z ISNUM
1210 JR LOOPNR
1220 RETURN PUSH HL
1230 XOR A
1240 LD DE,(DATEND)
1250 SBC HL,DE
1260 POP HL
1270 JR Z ERROR3
1280 JR NC ERROR3
1290 RET
1300 ERROR3 LD HL,#80A
1310 LD (CURSOR),HL
1320 RST #28
1330 DEFM "No Such Number!"

1340 DEFB 0
1350 JR TITEL
1360 ;
1370 ;Screenausgabe eines Zeichens
1380 ;09 in 0D gewandelt

```

```

1390 OUTSCR PUSH AF
1400 RST #30
1410 POP AF
1420 CP 9
1430 RET NZ
1440 LD A,#D
1450 JR OUTSCR
1460 ;
1470 ;"HAUPTPROGR,"MASK.FILE AUSGEBEN
1480 HMAFIO RST #28
1490 DEFW #C
1500 LD HL,TOPLIN
1510 LD (CURSOR),HL
1520 RST #28
1530 DEFM "Output Masked Files "

1540 DEFB 0
1550 LD HL,#80A
1560 LD (CURSOR),HL
1570 RST #28
1580 DEFM "Screen 1"

1590 DEFB #D
1600 DEFM "Print 2"

1610 DEFW #D
1620 CHOOSE DEFW BLINK
1630 CP "1"
1640 JR NZ ZP
1650 LD HL,TOPLIN+20
1660 LD (CURSOR),HL
1670 RST #28
1680 DEFM "Screen"

1690 DEFB 0
1700 LD HL,OUTSCR
1710 LD (AUSGAB),HL
1720 JR AUSG
1730 ZP CP "2"
1740 JR NZ CHOOSE
1750 LD HL,TOPLIN+20
1760 LD (CURSOR),HL
1770 RST #28
1780 DEFM "Print"

1790 DEFB 0
1800 LD HL,PRINT2
1810 LD (AUSGAB),HL
1820 AUSG CALL WHATFI
1830 LD (ASTFIL),HL
1840 CALL MAFIAU
1850 LOOPB DEFW BLINK
1860 CP #D
1870 JR Z AUSG ;NL=next Filenumber
1880 CP #1B
1890 JR NZ LOOPB
1900 LD HL,SCREEN
1910 LD (AUSGAB),HL
1920 RET ;ESC=back to Menue
1930 ;
1940 ;Data-Wortnummer auf Scr.u.i.Spei.finden
1950 FNDWD PUSH IX ;Wortnummer in IX
1960 POP BC ;nach BC
1970 PUSH BC
1980 LD HL,#80A
1990 LOOPW1 LD A,(HL)
2000 INC HL
2010 CP 0
2020 JR Z LOOPW1 ;Margins
2030 CP " "
2040 JR NZ LOOPW1
2050 INC HL ;Wordstart Screen
2060 DEC BC
2070 LD A,B

2080 OR C
2090 JR NZ LOOPW1
2100 POP BC ;Speicher
2110 LD DE,(ASTFIL)
2120 LOOPW2 DEC BC
2130 LD A,B
2140 OR C
2150 RET Z

;Wortstart in (DE)
2160 LOOPW3 INC DE
2170 LD A,(DE)
2180 CP " "
2190 JR NZ LOOPW3
2200 INC DE
2210 JR LOOPW2
2220 ;
2230 ;EINGABE EINES BEST.WORTES IN FILE
2240 MAFIEI RST #28
2250 DEFW #C
2260 LD HL,OUTSCR ;Ausg.auf Schirm
2270 LD (AUSGAB),HL
2280 LD HL,TOPLIN
2290 LD (CURSOR),HL
2300 RST #28
2310 DEFM "Word Input in Masked File "

2320 DEFB 0
2330 LD HL,#80A
2340 LD (CURSOR),HL
2350 RST #28
2360 DEFM "WORD Number ?"

2370 DEFW #D
2380 CALL KEYHEX
2390 PUSH HL
2400 POP IX ;Wortnummer in IX
2410 NXTCH RST #28
2420 DEFW #DOD
2430 DEFW #D
2440 CALL WHATFI
2450 LD (ASTFIL),HL
2460 PUSH HL
2470 CALL MAFIAU
2480 POP HL
2490 LD (ASTFIL),HL
2500 MIST CALL FNDWD
2510 LD (CURSOR),HL
2520 CHWD PUSH DE
2530 DEFW BLINK
2540 POP DE
2550 CP #13 ;Cursor ud
2560 JR Z CHWD
2570 CP #14 ;down
2580 JR Z CHWD
2590 CP 8 ;Backsp.
2600 JR Z CHWD
2610 CP #1B ;ESC
2620 JR Z ENDCH
2630 CP #D ;NL=next File
2640 JR Z NXTCH
2650 CP " " ;Sp.mit Bit 7 um Wordend zu
2660 JR NZ OUTC ;vermeiden
2670 LD A,#A0
2680 OUTC PUSH AF
2690 RST #30
2700 POP AF
2710 CP #11 ;Cursor left
2720 JR NZ RIGHT
2730 DEC DE
2740 JR CHWD
2750 RIGHT CP #12
2760 JR NZ LDWD
2770 INC DE

```

