

Einzelpreis DM 15,-

3. Jahrgang

Mikro

Das Profi-Magazin für alle Anwender von MZ-700/800

MIKA

Mein Neffe Mika schlachtete sein Sparschwein und ordnete das Geld zu Häufchen. Es waren je die gleiche Anzahl von Stücken zu Einpfennig, Zweipfennig, Fünfpfennig, Zehnpfennig und Fünfzigpfennig. Der Gesamtwert betrug 5,44 DM. Ich sollte ohne körperliches Zählen die Menge der Münzen der fünf Haufen bestimmen. Spaßeshalber schlug ich zwei parallele Wege ein. Der Erste war der rein algebraische, der Zweite ein teilnumerischer Weg mit einem Basicprogramm.

$$5.44 = 0.01 * X + 0.02 * X + 0.05 * X + 0.1 * X + 0.5 * X$$

$$5.44 = (0.01 + 0.02 + 0.05 * 0.1 + 0.5) * X$$

$$5.44 = 0.68 * X$$

$$X = 5.44 / 0.68$$

$$X = 8$$

$$M = 8 * 5$$

$$M = 40$$

1) Die algebraische Lösung

```

10 X=1
20 Y=.01*X+.02*X+.05*X+.1*X+.5*X
30 PRINT X;" ; " ; Y;" // " ;
40 IF Y=5.44 GOTO 60
50 X=X+1:GOTO 20
60 M=X*5
70 PRINT "M=" ; M
80 END

```

2) Die Basicprogramlösung mit MZ-2Z046

Das Erstaunen war groß. Die Abbruchbedingung Y=5.44 wurde erreicht, ein selbstständiger Abbruch erfolgte nicht. Er wurde dann von Hand eingeleitet.

1; .60// 2; 1.36// 3; 2.04// 4; 2.72// 5; 3.4// 6; 4.08// 7; 4.76// 8; 5.44// 9; 6.12// BREAK

Nach dem Umschreiben der Zeile 20 arbeitete das Programm korrekt:

```

20 Y=.5*X+.1*X+.05*X+.02*X+.01*X

```

Editorial

Nur geringe Resonanz fanden die Aufrufe in der Ausgabe 3+4/89 des Magazins. Zur Erinnerung: Wir baten im Editorial, die Meinungen der Leser zum Magazin und zu den behandelten Themen mitzuteilen und in einem Artikel, die Ansichten zur geplanten Technischen Bibliothek kundzutun.

Ein Einsender regte die Programmierung der Farbdarstellung des Basic für 256 Farben auf Maschinenebene an und sandte einen eigenen Entwurf in Basic, den wir abdrucken. Ob eine Realisierung in ML möglich ist, wollen wir gerne prüfen. Haben die Leser eine Idee dazu? Oder sogar bereits eine Lösung?

Für die technische Bibliothek ging die Beschreibung der Universal-Interface-Card MZ-80-102 in deutscher Sprache und der Schaltplan bei uns ein. Ein Leser bot die Unterlagen der RAM-Card an. Warum sendet er uns diese denn nicht gleich zu? Wir bitten darum. Bei der Durchsicht der Altunterlagen eines Kameraden fanden wir z.B. den Schaltplan der Hauptplatine und das Platinenlayout. Mehr über die Bibliothek folgt.

Was ist der Grund für die geringe Rückwirkung auf unsere Aufrufe? Selbst die vollkommene Zufriedenheit aller Leser erklärt diesen Umstand nicht. (Eine vollständige Zufriedenheit ist für den Autor nicht vorstellbar, auch nicht wünschenswert. Der Antrieb ginge verloren, Stillstand und Rückschritt wären die Folgen. Die Artikel im Magazin sollen aber auch dem Fortschritt dienen.) Interessiert denn niemanden das Aufzeichnungsverfahren des Kassettenrecorders und dessen Programmierung auf Maschinenebene oder eine Methode zur Darstellung in Basic von beweglichen Figuren im Vordergrund vor einem Hintergrundbild? Oder die Nutzung des Gauß-Rechenverfahrens zur Lösung von Gleichungen vom Geschlecht 3 oder größer? Anspruchs-

Impressum

Redaktion: Harald Schicke, Edgar Lefgrün
Mitarbeiter dieser Ausgabe: Hans-Peter Auel, Hans Werner und Bernd Birkenbach, Martin Bommel, Oliver Braun, Dirk Grube, Lars Hanke, Edgar Lefgrün, Günter Pudritz, Harald Schicke, Karsten Schulte, Jan Seng

Anzeigen: Harald Schicke

Vertrieb: MZ-Verlag

Druck: WEMCARD GmbH

Verlag:

MZ-Verlag Harald Schicke, Postfach
(für Pakete: Lindenweg 18),
D-2110 Buchholz 5

☎ 0 41 87/65 33

Telex: 051933521 dmbx g

ref: box:dm4:mz-verlag

BTX: 041876533

Magazin 700-800 ist eine unabhängige Zeitschrift und nicht SHARP Electronics angegliedert.

Magazin 700-800 erscheint sechs mal im Jahr. Der Einzelpreis beträgt DM 7,50. Im Abonnement kostet es DM 36,- pro Jahr (Ausland DM 42,-). Das Abonnement gilt grundsätzlich für ein Kalenderjahr und verlängert sich automatisch um ein Jahr, wenn es nicht bis sechs Wochen vor Ende des Kalenderjahres gekündigt wird.

Für unverlangt eingereichte Manuskripte und Fotos übernimmt der Verlag keine Haftung. Warennamen werden ohne Gewährleistung der freien Verwendbarkeit benutzt. Gerichtsstand ist Tostedt.

Manuskripte sind an den MZ-Verlag Harald Schicke, Postfach, D-2110 Buchholz 5 zu senden. Für den Inhalt namentlich gekennzeichnete Beiträge tragen die Autoren die Verantwortung. Mit dem Abdruck erwirbt der Verlag das ausschließliche Recht der Vervielfältigung, auch auf Tonträgern, und die Rechte sonstiger Wiedergabemöglichkeiten, z.B. fotomechanisch, auf Mikrofilm, auf Datenträgern usw., ebenso das Recht der Übersetzung in fremde Sprachen und das Recht der Veröffentlichung im In- und Ausland. Autoren erhalten ein kostenloses Belegexemplar (ab Beiträgen von mindestens einer Seite Länge) sowie ein Honorar von DM 25,- pro Seite.

ISSN 0931-8860

Mit kameradschaftlichen Grüß

Die Redaktion und
Ihr Edgar Lefgrün

Edgar Lefgrün

volle Themen gibt es doch in „Hülle und Fülle“. Sind Themen dieser Art zu schwierig oder nicht interessant genug. Wird Einfacheres gewünscht? Liebe Leserin, lieber Leser, ich wiederhole die Frage aus Heft 2/89. Was soll es denn bitte sein?

Der Schwerpunkt dieser Ausgabe ist die Textverarbeitung mit WORDSTAR. Für Blindschreibfähige ist die ASCII-Uralt-Tastatur des MZ-800 doch wohl ein Schreckenskabinett. Eine weitmögliche Anpassung an eine deutsche Maschinentastatur ist also wünschenswert. Sie wird im BIOS des CP/M durchgeführt. Auch die Druckeranpassung bereitet vielen Wordstar-Fans Schwierigkeiten. Die vorhandene Literatur wurde gesichtet und eine leichter verständliche Zusammenfassung als im Programmhandbuch erstellt. Auf einige weitgehend unbekannt Zusammenhänge wird dabei genauer eingegangen.

Freundliche Leserin, werter Leser, wir wünschen Ihnen Erfolg und Vergnügen beim Lesen dieser Ausgabe des Magazins.

Nachdem wir gut ein halbes Jahr auf Belleferung mit QDs durch die Firma SHARP vergeblich gewartet haben, müssen wir den Vertrieb von QDs einstellen. Es sieht so aus, als wenn SHARP die Versorgung damit nicht weiter aufrecht erhalten kann.

Es wird deshalb in Zukunft auch keine **Frei-Soft** oder sonstige Software mehr auf QD geben. Wir haben jedoch noch einen Vorrat an einmal bespielten QDs, die wir jetzt zum Preis der neuen QDs abgeben (DM 8,60/Stück). Mit ein wenig Glück können Sie also sogar noch Programme kostenlos dazu erhalten.

QDs liefern wir nur noch per Nachfrage solange der Vorrat reicht.

WordStar, **Dbase II**, **Multiplan** etc. sind als Originalsoftware unseres Wissens für den MZ-800 bzw. für CP/M-Rechner nicht mehr erhältlich. Die deutschen Distributoren dieser Programme wissen manchmal nicht mal mehr, was CP/M überhaupt ist.

Wir haben uns in unendlich vielen Telefonaten darum bemüht, diese Software, die nach wie vor sehr wertvoll ist, irgendwo zu erhalten.

Hier hilft sicherlich nur der Gebrauchtsoftwaremarkt. Vielleicht geben Sie eine Anzeige im Magazin auf?!

Die letzte Nummer wird, ebenfalls als Doppelausgabe, im November erscheinen. Nach drei Jahren im Dienste der MZ-Anwender werden wir dann das Magazin 700/800 einstellen müssen.

Der Grund ist ganz einfach: Knapp 50% aller Abonnenten haben sich bis zu sieben Monate Zeit gelassen, um ihr Abo zu bezahlen. 20% haben es überhaupt nicht bezahlt.

Wir sind damit tief in die roten Zahlen gerutscht - rentiert hat es sich eh

nie, auch wenn das viele immer noch nicht glauben wollen.

Es ist ja nicht damit getan, ab und an ein Heft herauszubringen. Täglich kommen viele Briefe und Anrufe mit Fragen, Bitten etc. Jeder erwartet - durchaus zu Recht - eine Antwort.

Aber das ist für einen Verlag sehr teuer.

Das Magazin könnte durchaus weiterleben, wenn es von jemandem gestaltet werden würde, der dies aus Hobby macht. Er muß nicht unbedingt etwas verdienen und seine Anhänger wissen, daß er/sie es hauptsächlich aus Vergnügen tut. Bei eventuellen Wartezeiten wird dann nicht gleich die vermeintliche Profitgier angeprangert.

Wer also Lust und Möglichkeiten hat, das Magazin weiterzuführen, der mag sich bei uns melden.

Wir selber prüfen auch, ob evtl. eine Vereinsgründung Möglichkeiten bietet, Ihnen weiterhin dienstbar zu sein.

Ansonsten werden wir noch bis zum Jahresende für Sie tätig sein. Wenn Sie noch etwas benötigen, sollten Sie es also bald bestellen.