```

2780      JR      CHWD
2790 LDWD  LD      B,A ;retten
2800      LD      A,(DE)
2810      CP      "
2820      JR      Z,MIST
2830      LD      A,B
2840      LD      (DE),A
2850      INC    DE
2860      JR      CHWD
2870 ENDCH LD      HL,SCREEN
2880      LD      (AUSGAB),HL
2890      RET
2900      ;
2910 PRINT2 CP      9
2920      JR      NZ,OUTPR
2930      LD      A,#D
2940 OUTPR CALL RET
2950      RET
2960      ;

```

```

1000 21 0A 08 22 29 0C DF 7B F4
1008 FE 1B 28 0F FE 0D 08 08 9B
1010 3E 20 20 F7 3E 0D 0D 0D FE
1018 F7 18 0B 07 3E 07 07 07 09
1020 01 01 00 00 20 21 09 08 AF
1028 03 23 BE 28 FC 3E 09 BE 45
1030 20 07 AF 23 BE 20 FC 18 2B
1038 EE 3E 07 BE 20 E9 AF 2A 1B
1040 04 20 ED 4E 28 54 30 76 C5
1048 C5 E1 AF ED 5B 04 20 ED 06
1050 52 EB 2A 06 20 19 ED 5B 4E
1058 08 20 EB ED 52 30 17 EF FO
1060 46 49 4C 45 20 54 4F 4F A2
1068 20 4C 4F 4E 47 00 DF 7B 22
1070 FE 0D 20 2A FA 06 20 AF ED
1078 5B 02 20 2A 06 20 AF ED F1
1080 52 E5 C5 E1 ED 5B 04 20 D9
1088 ED 52 E5 C1 2A 06 20 E5 B2
1090 D1 09 22 06 20 0B C1 ED 5B
1098 88 C1 ED 43 04 20 ED 5B BD
10A0 02 20 21 09 08 AF 23 BE 94
10A8 28 FB 7E 12 13 0B FE 09 90
10B0 20 07 AF 23 BE 20 FC 18 AB
10B8 EC 78 B1 20 E8 C9 C5 E5 58
10C0 2A 02 20 ED 5B 04 20 19 A1
10C8 E5 C1 ED 5B 06 20 EB AF 86
10D0 ED 52 EB E1 C5 E5 C1 2A 80
10D8 06 20 ED 42 22 06 20 E1 66
10E0 C1 C5 E5 2A 02 20 09 05 85
10E8 C1 D1 EB 03 ED B0 C1 18 EE
10F0 A9 2A 02 20 E5 EF 0C 00 D5
10F8 2B 01 00 00 3E 07 23 03 9F
1100 BE 20 08 ED 43 04 20 11 4F
1108 0A 08 ED 53 29 0C E1 7E FF
1110 FE 09 20 03 F7 3E 0D F7 84
1118 23 0B 78 B1 20 F1 11 CA 6C
1120 0B ED 53 29 0C EF 46 49 2F
1128 4C 45 20 4E 6F 20 00 08 CF
1130 F5 DF 68 F1 08 EF 20 20 A5
1138 20 20 43 6F 6E 74 69 6E F4
1140 75 65 20 6F 72 20 45 53 E4
1148 43 41 50 45 00 E5 DF 7B B1
1150 E1 C9 21 22 20 22 0D 20 31
1158 22 0F 20 22 13 20 21 01 31
1160 00 22 11 20 21 FF 27 22 2D
1168 15 20 21 FF 5F 22 08 20 77
1170 21 00 28 22 00 20 22 02 30
1178 20 C9 11 01 00 ED 53 04 C8
1180 20 23 22 06 20 3E C3 32 4F
1188 0A 20 21 F1 10 22 0B 20 3D
1190 EF 0C 00 41 CA 0B 22 29 32
1198 0C EF 44 41 54 41 20 49 27
11A0 4E 50 55 54 00 CD 00 10 D5
11A8 EF 0C 00 21 CA 0B 22 29 F5
11B0 0C EF 4E 65 78 74 20 46 C1
11B8 69 6C 65 20 2D 20 6F 72 51
11C0 20 45 53 43 41 50 45 20 C2
11C8 3F 00 DF 7B FE 1B 20 01 AC
11D0 C9 2A 04 20 ED 5B 02 20 62

```