Eins ist gewiß - Magazin hin oder her - der MZ-700/800 ist immer noch genauso gut wie einst. Zwingende Gründe zum Umsteigen gibt es also nicht.

Die gute Nachricht zum Abschluß:

Unser **Katalog** über Frei-Programme wird nun endlich in Kürze erscheinen. Wenn Sie ihn haben wollen, schicken Sie uns bitte DM 1,40 in Briefmarken.

Ihr MZ-Verlag Harald Schicke

Editorial	2
Wichtige Informationen	3
Inhalt	3
Sharp-BASIC/MBASIC (CP/M) und GWBASIC (MS-DOS)	4
Prüfprogramm: Mikro-CAD	11
64K-Static-RAM Karte für den MZ-800	12
Einführung in die Grafik des MZ-800	14
Listings von G. Pudritz	21
Listing von O. Braun	23
Listings von Jan Seng	24
Aufnahme von Daten mit der Schnittstelle RS 232C	25
WordStar anpassen	27
Listing von H-P Auel	36
Listing von K. Schulte	38
Kleinanzeigen	39
Fehlerkorrektur	39
MZ-800 Poster	40

MZ-800 Poster

Das auf Seite 40 klein und s/w abgedruckte Poster haben wir vor vier Jahren entwerfen und drucken lassen. Im Original hat es die Größe DIN A2 und ist natürlich farbig. Unsere Druckerei hat noch einen Stapel gefunden. Das Poster sollte bei keinem MZ-Fan fehlen, da es außerordentlich dekorativ ist. Wir liefern, so lange der Vorrat reicht. Für 1 Stück senden Sie bitte DM 2,40 in Briefmarken, für 3 Stück DM 5,00, für 10 Stück DM 10,00.

Sharp-BASIC / MBASIC (CP/M) und GW-BASIC (MS-DOS)

Sharp-BASIC / MBASIC (CP/M) und GW-BASIC (MS-DOS)

Vergleich und Transfer

1. Das Problem

Nach Erscheinen meiner Beiträge zum Thema Datentransfer in den Heften 4 und 5/88 habe ich sehr viele Anrufe und Zuschriften mit zusätzlichen Fragen erhalten. Diese will ich hier noch einmal aufgreifen und hoffentlich auch beantworten. Neben der Technik soll dabei auch auf Sinn und Nutzen der Übertragung eingegangen werden.

Der Computer-Anfänger wird bei seinen ersten Programmierversuchen mit BASIC anfangen. Das ist auch bei den sogenannten „Clean-Computern“ der MZ-Serie so, die ja nach dem Einschalten nur über den ROM-Monitor und keine Programmiersprache verfügen. Nach kurzer Zeit wachsen die Programmhalden auf den Disketten stark an, zumal durch Aufgabe der Produktion der MZ-Serie viele Programme heute frei verfügbar sind.

Wer mal mit C64-BASIC gearbeitet hat (SIMON'S ist nicht viel besser), wird vom Sharp-BASIC begeistert sein, sei es Kassetten- oder Disketten-BASIC mit allen Varianten.

Daneben gibt es noch weitere BASIC-Implementationen, die – z.T. unberechtigt – ein Kümmerdasein führen, z.B. das Microsoft-BASIC (MBASIC) unter Betriebssystem CP/M.

Viele, die ich befragt habe, scheuen sich, MBASIC zu benutzen. Als Gründe dafür werden immer wieder genannt:

1. fehlende Grafikmöglichkeiten (und Sound)
2. äußerst umständlicher Editor
3. keine Transfer-Möglichkeit von oder nach Sharp-BASIC

Genau diese Gründe waren auch bei mir ausschlaggebend, MBASIC früher kaum zu benutzen. Das hat sich mittlerweile geändert. Die Gründe dafür werden deutlich durch den näheren Vergleich einiger Merkmale.

Vorausschickend sei gesagt, daß ich beruflich einen sehr großen Datenbestand mit immer wiederkehrenden Fragestellungen bearbeiten muß. Dabei wurden Daten erzeugt oder gemessen mit dem MZ-700, MZ-800 und einem AT-kompatiblen IBM-PC. Dazu wurde PASCAL oder BASIC benutzt, alle drei Rechner sind an teilweise unterschiedlichen Orten nebeneinander im Einsatz, der MZ-700 ausschließlich als Meßrechner.

Da es zu aufwendig wäre, alle Programme neu zu schreiben, insbesondere die Grafikteile, bin ich also stets gezwungen, die Systeme nebeneinander zu benutzen und die Daten zu überführen. Bei der heutigen schnellen Entwicklung der Rechnertechnik übrigens eine ganz normale Situation.

Um es vorwegzunehmen: Es ist ohne weiteres möglich, alle Programme und Daten in das jeweils andere System zu übertragen, Sharp, CP/M und auch MS-DOS. Wie das geht, werden wir sehen.

Auf MS-DOS soll deswegen eingegangen werden, weil sich dadurch einige interessante Möglichkeiten ergeben, die Syntax von GW-BASIC gleich MBASIC ist, und weil MS-DOS heute eben mal zum Standard geworden ist.

2. Systemvergleich

Sharp-BASIC auf dem MZ-800 verfügt in der Tat über einen mächtigen Befehlsumfang für Grafik auf dem Bildschirm. Auch die Ton- und Musikfunktionen sind vielfältig. Das DEMO-Programm „OPENING“ zeigt beides sehr eindrucksvoll.

MBASIC hat demgegenüber im Befehlsvorrat keinerlei Grafik- oder Musikbefehle. Die Hardware-Voraussetzungen der Home-Computer waren vor allem bei den VIDEO-Bausteinen und Tongeneratoren sehr unterschiedlich und nicht unter einen Hut zu bringen. So mußte sich CP/M darauf beschränken, was alle Computer gemeinsam zu leisten imstande waren. Im Prinzip orientierte man sich an den leistungsschwächsten Systemen.

Aber in der Regel läuft eben jedes CP/M-Programm nach einer einmaligen Installation auf jedem CP/M-fähigen Rechner problemlos ab. Nur das BIOS ist überall anders, aber davon merkt der Benutzer normalerweise nichts.

Sicherlich wäre es möglich, auch Grafik-Bausteine in MBASIC zu integrieren, das wäre jedoch umständlich und ohne intime Kenntnisse von Maschinensprache und Rechnerarchitektur nicht realisierbar.

Ein weiterer Nachteil von MBASIC ist der sehr umständliche Editor. Viele Sharp-Benutzer wissen gar nicht, was ein Editor ist. Wohl deswegen nicht, weil er so gut funktioniert. Unter MZ-BASIC ist LISTen und EDITieren dasselbe. Auf dem ganzen Bildschirm kann man nach Herzenslust herumfahren, ändern löschen, nach RETURN ist alles akzeptiert. Weiterhin kann man sich beim Abtippen Arbeit ersparen, indem man ähnliche Zeilen einfach nochmals LISTet, mit der neuen Zeilen-Nummer versieht und nur die Abweichungen nachbessert. Geht bei MBASIC alles nicht, bei GW-BASIC selbstverständlich, dieser Editor ist noch besser als der von Sharp.

Wenn wir also das Argument des Transfers, der einigen Probleme bereitet einmal kurz beiseite lassen, stellt sich die Frage, warum also soll man MBASIC überhaupt verwenden? Dazu muß aufgezeigt wer-

Sharp-BASIC / MBASIC [CP/M] und GW-BASIC [MS-DOS]

den, wo die Nachteile des Sharp-BASIC liegen:

1. MZ-BASIC ist schreibweisentolerant. Es ist unerheblich, ob es heißt „PRINTUSING“ oder „PRINT USING“, „INPUTA“ oder „INPUT A“. Früher habe ich BASIC-Programme ohne Zwischenräume geschrieben, weil es sich schneller tippt und Speicherplatz spart. Das rächt sich aber, wenn man ein anderes BASIC benutzt oder die alten Programme dort nutzen will. Die Sharp-Syntax in der kurzen Schreibweise wird sofort angemerkert. Leider gibt es für BASIC keine Norm, jede Firma kocht(e) ihr Süppchen. Also wird schön mit SPACE geschrieben, Programme werden auch übersichtlicher.

2. Die unter Sharp-BASIC erzeugten (ASCII- oder BSD/BRD-) Dateien sind zu nichts kompatibel. Sie haben zwar, je nach verwendetem MZ-BASIC den Filecode 03 oder 04, das ist aber immer noch kein richtiger ASCII-Standard! Man schaue sich bloß einmal die LINE-FEEDS (0A) an (Floppy ausschalten, RESET, Speicher ansehen mit M und DUMP). Bei dem Versuch, einen solchen File in MBASIC, Turbo-PASCAL, Mix-C oder WORDSTAR zu laden, wird man zur Verzweiflung kommen. In der Regel reißt die Datei ab. Solche Dinge wird man nachbessern müssen, wenn ein Transfer beabsichtigt ist. Damit sei nochmals ein Riesenvorteil von MBASIC genannt:

Eine mit diesem Interpreter erzeugte Datei kann ohne besondere Maßnahmen sofort von allen CP/M-Programmen, von denen ich oben nur wenige genannt habe, gelesen und weiterbearbeitet werden. Wer also beispielsweise langsam auf PASCAL umsteigen möchte, kann alles weiterverwenden. Umgekehrt natürlich auch. Welche Anwendungsbreite!

3. Der Sharp-Zeichensatz entspricht nur bei den Großbuchstaben, den Zahlen und einigen Zeichen dem

ASCII-Standard. Die Bezeichnung im Handbuch ist also übertrieben und nicht korrekt.

Dieser Mißstand führt vor allem zu erheblichen Druckerproblemen, wenn aus verschiedenen Gründen nicht das BASIC MZ-22046 benutzt werden kann mit dem Befehl „INIT LPT:MO,S2“.

Ein nach MBASIC übertragenes Programm wird zwar nach den erforderlichen Nacharbeiten der Befehlssyntax auch mit den falschen Kleinbuchstaben laufen, aber ärgerlich ist die Sache doch. Deshalb habe ich mein Programm SHPASC.PAS nochmals überarbeitet, das uns das Umschreiben abnimmt.

4. Gegenüber MBASIC fehlen dem Sharp-BASIC einige sehr wichtige Befehle, die für den 'Hausgebrauch' auf den ersten Blick entbehrlich erscheinen mögen, für etliche Anwendungen jedoch hilfreich oder gar unverzichtbar sind.

- DEFDBL. Wohl die entscheidendste Funktion. Damit sich Variable mit doppelter Präzision definieren. Das Sharp BASIC ist mit seinen nur acht Nachkommastellen sehr ungenau. So läuft z.B. mein Astronomie-Programm zur Berechnung von Sonnen-, Mond-, Planeten- und Sternkonstellationen, sowie Finsternissen läuft nur mit MBASIC korrekt. Doppelte Genauigkeit ist überall dort erforderlich, wo Ergebnisse nicht durch mathematische Funktionen im klassischen Sinne, sondern durch Algorithmen mit hoher Schleifen-Durchlaufzahl errechnet werden. Es gibt noch Hudson-BASIC für den MZ-700 mit dieser Möglichkeit. Sonst sehr leistungsstark ist es leider ist es nur kassettensorientiert und verfügt nicht über LOAD/SAVE,A-Funktion.

- DEFINT bietet die Möglichkeit, Variablen als INTEGER zu definieren. Programme laufen dadurch, etwa

bei FOR-NEXT-Schleifen schneller ab. Man hat bei Sharp darauf verzichtet, weil es ohnehin sehr schnell arbeitet (+ Vergleichstest). Bei bestimmten Programmen kann es jedoch vorkommen, daß MBASIC Sharp auf diese Weise überholt.

Es gibt noch mehr solcher Vereinbarungen, es soll dazu nur noch erwähnt werden, daß man auch einen Typ in den anderen überführen kann und zwischen globaler und lokaler Definition wählen kann.

- INPUT hier ist die Fehlerkorrektur standardmäßig so eingebaut, daß bei Fehleingabe mit 'Redo from Start' erneut abgefragt wird. Die ERROR-Befehle stehen daneben auch noch zur Verfügung.

- LSET erweiterte Möglichkeiten der Feldbearbeitung

Das waren nur einige wenige Beispiele, die Reihe ließe sich noch fortsetzen. Einige aus meiner Sicht wichtige nicht bei MZ-BASIC verfügbaren sind genannt. Dabei geht es mir nicht darum, das Sharp-BASIC schlecht zu machen. Etliche Funktionen lassen sich auch bei Sharp realisieren, wenn auch umständlicher mit Hilfskonstruktionen. Bei Grafik, Sound und Bedienbarkeit ist es Spitzenklasse. Die mathematischen Funktionen kommen zu kurz.

Ein weiterer wesentlicher Vorteil von MBASIC ist die Aufwärts-Kompatibilität zu GW-BASIC unter MS-DOS und dem dazu gehörigen Compiler Quick-BASIC. Das ist nicht verwunderlich, denn alle drei stammen von MicroSoft.

Damit läuft ein ehemaliges Sharp-BASIC-Programm mit ein wenig Nacharbeit auf einem PC mit atemberaubender Geschwindigkeit ab. Wirklich eindrucksvoll! (+ Zeittest)

Auch zum MBASIC gibt es einen Compiler. Da ich das Paket leider nicht besitze, kann ich darüber

Sharp-BASIC / MBASIC [CP/M] und GW-BASIC [MS-DOS]

auch keine Aussage machen. Sollte einer der Leser darüber verfügen, bitte ich ihn, mit mir Verbindung aufzunehmen.

Zum S-BASIC gibt es ebenfalls einen Compiler von BBG, der aber nicht in der Lage ist, die Diskette anzusprechen. Welch ein Mangel, da er sonst doch recht schnell ist.

Zu diesem Compiler schnell noch ein kleiner Tip: Will man diskettenorientierte Programme austesten, ist es durchaus sinnvoll, sie über den Compiler laufen zu lassen. Vorher muß eventuell der Filecode des Programms in 05 abgeändert werden. Zwar werden alle Disketten-Befehle angemeckert, aber der zusätzliche Syntax-Check ist recht nützlich. Besonders bei Programmen mit sehr vielen bedingten Verzweigungen, bei denen einzelne Schleifen erst nach jahrelanger Benutzung vielleicht mal angesprungen werden. Der Compiler prüft auch sie vor einem Programmlauf durch.

3. Laufzeit-Test

Die Gegenüberstellung der BASIC-Varianten wäre nicht vollständig ohne einen Laufzeit-Test. Dazu gibt es in der EDV verschiedene Verfahren. Sie sind aber z.T. etwas praxisfremd für den Normalverbraucher, weil sie in der Regel auf mathematischen Extremfällen beruhen.

Aus der Mottenkiste meiner eigenen Lernphase habe ich vor Jahren mal aus dem Buch von G. O. HAMANN 'BASIC Schritt für Schritt mit dem MZ-700' das Schachprogramm eingetippt. Bei etwa 750 Zeilen geht das nicht ohne Fehler ab. Schließlich lief es dann doch. Wenig komfortabel und mit hohen Rechenzeiten behaftet, legte ich es bald weg. Bei solchen komplexen Problemen sind sie allerdings auch nicht anders zu erwarten.

Dieses Programm habe ich also für den Laufzeit-Test wieder ausgegraben. Dabei habe ich vor dem Start

noch weitere Überraschungen erlebt:

Der S-BASIC-Compiler mahnte einige Syntax-Errors an, die der Interpreter nicht bemerkt hatte, weil er nicht immer alle Programmteile braucht. Auf dem Quick-BASIC-Compiler des AT-kompatiblen PC stellten sich noch wesentlich mehr Fehler heraus, darunter auch ein fehlendes NEXT einer Schleife. Also prüft der S-BASIC-Compiler die Syntax nur oberflächlich zeilenweise, nicht aber die Programmstruktur wie Quick.

Erster Durchlauf ohne Integer-Ver einbarung. Es wurde die Rechenzeit bis zum ersten Zugvorschlag gemessen.

S-BASIC	S-COMPILER	MBASIC	BASCOM	GW-BASIC	QUICK-COMP
10'30''	2'12''	18'50''	nicht getestet	1'59''	14''

Nachdem alle Variablen in diesem Falle möglich-als ganzzahlig vereinbart wurden, ergab der zweite Durchlauf:

S-BASIC	S-COMPILER	MBASIC	BASCOM	GW-BASIC	QUICK-COMP
nicht möglich	nicht möglich	16'30''	nicht getestet	1'33''	11''

4. Programm-Transfer

Einige Möglichkeiten der Übertragung zwischen verschiedenen Systemen hatte ich in den Heften 4 und 5/88 bereits beschrieben.

Als 'Tor zur Außenwelt' ist eigentlich die serielle Schnittstelle nach RS232C unverzichtbar. Mit ihr ist es auch einfacher, das System zu verlassen, bzw. Dateien von außen zu importieren.

Wer keine serielle Schnittstelle besitzt, muß sich ein Zusatzprogramm beschaffen, das es erlaubt, auch unter CP/M die Kassette anzusprechen. CMT.COM tut dies, genauer es bitte in den Artikeln nachlesen.

Wenn wir uns noch einmal erinnern, besteht das grundsätzliche Problem der Inkompatibilität Sharp und CP/M vor allem an der unterschiedlichen Speicher-Organisation.

Bei Sharp liegt das BASIC „unten“, der Programmbereich liegt je nach verwendetem BASIC irgendwo bei 8000H. Dagegen beginnt unter CP/M die TPA bei 100H, die Programme liegen „oben“. Ohne Speicherverschiebungen geht da nichts, bei längeren Dateien wird man scheitern.

Also braucht man einen 'Zwischenspeicher'. Das kann ein Tonbandgerät oder ein anderer Rechner sein.

4.1 Tonbandgerät als Zwischenspeicher

Wer einen Akustikkoppler hat, ist fein raus. Man braucht keine galva-

nische Verbindung zum TB-Gerät, sondern es genügt, den Lautsprecher des Akustikkopplers in die Nähe des (eingebauten) Mikrofons zu bringen. Bei mir tut's der einfachste Kassettenrecorder im Hause ohne Probleme.

4.2 Senden eines SHARP-Programms

Zunächst muß das zu übertragende Programm geladen werden.

LOAD „FDx:name“

Es folgt die Schnittstellen-Initialisierung, mit der viele Anfänger große Probleme haben, da sie im Handbuch nur ganz kurz gestreift wird.

INIT „RS1:\$00,\$84“

Mit der ersten Hex-Zahl (Überwachungskode) wird die Schnittstelle, z. B. RS1 so eingestellt, daß Zeichen ohne sog. 'Handshaking-Protokoll' ausgesendet werden. Denn die kann der Kassettenrecorder selbstverständlich nicht liefern. Der Überwachungskode wird auf \$00 gesetzt.

Der Initialisierungskode (zweite Hex-Zahl) setzt die Schnittstelle auf 8N1, das ist die Regel.

8 bedeutet Zeichenlänge 8 Bit, N = keine Parität, 1 = 1 Stopbit.

Gemäß Schema auf Seite 6-43 des MZ-800-Handbuchs ergibt sich bei 8N1 die binäre Reihe 1000100B. Diese Zahl ergibt eben umgerechnet 84H. So einfach ist das! Man möge es nachvollziehen: Die rechte Stelle der Binärzahl bedeutet 2^0 , die nächste 2^1 , 2^2 , usw. $2^2=4$, $2^7=128$. Zusammen also 132D (dezimal). Das ergibt nach Adam Riese 84H (hexadezimal).

Nun muß der Datenkanal in Richtung Schnittstelle zum Senden geöffnet werden:

WOPEN #1,"RS1:name"

Den Dateinamen kann man auch weglassen, er wird nicht ausgewertet.

Jetzt kommt ein ganz trivialer Befehl, der es aber in sich hat:

LIST #1

Während der Interpreter sämtliche BASIC-Befehle in Kurzform, den sogenannten TOKENS verwaltet, die wir außerhalb des Sharp-Systems nicht gebrauchen können, setzt der List-Befehl sie wieder in die 'Normalform' um. Wir wollen ja Klarheit sehen und nicht die verstümmelte Form. Das nutzen wir hier aus, indem wir in die Schnittstelle

'hineinLISTen'.

4.3 Laden nach CP/M

Zurückholen läßt sich das Programm nun wieder mit einem DFÜ-Programm, z.B. MOVE-IT.COM

move-it	(Programm starten)
ta	(Talk)
ESC g	(Get file)

Recorder starten, nach Ende der Signale

ESC e (Ende) eingeben

sichern, fertig.

Wer das Programm nicht hat, kann auch das Systemprogramm PIP.COM benutzen:

pip a:name=aux:

Cursor steht, Pgm wartet auf Zeichen. Nach Empfang von ^Z (EOF) wird selbständig abgespeichert.

Es muß darauf geachtet werden, daß mit AUX auch die richtige Schnittstelle eingestellt ist. Notfalls mit SETUP.COM korrigieren.

Jetzt haben wir den alten Quelltext im CP/M, bevor wir ihn weiter bearbeiten, biegen wir noch die verkehrten Zeichen gerade, vor allem die LINE-FEEDs. Programm SHPASC.PAS verwenden!