```

11D8 19 22 02 20 23 22 06 20 B1
11E0 21 01 00 22 04 20 18 A8
11E8 ED 5B 06 20 1B AF E5 ED 03
11F0 52 E1 D8 EF 0C 00 11 CA E2
11F8 0B ED 53 29 0C 0C EF 45 4E 0B
1200 44 20 4F 46 20 44 41 54 04
1208 41 20 00 DF 66 EF 68 65 7C
1210 78 00 00 E1 C9 2A 00 20 22
1218 02 20 3E 01 08 CD 0A 20 8A
1220 FE 1B 20 20 01 C9 22 02 20 79
1228 CD E8 11 08 3C 27 08 18 8B
1230 EC 3A 83 0C 47 21 84 0C EF
1238 22 80 0C 2A 02 20 21 ED 8C
1240 80 0C 7E E5 21 82 0C BE AE
1248 E1 28 30 1A BE 23 20 F2 AO
1250 E5 13 1A FE 20 28 0B FE C3
1258 2F 28 07 BE 23 28 F2 E1 A4
1260 18 DC F1 05 C8 1A FE 2F 6B
1268 20 06 13 1A FE 20 20 FA 05
1270 13 1A FE 07 C8 ED 53 80 3C
1278 0C 18 CO 17 1A FE 20 C8 81
1280 FE 2F 20 F7 13 ED 53 80 A9
1288 0C 18 B0 EF 0C 00 21 CA 54
1290 0B 22 29 0C EF 53 75 63 1E
1298 68 62 65 67 72 69 66 66 E7
12A0 65 20 65 69 6E 67 65 62 A1
12A8 65 6E 20 21 20 00 2A 06 1E
12B0 20 E5 2A 08 20 E5 21 84 A3
12B8 0C 22 02 20 23 22 06 20 85
12C0 21 01 00 22 04 20 21 00 5E
12C8 0F 22 08 20 CD 00 10 3E 4B
12D0 07 32 82 0C 06 00 21 84 54
12D8 0C 23 7E FE 20 20 01 04 DA
12E0 FE 07 20 F5 78 32 83 0C 4E
12E8 E1 22 08 20 E1 22 06 20 4E
12F0 C9 CD 8B 12 2A 00 20 22 A1
12F8 02 20 3E 00 08 CD 31 12 82
1300 08 3C 27 08 78 B7 20 03 D8
1308 CD 0A 20 2A 02 20 3E 07 A3
1310 23 BE 20 FC 23 22 02 20 87
1318 ED 5B 06 20 1B AF ED 52 A2
1320 38 DB EF 0C 00 21 CA 0B 37
1328 22 29 0C EF 41 6C 6C 20 BA
1330 46 69 6C 65 73 20 53 65 0E
1338 61 72 63 68 65 64 00 C9 7B
1340 3E 0D CD 67 17 21 26 11 41
1348 06 09 DF 6D 08 47 08 78 85
1350 CB 2F CR 2F CB 2F CB 2F 4B
1358 C6 30 CD 67 17 78 E6 0F 19
1360 C6 30 CD 67 17 7E 0D CD CC
1368 67 17 2A 02 20 7E EF 07 C8
1370 23 C8 FE 09 20 02 3E 0D E2
1378 CD 67 17 18 FO 2A 02 20 2A
1380 78 3D 27 47 28 19 23 7E 98
1388 FE 07 20 08 21 67 03 06 59
1390 05 DF 6D C9 FE 20 28 04 07
1398 FE 09 20 EA 23 18 E1 7E 56
13A0 FE 09 20 02 3E 0D 23 FE 48
13A8 0A C8 FE 20 C8 CD 67 17 RE
13B0 18 ED 2A 0D 20 3E 0D CD 37
13B8 67 17 3A 17 20 47 ED 5B 49
13C0 13 20 1B AF E5 ED 52 E1 D5
13C8 30 0B 05 28 23 7E 23 FE 05
13D0 07 28 EB 18 F8 EF 46 6F B1
13D8 72 6D 20 4E 6F 2E 20 6E 63
13E0 6F 74 20 65 78 69 73 74 23
13E8 69 6E 67 00 E1 C3 1A 15 0C
13F0 7E FE 07 28 28 FE 40 28 3C
13F8 0C FE 09 20 02 3E 0D CD 58
1400 67 17 23 18 EB 23 46 CB EC
1408 20 CB 20 CB 20 23 20 23 20
1410 7E E6 0F 80 47 23 E5 CD 33
1418 7D 13 E1 18 D3 2A 02 20 D4
1420 7E FE 07 23 20 FA C9 EF 4C
1428 0C 41 55 53 47 41 42 45 40
1430 46 4F 52 4D 41 54 3A 0D 54
1438 0D 42 69 6C 64 73 63 68 12
1440 69 72 6D 20 20 20 20 20 3C
1448 20 20 20 20 20 20 20 20 5C
1450 20 20 31 0D 47 61 6E 7A 72

```