Erst jetzt liegt das übertragene Programm in einer Form vor, in der es unter MBASIC lauffähig wäre, sieht man einmal von der Syntax ab.

5. Verwendung eines zweiten Computers

Die Verwendung eines zweiten Computers als Zwischenspeicher ist eleganter, allerdings gibt es auch hier wieder einiges zu beachten.

Zunächst braucht man das richtige Kabel, um die seriellen Schnittstellen miteinander zu verbinden. Die

Anschlußstecker sind entweder 9- oder 25-polig nach RS232C (Bauart SUB-D). Die Belegung finden Sie am Ende dieses Beitrags. Der RS232C-Standard ist etwas problematisch, deswegen muß darauf noch kurz eingegangen werden:

Trotz ihrer scheinbar eindeutigen Bezeichnungen verhalten sich die Leitungen anders als man zunächst vermuten würde. Das hängt nämlich davon ab, ob es sich bei den zu verbindenden Geräten um eine Datenendeinrichtung (DTE=Data Terminal Equipment) oder eine Datenübertragungseinrichtung (DCE = Data Communication Equipment) handelt. Ich sage es gleich; man weiß es nie, und am Einsatzzweck des Gerätes ist es auch nicht immer zu erkennen. Ein Computer kann durchaus DCE sein. Eine 'READ'-Leitung kann mal senden, mal empfangen. Immerhin verhält sich unser MZ-800 da halbwegs 'normal', er ist als DTE geschaltet, wenn wir die Original-Schnittstelle verwenden. (Beim Selbstbau kann man das selber bestimmen.).

Modems und Koppler sind meist DCE.

So müßte man zur Verbindung zweier MZ-800 oder MZ mit PC o.ä. zwei Anschlußpaare kreuzen (s.Bild). Ich empfehle trotzdem etwas anderes:

Stellen Sie sich ein 1:1-Kabel her, bei denen die beschriebenen Anschlüsse durchverbunden sind. Man braucht nicht alle. Das Kabel sollte ausreichend lang sein, die serielle Übertragung ist nicht auf die 3m begrenzt wie bei der parallelen. Dazu stellen Sie sich dann zusätzlich ein sogenanntes 'Nullmodem' her, bei dem die Leitungen gekreuzt sind. Es besteht aus Kupplung und Stecker SUB-D (9- oder 25-polig, je nach Gerät), die man durch ca. 3cm lange Distanzhülsen auf Abstand hält. Mit entsprechenden Schrauben festziehen, die Leitungen können danach angelötet

Sharp-BASIC / MBASIC (CP/M) und GW-BASIC (MS-DOS)

werden. Das ganze ist dann so fest, daß man auf ein Gehäuse verzichten kann.

Mit dieser 'Ausrüstung' können Sie fast alle seriellen Übertragungsprobleme lösen.

Um ein bißchen Probieren wird man nicht herumkommen, es sei auch nicht verschwiegen, daß sich einige Geräte, auch Computer sehr exotisch verhalten. Ich verweise auf das im Sybex-Verlag erschienene Buch 'Daten-Kommunikation mit RS232C'.

Nach Einstellung der richtigen Baudrate auf beiden Seiten geht nun alles sehr einfach.

Datei senden geht aus dem Sharp-System heraus wieder mit dem LIST#n-Befehl, von CP/M aus mit PIP.COM oder DFÜ-Programm.

Der empfangende Rechner empfängt ebenso, PCs mit COPY-Befehl, nachdem mit dem MODE-Befehl die richtigen Übertragungsparameter eingestellt wurden, z.B. 8N1.

Wenn man das alles einmal richtig hinbekommen hat, ist alles weitere ein Kinderspiel. Ich habe auf alle beschriebenen Arten sehr viel Daten und Programme übertragen, dabei ist nur ein einziges mal ein falsches Zeichen aufgetreten. Das ist im Falle eines BASIC-Programms meist nicht tragisch, weil der Fehler bemerkt wird und verbessert werden kann.

Beim ersten Austesten sollte man eine kleine Baudrate einstellen, um die Vorgänge besser beobachten zu können. Ich verwende problemlos nur noch 19200 Baud!

Für die absolute Übertragungssicherheit, etwa bei Maschinenprogrammen empfiehlt sich ein Übertragungsprotokoll, etwa das bekannte KERMIT. Es ist ein Freipro-

gramm, läuft bei mir aber nicht richtig auf dem 800er. Auf dem PC hingegen bestens. Wenn ich meinem Bruder nach Hamburg Daten überspiele, wird bei schlechter Telefonverbindung ein nicht korrekt übermittelter Block sofort noch einmal wiederholt. Wahrscheinlich mache ich auf dem 800er einen Bedienungsfehler, so daß ein Leser weiterhelfen kann?

Da die Übertragung von mehreren Programmen unter CP/M mit PIP.COM wegen der Tipparbeit umständlich ist, kann man auch eines der vielen Dienstprogramme, wie WASH, SWEEP oder NSWEEP benutzen. Als Geräteart ist dann häufig 'P' zu wählen. P wie 'PUNCH' stammt noch aus der Zeit des Dampfcomputers mit Lochstreifenstanzer. Beim 800er bedient er den Hilfskanal AUX, also die serielle Schnittstelle (mit SETUP.COM ansehen!).

6. Feinbearbeitung von BASIC-Dateien

Viele Anfragen habe ich erhalten zur weiteren Bearbeitung der übertragenen Quelltexte. Deswegen will ich auch darauf nochmal eingehen.

Wenn die Bearbeitung im CP/M erfolgen soll, benötigen wir einen leistungsfähigen Editor, der mindestens so gut ist wie der von Sharp. WORDSTAR und Turbo-PASCAL haben genau das, was wir brauchen, eine Austausch-Funktion.

Bei WORDSTAR sollte der N-Modus (Non-Document) gewählt werden, weil er das achte Bit nicht benutzt, in dem die in diesem Falle lästigen Format-Befehle versteckt sind.

Nachdem das unfertige Programm geladen wurde, schauen wir uns an, wo die Syntax noch nicht stimmt. Das ist z.B. PRINT\$, INPUT\$, IFA=0THENB=0, also alles Konstruktionen, die nach Zwischenräumen verlangen.

Wer erstmals einen MBASIC-Quelltext bearbeitet, wird zunächst einmal einen Testlauf mit dem Interpreter starten, um festzustellen, was alles nicht läuft. Das notieren wir. Anschließend wird versucht, mit CONT weiterzumachen. Bis zum nächsten Fehler und so weiter. Die korrekte Syntax und Funktionsweise sind beschrieben in dem im Markt&Technik-Verlag erschienenen Buch 'BASIC-80 und CP/M' von J. J. Purdum.

Nach Starten des WS- oder Turbo-Editors wird mit ^QA (^ ist das CONTROL-Zeichen) die Austausch-Funktion aufgerufen. Es wird gefragt, was getauscht werden soll. Die Frage wird einfach mit IF beantwortet (nur I und F eingeben, danach RETURN). Danach wird gefragt, wogegen ausgetauscht werden soll. Mit I, F und einer Leertaste liegen wir richtig. Die wird nämlich auch berücksichtigt.

Nach RETURN fragt der Rechner die Optionen ab. 'g' ändert im gesamten Text. Es muß nun immer - je nach Version - mit J/N oder Y/N bestätigt oder verworfen werden. Sind wir unserer Sache ganz sicher, kann mit 'ng' auf die Nachfrage verzichtet werden (nur 'n', 'g' und RETURN eingeben).

Wenn das abgelaufen ist, entdecken wir nach jedem 'IF' ein Leerzeichen, genau so wollten wir es haben. Mit 'THEN' verfahren wir ebenso, allerdings muß bei der zweiten Abfrage vor dem 'I' von 'THEN' noch ein Leerzeichen mit der SPACE-Taste eingegeben werden (danach auch).

Da der CURSOR nach der IF-Austausch-Operation am Ende des Textes stand, müßte nun 'ngb' eingegeben werden, damit rückwärts ausgetauscht wird ('b' für 'back'). Bei kürzeren Programmen kann man das machen, bei längeren empfiehlt es sich, vor jeder erneuten Operation mit ^QR wieder an

den Anfang zu springen, weil die Sache rückwärts langsamer abläuft.

Die Editoren von Turbo-PASCAL und WORDSTAR arbeiten in dieser Hinsicht gleich. Und auch sonst sind bei beiden kaum Unterschiede zu finden. Wer in der glücklichen Lage ist, über einen PC oder AT zu verfügen, kann die Programme sofort dorthin übertragen und das GW-BASIC oder den Quick-BASIC-Compiler zur Änderung benutzen. Sie sind, wie gesagt aus einem Hause und in der genannten Reihenfolge aufwärtskompatibel.

Die Editoren sind nochmals um einiges besser als der von Sharp. Änderungen und Tests können vorgenommen werden, ohne das System zu verlassen.

Wenn keine neuen, GW-spezifischen Befehle hinzugefügt werden, läuft das Programm nach der Bearbeitung natürlich auch wieder unter CP/M. Der Rücktransfer erfolgt wieder wie oben beschrieben. Da MS-DOS aus CP/M hervorgegangen ist, gibt es keine Kompatibilitätsprobleme.

7. Zurück ins SHARP-BASIC

Komplizierter ist die Rück-Übertragung von CP/M oder MS-DOS in das Sharp-System. Ich habe selbst dafür zwar kaum Anwendung, aber stelle die Lösung trotzdem hier vor, falls einer der Leser das braucht.

Auf der MZ-Seite wird mit einem kleinen Hilfsprogramm der ASCII-Text über die Schnittstelle eingelesen:

```
10 INIT „RS1:$04,$84“
20 ROPEN #1,“RS1:“
30 WOPEN #2,“FD2:“name“
40 INPUT #1,A$
50 PRINT A$;
60 PRINT #2,A$
70 GOTO 40
```

Mit dem COPY-Befehl von MS-DOS, PIP.COM von CP/M oder einem ge-

eigneten DFÜ-Programm empfangen wir ASCII-Code, und zwar korrekten. Auf dem Bildschirm des MZ-800 kann man die Übertragung beobachten.

In Zeile 10 wird wieder die Schnittstelle des MZ eingestellt. Der Überwachungscode ist diesmal auf 04H (entspricht 0000100B binär) gesetzt, damit ist CTS (CTS=Clear To Send) aktiviert. Im Falle des Empfangs heißt das, daß nur dann Daten gesendet werden können, wenn der 800er bereit ist (er muß ja noch die Zeichen auf den Bildschirm bringen und in die Datei 'name' schreiben).

Zeile 20 macht die serielle Schnittstelle empfangsbereit. Es muß jetzt aber noch eine zweite Datei geöffnet werden (Zeile 30), in die wir die Daten dauerhaft hineinschreiben.

Zeile 40 empfängt Zeichen, 50 zeigt sie an, 60 schreibt sie weg. Nach jedem Zeichen wird durch 70 alles wiederholt.

Wie man sehen kann, werden die Kleinbuchstaben bei Sharp wieder falsch interpretiert. Das kann man durch eine kleine READ-DATA-Anweisung beheben.

Da es Probleme mit dem Dateiende-Zeichen EOF (=^Z) geben kann, muß die Datei noch geschlossen werden. Das geht problemlos 'von Hand'. Mit SHIFT/BREAK abbrechen und CLOSE eingeben, danach RETURN. Damit werden beide geöffneten Datenkanäle geschlossen. Man hört es auch, weil die Floppy wieder anspringt. Erst jetzt ist der Text gerettet.

Wer neben der seriellen Schnittstelle auch noch über eine RAM-Floppy verfügt, sollte diese bei langen Übertragungsvorgängen nutzen.

Nun haben wir einen lesbaren (Sharp)-ASCII-File, im Directory mit BSD gekennzeichnet. Der Versuch,

ihn mit dem Befehl LOAD "name",A zu laden scheitert kläglich mit einem 'Illegal Data Error'. Es liegt wieder an den verdrehten LINE-FEEDS. Man kann sie auf der Senderseite herausfiltern (TELIX unter MS-DOS hat solche Optionen, ALT-O drücken!). Nun erst läßt sich ein File mit LOAD „name“.A einlesen.

ACHTUNG: Nach dem Einlesen von längeren ASCII-Dateien mit LOAD,A schaltet die Floppy ab, der CURSOR blinkt nicht und man vermutet zunächst einen Programmabsturz. Nach einiger Wartezeit erscheint jedoch der Cursor. Das liegt daran, daß der Rechner erst einmal den Sharp-ASCII-Code, den wir ja bei LOAD,A einlesen in das durch den Interpreter verwertbare Programmcode mit TOKEN umsetzen muß. Das dauert ein wenig. Man kann es mal beobachten, wenn mit CONTROL/RESET der Vorgang abgebrochen wird. Ein anschließendes LIST zeigt, an welcher Stelle der Übersetzungsvorgang unterbrochen wurde.

Um das Programm zu sichern, gibt man erst einmal den normalen SAVE-Befehl ein, bevor man einen Testlauf unternimmt.

Viel Spaß beim Tuning Ihrer Programme!

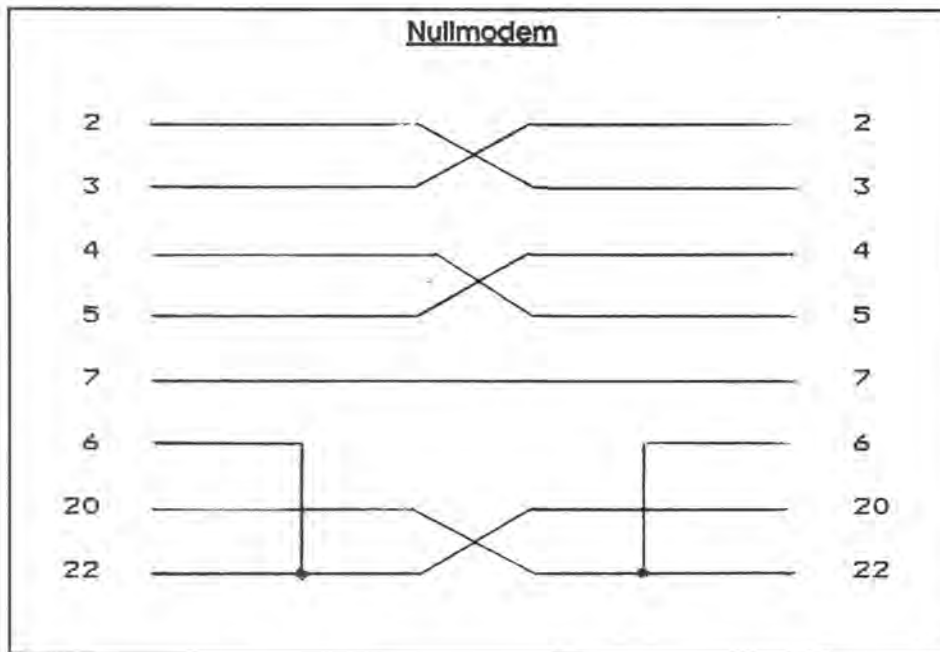
Dirk Grube, Parnaßweg 7,
D-2320 Plön

Sharp-BASIC / MBASIC [CP/M] und GW-BASIC [MS-DOS]

Steckerbelegung nach RS 232C (V.24)

25pol	9pol	Abk.	Bedeutung	
1	-		Protective Ground	Gehäuse-Erde
2	3	TxD	Transmit Data	Sendedaten
3	2	RxD	Receive Data	Empfangsdaten
4	7	RTS	Request to send	Sendeteil einschalten
5	8	CTS	Clear to send	Sendebereitschaft
6	6	DTS	Data Set Ready	Betriebsbereitschaft
7	5	GND	Signal Ground	Signal-Erde
8	1	DCD	DataCarrier Detect	Empfangssignalpegel
20	4	DTR	DataTerminal Ready	DEE betriebsbereit
22	9	RI	Ring Indicator	Ankommender Ruf

Für die in diesem Beitrag beschriebenen Funktionen werden nur die Anschlüsse TxD, RxD, RTS und GND benötigt. Dennoch sollte man sich für weitere Anwendungen gleich ein vollständiges 1:1-Kabel herstellen. Dzu ein Nullmodem.



PROGRAM shpasc;

(* Dirk Grube
Parnaßweg 7
2320 Ploen

04522 / 2017

Das Programm setzt den speziellen Sharp-Zeichensatz um auf korrekten ASCII-Code *)

VAR source, ziel : text;
sname, zname : STRING(40);
ch : char;
i : real;

a.zaehler : integer;

```
PROCEDURE p(c:byte);
BEGIN
  write(ziel,chr(c));
END;
```

```
PROCEDURE kodiereum (VAR
ch:char);
VAR z:byte;
```

```
BEGIN
  z:=ord(ch);
```

```
IF (z>=$20) THEN p(z);
```

```
BEGIN
  CASE z OF
    $0D:BE-
GINp($0D);p($0A);END;
```

```
$20 : p($20);
$2B : p($76);
$3B : p($36);
$4B : p($46);
$5B : p($56);
$90 : p($5F);
$92 : p($65);
$93 : p($60);
$94 : p($7E);
$96 : p($74);
$97 : p($67);
$98 : p($68);
$9A : p($62);
$9B : p($78);
$9C : p($64);
$9D : p($72);
$9E : p($70);
$9F : p($63);
```

```
$A0 : p($71);
$A1 : p($61);
$A2 : p($7A);
$A3 : p($77);
$A4 : p($73);
$A5 : p($75);
$A6 : p($69);
$A8 : p($5C);
$A9 : p($6B);
$AA : p($66);
$AB : p($76);
$AD : p($7D);
$AE : p($7E);
$AF : p($6A);
```

```
$B0 : p($6E);
$B2 : p($5D);
```


Das erste Prüfprogramm: Mikro-CAD

```
SB4 : p($6D);  
SB7 : p($6F);  
SB8 : p($6C);  
SB9 : p($5B);  
SBA : p($7C);  
SBB : p($7B);  
SBD : p($79);
```

```
END;  
END;
```

```
END;
```

```
BEGIN
```

```
write('Name des Quellprogramms: '); readln(sname);  
write('Laenge des Quellprogramms in Byte: '); readln(zaeher);  
assign(source,sname);  
reset(source);  
write('Name der Zieldatei: '); readln(zname);  
assign(ziel,zname);  
rewrite(ziel);
```

```
for a:=1 to zaeher do begin;
```

```
    read(source,ch);  
    kodiereum(ch);
```

```
end; (* do *)
```

```
write (ziel,EOF);  
close(ziel);  
END.
```

Premiere – das erste Prüfprogramm im Magazin: Mikro-CAD

Hiermit möchten wir allen Besitzern des MZ-800 mit 5,25" Doppellaufwerk und V-RAM-Erweiterung ein neues Zeichenprogramm vorstellen. Dieses umfangreiche vierteilige Programmpaket mit einer Gesamtlänge von rund 50 kByte unterstützt die Erstellung von Zeichnungen, Titelbildern, kleinen Grafiken, Bildhintergründen usw. und schließt zum Schluß damit ab, daraus ein eigenständiges Programm zu produzieren.

Zur Anwendung kommen alle unter BASIC verwendbaren Grafikbefehle: LINE, BOX, SET, RESET, CIRCLE, PAINT, SYMBOL, COLOR, PAINT, und PATTERN. Die Benutzerführung wurde mit kurzen Abfragen recht übersichtlich gestaltet, so daß man zwar anfangs nicht ohne Bedienungsanleitung auskommt, jedoch nach kurzem Gebrauch das Programm auch „blind“ bedienen kann.

Mit einem aufrufbaren Mustergenerator können Muster mit einer Kantenlänge von 24x24 Punkten entworfen und in einer editierbaren Musterbibliothek abgelegt werden. Diese Muster werden mit dem PATTERN-Befehl in die Zeichnung eingetragen.

In der vorliegenden Version werden sämtliche Zeichnungen mit einer Auflösung von 320x199 Punkten, dafür jedoch in bis zu 16 Farben, gezeichnet. Nach ersten Rücksprachen ist auch eine Version mit doppelter Auflösung (vier Farben) möglich, die jedoch nur bei ausreichendem Interesse angeboten wird.

Sämtliche Befehle werden sofort auf dem Bildschirm und dort auch an der richtigen Position angezeigt, die bis zum Befehlsabschluß verändert werden kann. Ein Vorteil, der sich bald, allein durch Zeitersparnis, bezahlt macht.

Eine interessante Variante ist außerdem, daß sich die FD-Laufwerke nach einiger Zeit abschalten, sobald sie nicht aktiv sind. Dies werden nicht nur die Energiesparer unter uns begrüßen.

Sollten Sie in der glücklichen Lage sein, eine 64k RAM-Karte zusätzlich angeschlossen zu haben, wird diese automatisch erkannt und zum gegenseitigen Aufruf der Unterprogramme herangezogen. Leider kann diese Funktion nicht unterdrückt werden.

Ein weiterer Nachteil ergibt sich aus der Verwendung einer Random-Datei: Wird für den SYMBOL-Befehl eine Textzeile eingegeben, so kann diese nur maximal 32 Zeichen lang sein. Um also 40 Zeichen in einer Zeile ausgeben zu können, muß man die nächsten 8 zusätzlich ansetzen.