```

1458 65 20 46 69 6C 65 73 20 04
1460 64 20 75 63 20 32 20 44 9F
1468 20 20 20 6E 20 20 69 6E 43
1470 61 74 65 74 2E 46 6F 72 8F
1478 62 65 73 74 20 20 20 20 76
1488 33 0D 00 DF 7B FE 31 20 85
1490 05 21 F1 10 18 30 FE 32 43
1498 20 05 21 40 13 18 FE 43 82
14A0 33 20 E8 EF 0C 4E 55 4D DA
14A8 4D 45 52 20 64 65 73 2C 1C
14B0 46 6F 72 6D 75 6C 61 72 0C
14B8 73 3F 0D 00 DF 7B E6 0F DA
14C0 32 17 20 21 B2 13 22 0B 50
14C8 20 C9 2A 06 20 22 02 0B 59
14D0 C3 7A 11 21 DC 14 22 0B 70
14D8 20 C3 F1 12 12 CD F1 10 E5 85
14E0 FE 1B 28 2E 21 CA 0B 22 7B
14E8 29 0C EF 43 48 41 4E 47 81
14F0 45 20 21 21 F1 10 20 20 03
14F8 43 55 52 53 5F 52 74 74 7E
1500 6F 20 45 6E 64 20 6F 66 B0
1508 20 46 69 6C 65 21 0B CD AB
1510 00 10 21 F1 10 22 00 20 0B A4
1518 E1 C9 EF 0D 44 41 54 45 F1
1520 49 56 45 52 57 41 6C 54 A3
1528 55 4E 47 0D 0D 4E 45 75 69
1530 65 20 44 61 74 65 69 20 D1
1538 61 6E 6C 65 67 65 6E 20 47
1540 20 20 20 20 20 20 31 0D 53
1548 41 6C 6C 65 20 46 69 6C 16
1550 65 73 20 61 75 73 67 65 72
1558 62 65 6E 20 20 20 20 20 42
1560 20 32 0D 46 69 6C 65 20 74
1568 73 75 63 68 65 6E 20 20 43
1570 20 20 20 20 20 20 20 20 85
1578 20 20 20 20 33 0D 46 69 FC
1580 6C 65 20 61 6E 68 7B 6E A6
1588 67 65 6E 20 20 20 20 20 77
1590 20 20 20 20 20 20 20 20 34 B9
1598 0D 46 69 6C 65 20 7B 6E 43
15A0 64 65 72 6E 20 20 20 20 DE
15A8 20 20 20 20 20 20 20 20 BD
15B0 20 20 35 0D 46 6F 72 6D DB
15B8 75 6C 61 72 20 65 72 73 EB
15C0 74 65 6C 6C 65 6E 20 20 99
15C8 20 20 20 20 20 20 36 0D 46 06
15D0 6F 72 6D 75 6C 61 72 20 07
15D8 7B 6E 64 65 72 6E 20 20 BF
15E0 20 20 20 20 20 20 20 20 F5
15E8 3F 0D 41 6C 6C 65 20 46 25
15F0 67 72 6D 75 6C 61 72 65 6C
15F8 20 61 75 73 67 65 62

```

16D8 17 CD 14 12 CD 50 17 C3 EF
 16E0 1A 15 FE 30 39 C2 16 DF FC
 16E8 5B FE 30 20 3D 10 20 15 15
 16F0 22 0C 0C 2A 06 21 02 0E 00
 16F8 0C DF 66 EF 44 61 74 65 0C
 1700 69 65 6E 64 65 0D 41 62 CC
 1708 63 70 65 69 63 74 20 45 72
 1710 6E 20 6D 69 74 20 45 4E B2
 1718 54 45 52 0D 00 DF 7B FE 7F
 1720 0D 28 02 DF 5B DF 57 C3 A1
 1728 1A 15 FE 4D 20 03 C3 22 C1
 1730 19 FE 42 C2 1A 15 C3 00 54
 1738 1C 21 00 20 11 18 20 01 F6
 1740 0A 00 01 0A 00 0D ED B0 C9 FO
 1748 00 20 01 0A 00 0D ED B0 C9 FO
 1750 21 00 20 11 0D 20 01 0A F1
 1758 00 ED B0 21 18 20 11 00 76
 1760 20 01 0A 00 ED 80 C9 F5 FD
 1768 DF 6F F1 FE 0D C0 3E 0A D1
 1770 DF 6F 06 30 C3 40 00 F5 03
 1778 D5 C5 AF B8 EB 21 00 00 9C
 1780 28 05 19 38 17 10 FB C1 F8
 1788 D1 F1 C9 F5 D5 13 11 00 05
 1790 AF ED 42 38 03 13 18 F9 E4
 1798 EB D1 F1 C9 EF 0D 49 4C B6
 17A0 4C 45 47 41 4C 20 43 41 00
 17A8 52 52 59 20 45 52 52 4F 14
 17B0 52 0D 00 C9 DF 63 DF 79 89
 17B8 38 57 2A 0C 0C AF 7C E6 B1
 17C0 0F C6 F6 38 4C 7C 06 60 C8
 17C8 38 47 7D E6 0F C6 F6 38 C4
 17D0 40 7D C6 60 38 3B E5 CB ED
 17D8 3C CB 3C 60 3C CB 3C 44 84
 17E0 21 E8 03 CD 77 17 EB E1 2A
 17E8 E5 7C E6 0F 47 21 64 00 F7
 17F0 CD 77 17 19 EB E1 E5 CB E1
 17F8 3D CB 3D CB 3D CB 3D 45 A9
 1800 21 0A 00 CD 77 17 19 EB A2