Alles in allem kann man sagen, daß dieses Programm für viele Anwendungen benutzt werden kann, sei es, um schnell mal eine Geburtstagskarte zu entwerfen, ein Titelblatt für ein selbgeschriebenes Programm aufzuzeichnen usw.

Für die Bestellung bieten wir Ihnen folgende zwei Möglichkeiten:

1. Sie bestellen sofort das Mikro-CAD Gesamtpaket zu DM 69,00 oder
2. zunächst einmal das Prüfprogramm für DM 20,00, überlegen in Ruhe, ob Mikro-Cad Ihren Ansprüchen genügt, und bestellen erst dann für eine Restsumme von DM 55,00 das Gesamtpaket nach.

Alle Preise gelten zuzüglich Porto und Versandkosten in Höhe von DM 5,00. Der Versand erfolgt nur auf 5,25"-Diskette.

64K-Static-RAM Karte für den MZ-800

64K-Static-RAM Karte für MZ-800

Hier ist die Schaltung einer 64K RAM-Karte, die im Gegensatz zur SHARP 64K RAM-Karte, mit CMOS-RAM und mit einer Batteriepufferung ausgestattet ist. Die Karte wird bei uns in einer selbstgebauten Erweiterungsbox betrieben, die über ein 50-poliges Flachbandkabel mit dem MZ-800 verbunden ist. Die Programme und die Daten bleiben auch nach dem ausschalten des Rechners erhalten. Dies ist besonders für die Besitzer eines MZ-800 mit CMT von Interesse.

Die Adressierung wird mit einem 74LS138 und einigen NAND- und NOR-Gattern vorgenommen. Durch einen Schreibbefehl (OUT (C),L) auf die Adresse 0EBH (dez. 235) wird die Speicheradresse (hier in den Reg. B und L) in die vier Zähler übernommen. Auf den Datenleitungen D0 - D7 liegen die acht unteren Adreßbits und auf den Adreßleitungen A8 - A15 die oberen Bits. Die Ausgänge A0' - A12' der Zähler liegen direkt an den Speicher-ICs, die übrigen drei Ausgänge dienen zur Auswahl der einzelnen SRAMs. Die Daten können über die Portadresse 0EAH (dez. 234) gelesen und geschrieben werden. Nach jedem Lesen oder Schreiben der Daten wird automatisch der Zählerstand um eins erhöht. Diese Möglichkeit wird von manchen Programmen, aber nicht vom 800er BASIC genutzt. Die zur Pufferung eingesetzten NiCd-Akkus werden automatisch nachgeladen.

Die Karte läuft schon über zwei Jahre zu unserer vollsten Zufriedenheit. Am Anfang traten noch beim Ein- und Ausschalten des Rechners Probleme auf. Einzelne Bytes veränderten ihren Inhalt durch Störimpulse. Durch den Einbau einer Zusatzschaltung, CMOS 4001 am Eingang CS2 der RAM-Bausteine, traten die Veränderungen nicht mehr auf. Beim Abfallen der Versorgungs-

spannung unter +4V, werden die Speicher-ICs deaktiviert.

Für das sichere Arbeiten der Zusatzschaltung muß sie noch eingestellt werden. Dazu wird ein Voltmeter an den Punkt X angeschlossen und das 1k-Poti so gedreht, daß der H-Pegel angezeigt wird. Das Poti langsam zurückdrehen bis L-Pegel angezeigt wird und dann noch ein bißchen weiter. Damit ist die Karte einsatzbereit. Wenn die Batteriepufferung nicht benötigt wird, so entfällt auch der IC 4001 und die Pins 26 und 28 an den SRAMs müssen fest auf +5V gelegt werden.

mich an das Datenblatt des Herstellers gehalten. Die gesamte Schaltung vereinfacht sich entsprechend, wenn anstatt der 6264-Typen (8KByte) RAM-ICs mit größerer Speicherkapazität benutzt werden. Dadurch wird die Dekodierung der SRAMs vereinfacht und Platz gespart.

Hier ist noch ein kleines Maschinenprogramm zum Testen der RAM-Karte. Durch das Testprogramm wird der Inhalt der Speicher-ICs verändert. Nach dem Testen muß die RAM neu initiiert werden.

2000H	3E	08	D3	CE	D3	E4	CD	09	00	3E	3E	CD	12	00	11	A3
2010H	11	CD	03	00	1A	FE	1B	CA	AD	00	FE	3E	20	E8	13	1A
2020H	D6	4C	28	04	FE	07	20	DE	67	13	1A	D6	4D	28	04	FE
2030H	02	20	D3	B7	28	02	3E	37	32	5E	20	32	71	20	7C	B7
2040H	F5	C4	62	20	F1	CC	4A	20	18	BC	21	00	00	37	DC	75
2050H	20	CD	7B	20	DB	EA	57	BB	C4	83	20	23	7C	B5	00	20
2060H	ED	C9	21	00	00	37	DC	75	20	CD	7B	20	D3	EA	23	7C
2070H	B5	00	20	F2	C9	44	0E	EB	ED	69	C9	7D	CB	44	28	01
2080H	2F	5F	C9	D5	E5	CD	09	00	CD	BA	03	CD	0C	00	4A	7B
2090H	AA	57	06	08	AF	CB	21	CE	30	CD	12	00	CD	B1	0F	CB
20A0H	22	3E	71	30	02	3E	17	2B	CB	DC	77	10	E7	CD	B3	09
20B0H	E1	D1	FE	CB	C0	F1	C9									

Wir benutzen zwei getrennte 64K-Blöcke, die mit einem Taster und durch Software umgeschaltet werden können. Die einfachste Möglichkeit, die Umschaltung mit einem Schalter, habe ich extra aufgezeichnet.

Noch ein paar Tips zum Aufbau:

Wegen Platzmangels auf der Platine habe ich jeweils vier SRAMs übereinandergesetzt und alle Pins bis auf Pin 20 (CS1) und Pin 26 (CS2) vorsichtig miteinander verlötet. Die Pins 20 und 26 werden entsprechend ihrer Zuordnung frei verdrahtet. Durch das Übereinandersetzen der RAM-ICs erspart man sich auch beim Aufbau des 2. 64K Blocks die vier Zähler und die Adressierungs-ICs.

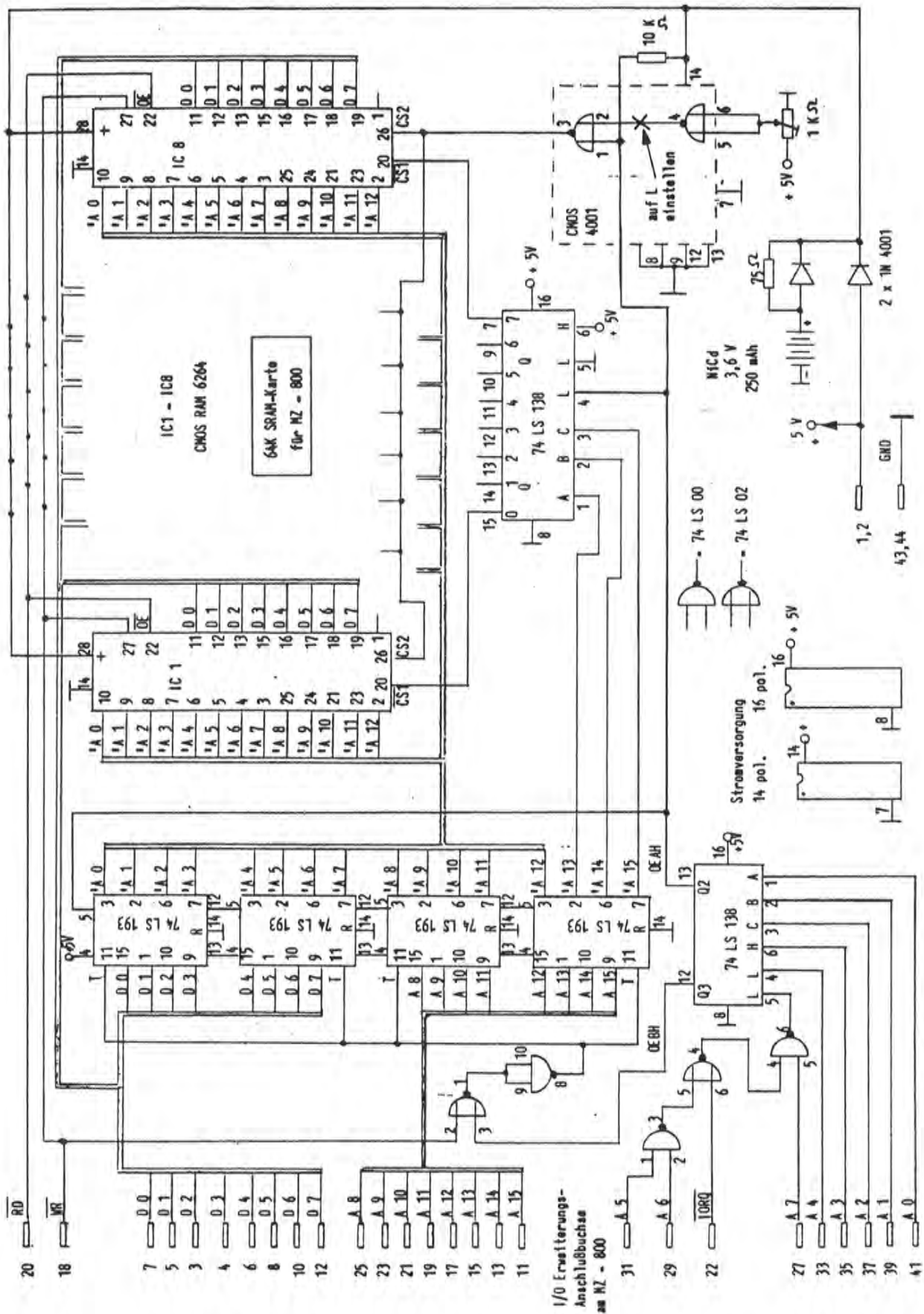
Die Zuordnung der Adreß- und Datenleitungen an die SRAM-Pins ist beliebig. In der Zeichnung habe ich

Das Testprogramm wird im Monitor mit 'M' ab Adresse 2000H eingegeben und mit 'S' auf Kassette abgespeichert. Das Programm wird mit J2000 gestartet. Es erscheint nun das Prompt-Zeichen.

Eingabe 1. Buchstabe: 'L' für Lesen oder 'S' für Schreiben.
Eingabe 2. Buchstabe: 'O' für ohne oder 'M' mit automatischem Hochzählen der Adressen. CR drücken.