 1808 E1 7D E6 0F 6F 26 00 19 21
 1810 C9 EF 0D 57 72 6F 6E 67 FA
 1818 20 49 6E 70 75 74 0D 00 6D
 1820 18 92 21 00 0F 11 84 0C B3
 1828 ED 52 3E FF 12 13 2B 7C 88
 1830 B5 20 F7 3E 07 32 00 0F 9A
 1838 11 84 0C 2A 02 20 7E 23 DE
 1840 FE 07 28 3F FE 3A 20 F6 12
 1848 23 1A FE 07 28 0E 7E 12 68
 1850 13 FE 20 28 E9 FE 09 20 D1
 1858 EF 23 18 E2 EF 0C 00 21 98
 1860 CA 0B 22 29 0C EF 45 52 2A
 1868 5A 4F 52 3A 20 44 61 74 E6
 1870 61 20 74 6F 6F 50 6C 6F 56
 1878 6E 67 0D 00 CD 50 17 E1 87
 1880 C3 1A 15 D2 C9 EF 0F 4E AF
 1888 75 6D 62 65 72 20 6F 66 B0
 1890 20 46 69 6C 65 73 3F 0D 07
 1898 00 CD B4 17 22 80 0C 21 17
 18A0 84 0C 01 00 00 7E FE 07 CC
 18A8 23 03 20 F9 ED 43 82 0C BD
 18B0 2A 00 20 22 02 20 2A 02 82
 18B8 20 09 11 FF 5F AF ED 52 56
 18C0 28 20 38 1E EF 0D 4E 75 35
 18C8 6D 62 65 72 20 6F 66 20 9B
 18D0 46 69 6C 65 73 20 74 6F DE
 18D8 6F 20 68 69 57 68 0D 00 2C
 18E0 18 A3 ED 5B 02 20 21 84 C2
 18E8 0C ED B0 ED 53 02 20 ED F8
 18F0 53 06 20 ED 4B 80 0C 0D 50
 18F8 78 B1 28 0A ED 43 80 0C 27
 1900 ED 4B 82 0C 18 B0 EF 0C A2
 1908 00 21 CA 0B 22 29 0C EF 5D
 1910 41 6C 6C 20 46 69 6C 65 E2
 1918 73 20 6C 6F 61 64 65 65 2D
 1920 00 C9 EF 0C 4D 61 73 6B 89
 1928 69 65 72 74 65 20 44 61 1F
 1930 74 65 69 20 20 49 4E 4D
 1938 49 49 45 52 45 4E 4D 20 DF
 1940 31 0D 4D 61 73 6B 69 65 F1
 1948 72 74 65 20 46 69 6C 65 4C
 1950 73 20 61 75 73 67 65 62 73

1958 65 6E 20 20 20 20 20 32 0D 03
 1960 45 49 4E 47 41 42 45 20 84
 1968 69 6E 20 6D 61 73 6B 2E 52
 1970 20 20 20 20 20 20 20 20 20 DC
 1978 20 20 20 20 20 20 20 20 20 6B
 1980 0D 0D 0D 0D 0D 0D 0D 0D 0D 01
 1988 00 DF 7B FE 31 C2 A5 19 AA
 1990 CD 52 11 CD 39 17 CD CA 8D
 1998 14 CD 22 18 CD 50 17 CD CD
 19A0 85 18 C3 22 19 FE 32 20 A4
 19A8 06 CD 6C 1A C3 22 19 FE 16
 19B0 33 C2 1A 15 CD 25 1B C3 BD
 19B8 22 19 ED 5B 0D 2C 21 0A AC
 19C0 08 22 29 0C 1A FE 3A 28 B2
 19C8 09 FE 07 C8 CD 0A 20 13 C1
 19D0 18 F2 CD 0A 20 3E 20 CD 15
 19D8 0A 20 2A 02 20 7E F5 CD A7
 19E0 0A 20 F1 FE 20 23 20 F5 6A
 19E8 22 02 20 13 13 1A FE 20 A3
 19F0 13 20 FA 1A FE 07 C8 18 35
 19F8 CB 21 8A 0B 22 29 0C EF D8
 1A00 0D 0D 0D 0D 0D 0D 0D 0D 82
 1A08 0D 0D 0D 0D 0D 0D 0D 0D 7D
 1A10 21 4A 08 22 29 0C EF 46 29
 1A18 69 6C 65 20 4E 75 6D 62 1E
 1A20 65 72 20 3F 0D 00 CD B4 FE
 1A28 17 E5 C1 2A 00 20 0B 78 CC
 1A30 B1 28 08 7E 23 FE 07 28 F9
 1A38 F5 18 F8 E5 AF ED 5B 06 39
 1A40 20 ED 52 E1 28 03 30 01 F6
 1A48 C9 21 0A 08 22 29 0C EF A4
 1A50 4E 6F 20 53 75 63 68 20 FA
 1A58 4E 75 6D 62 65 72 21 00 FC
 1A60 18 AE F5 F7 F1 FE 09 C0 E4
 1A68 3E 0D 18 F6 EF 0C 00 21 F7
 1A70 CA 0B 22 29 0C EF 4F 75 69
 1A78 74 70 75 74 20 4D 61 73 A0
 1A80 6B 65 64 20 46 69 6C 65 6E
 1A88 73 20 00 21 0A 08 22 29 B3
 1A90 0C EF 53 63 72 65 65 6E 05
 1A98 20 20 20 20 20 20 20 20 31 0D B0
 1AA0 50 72 69 6E 74 20 20 20 27
 1AA8 20 20 20 20 32 0D 00 DF 60
 1AB0 7B FE 31 20 16 21 DE 0B B4
 1AB8 22 29 0C EF 53 63 72 65 A5
 1AC0 65 6E 00 21 62 1A 22 0B 77