Beim Schreiben werden die die RAM-ICs mit einem bestimmten Bitmuster beschrieben. Beim Lesen wird der Inhalt auf Richtigkeit überprüft. Wenn alles OK ist, erscheint wieder das Prompt-Zeichen. Werden beim Lesen Fehler festgestellt, so werden Bits invers angezeigt. Zum Weiterlesen beliebige Taste drücken. Abbruch mit SHIFT-BREA. Bei Batteriepufferung muß auch nach mehrmaligem Ein- und Aus-

64K-Static-RAM Karte für den MZ-800



Einführung in die Grafik des MZ-800

schalten des Rechners der Test positiv verlaufen.

Der Autor lehnt, wie bei solchen Schaltungen üblich, für eventuelle Schäden jede Haftung ab.

Hans Werner und Bernd Birkenbach, Wiener Str. 31, 6000 Frankfurt 70

Einführung in die Grafik des MZ-800

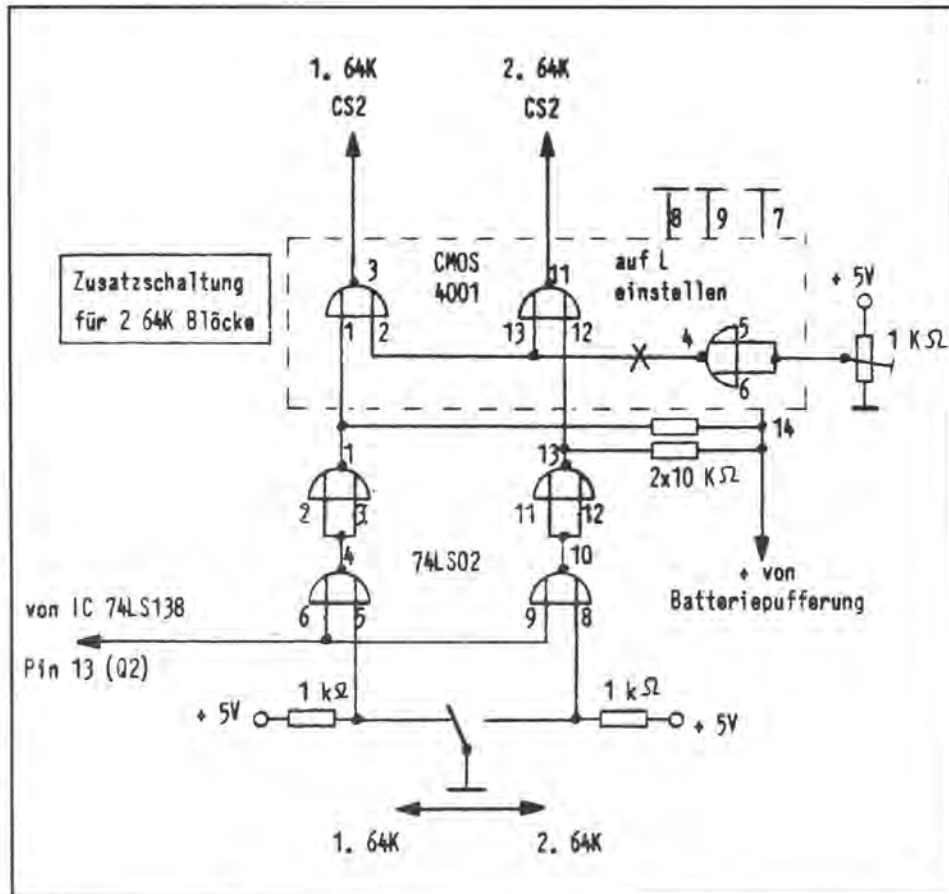
Grundlegendes

Eine Grafik wird in einem Speicherbereich, dem sogenannten Video RAM, abgelegt. Damit wir die Grafik-Befehle des BASIC verstehen, müssen wir zunächst einige Begriffe, die mehr oder weniger eng mit der Hardware verbunden sind, klären. Dieser Teil kann kaum lang genug ausfallen, da er im Handbuch vollständig unterschlagen wird. Danach will dieser Artikel noch Klarheit in einige unvollständig bzw. falsch beschriebene Befehle bringen. Ich bemühe mich, das Handbuch nicht zu wiederholen, gebe jedoch verweise für den unerfahrenen Leser.

Bitplanes

Da wir kein Bild in den Monitor malen wollen, müssen wir die Grafik in ein für den Computer verständliches Format bringen. Zunächst wird das Bild in Rasterpunkte – sogenannte Pixel – aufgeteilt. Die Anzahl der Pixel bestimmt die Auflösung. Der MZ-800 hat zwei Auflösungsmodi. Zum einen 320*200 und zum anderen 640*200 Pixel.

Setzen wir voraus, daß nur zwei Farben zur Wahl stehen – auch monochrom genannt, da ein Punkt praktisch nur gesetzt oder nicht gesetzt (= in der Hintergrundfarbe gesetzt!) sein kann. In diesem Fall bietet es sich an, für jeden Punkt des Bildschirms im Video RAM ein Bit zu



reservieren, das aussagt, ob der Punkt nun in Hinter- oder Vordergrundfarbe erscheinen soll. Der Speicherbereich, der für die Darstellung des gesamten Bildes dann nötig ist, ist eine Bitplane. In einer Bitplane gibt es also für jedes Pixel genau ein Bit, welches besagt, ob das Pixel gesetzt sein soll oder nicht.

Nun hat der MZ-800 jedoch nicht nur die Möglichkeit, zwei Farben darzustellen, sondern kann mit erweitertem Video RAM bis zu 16 Farben gleichzeitig darstellen. Dies ist mit der einfachen Bitplane nicht möglich. Nun könnte man natürlich für jede Farbe eine eigene Bitplane anlegen. Daß dies jedoch nicht der Fall ist und warum man das nicht tut, zeigt eine kleine Rechnung. Der MZ-800 kann bei voller Speicheraufrüstung – also 32K Video RAM – 16 Farben gleichzeitig bei einer Auflösung von 320*200 Pixeln darstellen. Legte man für jede Farbe eine eigene Bitplane an, so wären das:

$320 \cdot 200 = 64.000$ Pixel bzw. Bits pro Bitplane
 $16 \text{ Bitplanes} \cdot 64.000 \text{ Bits} = 1.000 \text{ KBit} = 125 \text{ K}$

Nun sind 32K aber nur ein knappes Viertel davon, und es ist genauso viel Information darin enthalten. In 32K passen jedoch nur vier ganze Bitplanes!

Farbtafelwerte

Die Lösung des Knotens heißt: Farbtafelwert. Nehmen wir aus jeder Bitplane das einem Pixel entsprechende Bit, so erhalten wir bei vier Bitplanes, eine 4 Bit Wert für jedes Pixel – aus jeder Bitplane ein Bit. Interpretieren wir diesen Wert als Binärzahl, so lassen sich mit vier Bit die Zahlen von 0 bis 15 darstellen. Diese 16 Werte, die Aufschluß über die Kombination der Bitplanes geben, sind die Farbtafelwerte. 16 Farben und 15 Farbtafelwerte, dies ist kein Zufall. In einem Betriebsmodus gibt es immer soviel Farbtafelwerte wie

Einführung in die Grafik des MZ-800

in diesem Modus Farben gleichzeitig dargestellt werden können. Bei zwei Bitplanes gibt es nur vier Farbtafelwerte, also können auch nur vier Farben gleichzeitig dargestellt werden.

Farbwerte

Die Anzahl der Farbwerte ist in der Regel eine Systemkonstante. Der Farbwert ist nichts weiter als eine Kennnummer für die darstellbaren Farben. Sie sind unabhängig von Auflösung oder Anzahl der Bitplanes. Welchen Modus Sie auch wählen, auf dem MZ-800 ist der Farbwert für schwarz immer 0, für braun immer 6 und für weiß immer 15. Im Handbuch finden sich diese Werte als Farbcodes bezeichnet im Anhang A (A-3). Die angegebenen Farben variieren jedoch nach Monitor und Übertragungsweg. Während Farbe 6 auf einem Fernseher über Videokabel satt gelb aussieht, erscheint er auf einem RGB-Monitor eher als helles Kastanienbraun.

Paletten

Nach Auswahl der Anzeigebe triebsart im dem INIT-Befehl – siehe hierzu ebenfalls Anhang A (übrigens: Betriebsart M0 existiert nicht. Druckfehler!) – will man nun den Farbtafelwerten Farben zuordnen. Dies übernimmt die Farbtafel, die normalerweise nur ein anderes Wort für Palette ist. Beim MZ-800 im Modus M2 ist Palette und Farbtafel jedoch zu unterscheiden. Der Grund dafür liegt in der Hardware.

Der MZ-800 realisiert die Paletten durch Register in seinem Custom Chip. Eine andere Möglichkeit für eine Farbtafel ist auch nicht vorgesehen. Leider verfügt der Custom Chip nur über Palettenregister. Habe ich nur vier oder weniger Farbtafelwerte, so kann ich über die Register eine vollständige Farbtafel erzeugen. In der Farbtafel steht dann für jeden Farbtafelwert ein entsprechender Farbwert. Diese Zuordnung kann mit dem PAL-Befehl beeinflusst werden. Der PAL-Befehl

hat das Format:

PAL Farbtafelwert, Farbwert

Um zum Beispiel dem Farbtafelwert 2 die Farbe Grün zuzuordnen, lautet der Befehl: PAL 2, 4, da 4 der Farbwert von Grün ist. Sollte bereits eine Grafik auf dem Bildschirm sein, so werden alle Pixel, deren Bitplane-Kombination den Farbtafelwert 2 ergeben, ihre Farbe in Grün verändern und jeder Punkt, der von nun an mit dem Farbtafelwert 2 in die Bitplanes geschrieben wird, wird ebenfalls grün erscheinen, bis die Palettenzuordnung durch einen neuen PAL-Befehl für Farbtafelwert 2 geändert wird.