 1AC8 20 18 17 FE 32 20 ED 21 82
 1AD0 DE 0B 22 29 0C EF 50 72 DB
 1AD8 69 6E 74 00 21 C7 1B 22 62
 1AEO 0B 20 CD F9 19 22 02 20 48
 1AE8 CD BA 19 DF 7B FE 0D 28 2F
 1AF0 F1 FE 1B 20 F6 21 F1 10 4C
 1AF8 22 0B 20 C9 DD E5 C1 C5 70
 1B00 21 0A 08 7E 23 FE 00 28 15
 1B08 FA FE 3A 20 F6 23 0B 78 11
 1B10 B1 20 F0 C1 ED 5B 02 20 17
 1B18 0B 78 B1 C8 13 1A FE 20 7A
 1B20 20 FA 13 18 F3 EF 0C 00 6E
 1B28 21 62 1A 22 0B 20 21 CA 18
 1B30 0B 22 29 0C EF 57 6F 72 D4
 1B38 64 20 49 6E 70 75 74 20 07
 1B40 69 6E 20 4D 61 73 6B 65 43
 1B48 64 20 46 69 6C 65 20 20 A7
 1B50 00 21 0A 08 22 29 0C EF E4
 1B58 57 4F 52 44 20 4E 75 6D FF
 1B60 62 65 72 20 3F 0D 00 CD ED
 1B68 B4 17 E5 DD E1 EF 0D 0D FA
 1B70 0D 0C CD F9 19 22 02 20 BB
 1B78 E5 CD BA 19 E1 22 02 20 3D
 1B80 CD FC 1A 22 29 0C D5 DF 89
 1B88 7B D1 FE 13 28 F8 FE 14 32
 1B90 28 F4 FE 08 28 F0 FE 1B FE
 1B98 28 26 FE 0D 28 CF FE 20 21
 1BA0 20 02 3E A0 F5 F7 F1 FE 96
 1BA8 11 20 03 1B 18 D8 FE 12 12
 1BBO 20 03 13 18 D1 47 1A FE 49
 1BB8 20 28 C5 78 12 13 18 C6 5B
 1BC0 21 F1 10 22 0B 20 C9 FE 11
 1BC8 09 20 02 3E 0D CD 67 17 A4
 1BDO C9 00 00 00 00 00 00 00 B4

NACHTRAG

Falls Sie Fragen zu bestimmten Programmen haben, wenden Sie sich direkt an die Verfasser. Hier die Adressen der Mitarbeiter der Nummer 3/4-82, die wir im letzten Heft aus Platzgründen nicht mehr unterbringen konnten.

Günter Böhm, Günter Kreidl, Wolfgang Mayer-Gürr und Josef Zeller siehe Impressum.

Michael Bach
 [redacted] Stegen

Otto Föbel
 [redacted] Erlangen

Eberhard Horch
 [redacted] Hannover

Tel.?
 Rüdiger Maurer
 [redacted] Taunusstein

Klaus Mombaur
 [redacted] Nürnberg

Tel.?
 Johannes Christian Lotter
 [redacted] Darmstadt

Christian Peter
 [redacted] Wien
 Österreich

Wolfgang von Jan
 [redacted] Langenhagen

Jürgen Loh
 [redacted] Radevormwald

Wolfgang Schröder
 [redacted] Reutlingen

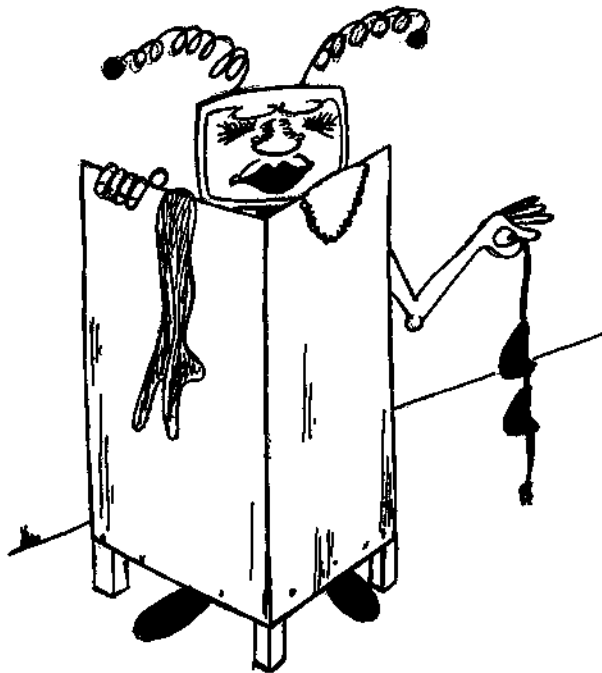
Tel.?
 Erich Mehnert
 [redacted] Mandelbachtal

Claus Stehlik
 [redacted] Judendorf
 Österreich

Tel.?
 Peter Brendel
 [redacted] Mannheim

Tel.?

NASCOMPL



Hallo liebe Leser,
diesmal möchte ich, oder besser muß ich, ein sehr heikles Thema aufgreifen: SEX UND COMPUTER. Zwei Gründe bewegen mich zu diesem Schritt. Erstens scheint ja nun der Frühling endlich Einzug zu halten, und zweitens wird dieses Thema schon seit einiger Zeit von verschiedenen Firmen in GENIEALER Werbung anvisiert. So hat doch erst kürzlich mein spanischer Kollege El Compl eine farbige Doppelseite veröffentlicht, in der durch raffinierte Kombination mit einem ziemlich sexy aussehenden Computer und einer Dame in Trauer der unbefangene Betrachter wohl zum Kauf eines kleinen Silberkreuzchens angeregt werden sollte. So geht das natürlich nicht, meine Herren! Wir Rechner lassen uns doch nicht auf so eindeutige Weise vermarkten. Da könnte ja demnächst einer hergehen und einen Rechner ohne Gehäuse abbilden. Oder denken wir gar, ~~das ist die Tastatur~~ RAM oder gar ROM ~~das ist die RAM~~. Wäre es Ihnen vielleicht recht, wenn man Ihre ~~RAM~~ Datenbus ~~RAM~~ oder vielleicht sogar ~~RAM~~ Lüfter ~~RAM~~, müssen wir uns gemeinsam zur Wehr setzen.

In diesem Sinne Ihr NASCOMPL
(Der Artikel wurde zensiert MKS)

IMPRESSUM

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

RESSORTS:

MASCHINENPROGRAMME:

Günter Böhm,
Karlsruhe, Tel.
Günter Kreidl,
Tel.
Straelen

BASIC und FLOPPY:

Wolfgang Mayer-Gürr,
Recklinghausen
Tel.
Recklinghausen

HARDWARE:

Josef Zeller,
Neu-Ulm
VERLAG: NASCOM JOURNAL, c/o MK-Systemtechnik
Pater-Mayer-Str.6, 6728 Germersheim
Tel. 07274/2756 Telex 453500 mksd

VERTRIEB: Direktvertrieb durch den Verlag
Erscheinungsweise: monatlich

Bezugspreis: Im In- und Ausland 48.- für ein Jahresabonnement. Abonnements können aus technischen Gründen immer nur für die Dauer eines Kalenderjahres, d.h. vom 1.1. bis 31.12. laufen. Bei Bestellung nach dem 1.1. werden die fehlenden Hefte mit der ersten Lieferung bis zum Bestellzeitpunkt automatisch mitgeliefert. Bei nicht fristgerechter Kündigung verlängert sich das Abonnement automatisch um ein Jahr. Die Kündigung für das Folgejahr muß bis spätestens sechs Wochen vor Jahresende erfolgen.

Bezugsmöglichkeiten: Durch Bestellung bei MK Systemtechnik.

Bankverbindungen: Alle Zahlungen für das NASCOM JOURNAL unter Angabe der Rechnungsnummer an MK - Systemtechnik, Germersheim,

Zahlung: Nach Eingang Ihrer Bestellung erhalten Sie von uns die ausstehenden Hefte bis zur aktuellsten Ausgabe sowie eine Rechnung. Bitte, zahlen Sie dann den Rechnungsbetrag.

Bitte keine Vorauszahlungen!

Bitte, Anfragen wegen Abonnements oder Lieferung nicht an die Redaktion sondern nur an den Verlag. Die Autoren tragen die Verantwortung für ihre Beiträge selbst. Für Fehler in Text, Bildern und sonstigen Angaben kann keine Haftung übernommen werden.

nascom

Die Alternative!

Kein »langweiliger Computer«

NASCOM 1 und NASCOM 2 sind Computer für Selbsterbauer, Tüftler, erfolgreiche Do-it-yourself-Freunde. NASCOM-Computer werden niemals langweilig! Die Systeme 1 und 2 sind keine fertigen »Kästen« ohne Erweiterungsmöglichkeit. Der hochwertige Platinsatz Computer und Keyboard kann so aufgebaut, erweitert und »verpackt« werden, wie Sie es wünschen.

Für Vollpreis gibt die NASCOMs auch als Bausatz. Aber aufgepaßt: Das ist eine Sache nur für wirkliche Köpfer! Und damit es auch nach dem Aufbauen nicht langweilig wird, gibt es das monatlich erscheinende NASCOM-JOURNAL. Eine Zeitschrift speziell für NASCOM-Freaks vollgestopft mit Hardware- und Software-Ideen, Kleinanzeigen, den neuesten Infos, und, und, und. . .

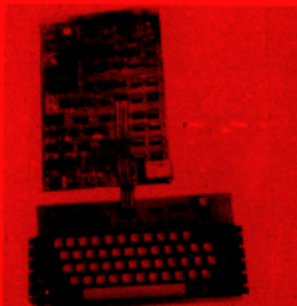
Die NASCOMs sind keine »Spielcomputer«. Mehr als 60% aller NASCOM-Systeme werden als sogenannte

»OEM-Baugruppen« von professionellen Anwendern in eigene Systeme eingebaut. Ingenieurbüros verwenden den NASCOM als Entwicklungssystem. Die Anwendungsmöglichkeiten sind mehr durch Ihre Phantasie begrenzt. Ein NASCOM-System kann fast alle gängigen Probleme lösen.

Mit NASCOM wachsen!

NASCOM-Systeme sind aufwärtskompatibel. Das kleinste, preisgünstigste NASCOM 1-System kann bis auf NASCOM 3-Level mit Floppy-Laufwerken und CP/M* ausgebaut werden. Bildschirm-Aussteuerung, Tastatur Betriebssystem und Systemsoftware sind durchweg kompatibel. Ohne faule Kompromisse!

Mit NASCOM-Systemen gehen Sie kein Risiko ein. Ihr NASCOM wächst mit!



NASCOM 1

Spezifikationen:

- QWERTY-Tastatur, aufgebaut mit hochwertigen Magnetstaben
- NAS-SYS Betriebssystem (2k Byte)
- 16 I/O-Leitungen
- Video (BAS) und TV-Ausgang
- 1k RAM, ausbaubar auf 192k RAM
- Display 48 Zeichen in 16 Zeilen

ab DM 938,-



NASCOM 2

Spezifikationen:

- Wie NASCOM 1, jedoch zusätzl.:
- 8k Mikrosort-BASIC u. 8k Stat. RAM
 - Z80A-Mikroprozessor, 4 MHz
 - Erweiterte Tastatur 57 Tasten
 - Integrierte Bus-Pufferung
 - Bis 192k Byte RAM
 - Grafik-Möglichkeiten: 48 x 96 Punkte
 - Serielle Schnittstelle, Baudrate wählbar, RS232C/20mA
 - 16 parallele Ein/Ausgabeleitungen (Z80APIO)

ab DM 1950,-

nascom 3 — der Profi



Spezifikationen: Wie NASCOM 2, jedoch zusätzl.:

- 0.35 Megabyte pro 5.25-Zoll Laufwerk
- Betriebssystem CP/M* 2.2 oder NAS-DOS
- Bildschirmausgabe erweiterbar auf 80 x 25 Zeichen

ab DM 2735,-

Die dritte NASCOM-Generation

NASCOM 1 und 2 haben OEM-Board, Schulungscomputer, Kompaktrechner etc. ca. 20 000 mal Ihren Partner gefunden. Der NASCOM 3 möchte Ihr persönlicher Computer werden! Er möchte Ihnen helfen, sich selbst fortzubilden, im Beruf weiter zu kommen, auch mal in die Computertechnik »rein zu riechen«. Ingenieurbüros und Softwareingenieuren dient der NASCOM 3 als preisgünstiges Entwicklungssystem.

Universelle Betriebssoftware

Der NASCOM 3 kennt zwei Betriebssysteme: Das CP/M* (Version 2.2) — inzwischen Standard — und sein eigenes NAS-DOS. Die 5-Zoll Floppys bieten eine Speicherkapazität von 0.35 Megabyte pro Laufwerk (single sided, double density, double tracked). Damit wird das Spektrum universeller CP/M*-Software verfügbar!

**Wir informieren Sie unverbindlich:
Fordern Sie Ihr NASCOM-INFO-PAKET an! ****

Unsere Händler:

Heinz Vogel Verlag GmbH & Co.
Lehrmittelzentrum Herr Seloff
Innsbrucker Straße 96
2800 Bremen-Findorff
☎ (0421) 35 10 69

Christian Lampson
W.-Leuschner-Straße 4
6085 Nauheim
☎ (061 52) 567 30

MK-SYSTEMTECHNIK
Michael von Keitz
Plaffenberg 4
5650 Solingen
☎ (021 22) 4 72 67

MK-SYSTEMTECHNIK
Kriegstraße 164
7500 Karlsruhe
☎ (07 21) 2 92 43

Radio Zinburg
Herr Zinburg, Jr.
Röhstraße 10
5760 Arnsberg
☎ (029 32) 3 45 10

Graf Elektronik Systeme GmbH
Postfach 1610
8791 Kempten
☎ (08 31) 6 19 30

Autorisierter Distributor:

MK-SYSTEMTECHNIK
Pater-Mayer-Straße 6
6728 Germersheim
☎ (072 74) 20 93
Telex 453500 mks d

CP/M* ist ein eingetragenes Warenzeichen der DIGITAL RESEARCH

** NASCOM-INFO-PAKET gegen DM 2,- in Briefmarken (wird bei Kauf angerechnet)