Probleme gibt es nun bei 16 Farbtafelwerten. Für 16 Farbtafelwerte nur vier Register! Der nun von Sharp angewandte Trick ist aus der Not geboren aber einleuchtend. Man macht zunächst die grundlegende Übereinkunft, daß nicht definierte Farbtafelwerte den Farbwerten gleichzusetzen sind. Also Farbtafelwert 2 hat auch den Farbwert 2, also Rot, wenn in den Registern nichts anderes steht. Da die Palettenregister per Hardware nur auf zwei Bitplanes reagieren – nämlich auf die unteren zwei, die oberen beiden Planes jedoch nicht unberücksichtigt beliben dürfen, werden die Register erst durch eine bestimmte Kombination der oberen Planes aktiviert. Für den BASIC-Programmierer heißt das, daß er die 16 Farbtafelwerte in vier aufeinanderfolgenden Blöcke mit jeweils vier aufeinanderfolgenden Farbtafelwerten aufteilen muß. Die Paletten werden nur auf den einen, gerade aktivierten Block angewandt. Die anderen Farbtafelwerte gelten als nicht definiert und werden somit direkt als Farbwerte interpretiert. Das hat enorme Konsequenzen. Habe ich also zunächst bei aktiviertem Block 0 einen Punkt mit Farbtafelwert 2 geschrieben, der wegen der Palettendefinition weiß war, und einen zweiten Punkt mit Tafelwert 6, der –

da keine Palette angewandt wird – gelb bis braun erscheint, und verändere ich nun bei gleichem Registerinhalt den Block nach 1, so erscheint der erste Punkt rot, da keine Palette mehr angewandt wird, und der zweite weiß, da Register 2 nun auf Farbtafelwert 6 angewandt wird. Man ändert die Register, indem man den PAL-Befehl genauso verwendet wie in den anderen Modi. Liegt der Farbtafelwert außerhalb des Blocks, gibt es jedoch eine Fehlermeldung.

Die Blöcke und die Umschaltung mit Hilfe des INIT-Befehls sind im Anhang A beschrieben. Die Beispiele und den Rest von Punkt 5 vergißt man aber am besten schnell. Als besseres Beispiel zum Experimentieren schlage ich deshalb das **Listing 1** vor. Es zeigt nach Eingabe der Blocknummer und der Register (nicht der Farbtafelwerte, selbige werden erst in den Zellen 510 bis 540 berechnet!) die Zuordnungen der Farbtafelwerte zu den Farben.

Wem die dauernde Blockumschalterei lästig erscheint, der sollte sich einmal **Listing 2** ansehen. Es programmiert die Hardware selbst und nimmt auch die Blockumschaltung selbst vor. Übrigens: Programmiert man diese Routine als Maschinensprache, so ist sie schneller als der normale Weg. Wie diese Routine in ein Programm eingebaut wird – hier Listing 1 – zeigt **Listing 3**. Listing 1 und 3 werden übrigens durch Eingabe einer Zahl größer als 3 für die Blocknummer abgebrochen.

Farbspezifikation

Bei den meisten Grafikbefehlen ist das erste Parameterpaar die Farbspezifikation. Die beiden Werte werden grundsätzlich in eckigen Klammern angegeben. Nach den Klammern folgt kein Komma! In Klammern steht vor dem Komma der Farbtafelwert, der in die Bitplanes geschrieben werden soll, und nach dem Komma der Überlagerungsmodus. Der Überlagerungsmodus

Einführung in die Grafik des M2-800

bei den Befehlen RESET und BLINE unterscheidet sich von den anderen und wird deshalb später noch behandelt. Bei allen Befehlen kann der Überlagerungsmodus die Werte 0 und 1 annehmen.

Bei den normalen Befehlen ist der übliche Modus 0. Hier wird in die Bitplanes exakt der gewählte Farbtafelwert geschrieben. Jeder Punkt erscheint dann also in der Farbe, die dem Farbtafelwert in der Farbtafel zugeordnet ist.

Der Modus 1 verknüpft den bereits vorhandenen Farbtafelwert aus den Bitplanes mit dem neuen Farbtafelwert mit einer logischen ODER-Operation. Die Ergebnisse solcher Verknüpfungen sind in einer Tabelle im Anhang A (A-6) dargestellt. Das Ergebnis wird dann als neuer Farbtafelwert in die Planes geschrieben. Überschreibe ich also im 16 Farben Modus, ohne Paletten gesetzt zu haben, einen blauen Punkt mit einem grünen, so erhalte ich einen Punkt in Cyan.

Übrigens: Wenn man bei der Farbspezifikation das Komma und den Modus ausläßt, wird für den Modus automatisch der mit dem COLOR-Befehl eingestellte Modus gewählt. Ist kein COLOR-Befehl gegeben worden, so ist 0 voreingestellt.

Standardbefehle

Die Standardbefehle SET, BOX, LINE und CIRCLE sind im Handbuch genügend beschrieben. Es sei lediglich noch erwähnt, daß mit Palettencode der Farbtafelwert gemeint ist. Zu CIRCLE kann man noch sagen, daß ein Aspekt von 0.75 einen optisch etwa runden Kreis ergibt.

RESET und BLINE

Diese Befehle entsprechen SET und LINE, jedoch wird der Überlagerungsmodus anders interpretiert. Überlagerungsmodus 0 halte ich für Quatsch! Hier wird lediglich der angegebene Farbwert invertiert, daß heißt jedes Bit der Binärzahl in sein

Gegenteil verkehrt, und das Ergebnis wird als Farbtafelwert für einen LINE-Befehl verwendet. Das Ergebnis berechnet sich wie folgt:

Anzahl der Farbtafelwerte - Farbtafelwert

Bei 16 Farben zum Beispiel bewirkte ein BLINE mit Farbtafelwert 5 genau dasselbe wie ein LINE-Befehl mit Farbtafelwert 11, da $16-5=11$.

Modus 1 ist ein wenig sinnvoller. Hier wird der bereits vorhandene Farbtafelwert aus den Bitplanes ausgelesen. Bei diesem werden dann gesetzte Bits, die ebenfalls im angegebenen Farbtafelwert gesetzt sind, zurückgesetzt. Da dies relativ kompliziert ist, sollten Sie einmal mit **Listing 4** experimentieren, wenn Sie die Video RAM-Erweiterung besitzen, oder mit **Listing 5**, wenn nicht.

Das Programm fragt nacheinander den zu setzenden Farbtafelwert, den Überlagerungsmodus und eine Hintergrundbitmap ab. Diese Farbtafel für den Hintergrund sehen Sie dann in dem großen Rechteck in der Mitte. Die Linien werden mit dem BLINE-Befehl erzeugt und die Punkte sowie das kleine Rechteck mit dem RESET-Befehl. Beachten Sie, daß die Linien und das kleine Rechteck über den Rand des Hintergrundrechtecks hinaus geschrieben werden. Dies ist auch dann der Fall, wenn beides nur innerhalb des Hintergrundrechtecks zu sehen ist! Nach Erstellen der Zeichnung wartet das Programm auf einen Tastendruck. Wird hier ein großes „Q“ eingegeben, so bricht das Programm ab.

Sollten Sie noch Probleme mit den normalen Betriebsarten haben, ändern Sie in dem jeweiligen Listing einfach alle BLINES in LINES und alle RESETs in SETs und experimentieren Sie aufs Neue.

PAINT

Obwohl der PAINT-Befehl im Handbuch ziemlich vollständig beschrieben ist, will ich hier noch einiges Ergänzende hinzufügen. Die Randfarben sind die Farbtafelwerte, die die zu füllende Fläche begrenzen sollen. Es sind nicht etwa Farbwerte oder Strings, die die Farbe beschreiben.

Es ist darauf zu achten, daß die Fläche auch wirklich einen geschlossenen Kreis bildet. Ein nicht als Randfarbe deklarierter Punkt auf einer Begrenzungslinie kann dazu führen, daß der ganze Bildschirm gefüllt wird! Experimentieren Sie einmal mit den **Listings 6** und **8**, je nachdem, ob Sie im Besitz einer Video RAM-Erweiterung sind oder nicht. Erklärungen zu den Listings stehen als Kommentare im Listing selbst. **Listing 7** ist eine Abwandlung von Listing 6, die zeigt, daß nicht die Farben als Begrenzung interessant sind, sondern eben die Farbtafelwerte.

Sonstiges

Die restlichen Befehle halte ich für ausreichend beschrieben. Sollten trotzdem Probleme auftauchen, wenden Sie sich bitte an mich oder das Magazin 700/800. Benachrichtigen Sie uns bitte auch, wenn Sie weiterführende Kurse wünschen.

Ich wünsche Ihnen noch viel Spaß mit den neu erworbenen Kenntnissen und Ihrem Computer.

MGR-Software, Asgard
Lars Hanke, Schlachthofstr. 67,
D-4690 Herne 2

Einführung in die Grafik des MZ-800

```
10 REM Listing 1
20 REM *** Experimentierprogramm mit Paletten und 4 Bit-Planes ***
30 REM
40 REM Copyright (C) 1989 by MGR-Software, Asgard
50 REM written as Demo for MZ-800 with extended video RAM
60 REM Author -> Lars Hanke
70 REM rev. -> 24.05.1989 MGR, Asgard
80 REM
90 REM
100 INIT "CRT:M2":REM CLS UND 4 BIT-PLANES AKTIVIEREN
110 DIM BLK$(4)
120 BLK$(0)="B0"
130 BLK$(1)="B1"
140 BLK$(2)="B2"
150 BLK$(3)="B3"
160 LABEL "DRWNEU"
170 A$=""
180 FOR I=1 TO 15:A$=A$+" ":NEXT I
190 A$=A$+A$+"HH"
200 FOR I=15 TO 0 STEP -1
210 SYMBOL [I] 34,96,A$,1,1,0
220 A$=RIGHT$(A$,LEN(A$)-2)
230 NEXT I
240 CURSOR 0,0
250 PRINT "MGR-Software, Asgard"
260 PRINT "Paletten Experimentierprogramm"
270 PRINT
280 PRINT "Geben Sie nacheinander Block-"
290 PRINT "und Palettennummern an."
300 PRINT
310 CURSOR 4,10
320 PRINT "!0!1!2!3!4!5!6!7!8!9!A!B!C!D!E!F"
330 GOTO "DATLES"
340 LABEL "ENDE"
350 INIT "CRT:M4"
360 END
370 LABEL "DATLES"
380 CURSOR 0,15
390 INPUT "Block Nr.: ";BNR
400 IF BNR>3 THEN GOTO "ENDE"
410 CURSOR 0,16
420 INPUT "Palette 0: ";P0
430 CURSOR 0,17
440 INPUT "Palette 1: ";P1
450 CURSOR 0,18
460 INPUT "Palette 2: ";P2
470 CURSOR 0,19
480 INPUT "Palette 3: ";P3
490 INI$="CRT:M2,"+BLK$(BNR)
500 INIT INI$
510 PAL 0+4*BNR,P0
520 PAL 1+4*BNR,P1
530 PAL 2+4*BNR,P2
540 PAL 3+4*BNR,P3
550 GOTO "DRWNEU"
10000 REM ( Listing 2 )
10010 REM Unterroutine zum setzen der
10020 REM Paletten unter BASIC
10030 REM im Modus "CRT:M2" ohne CLS
10040 REM
10050 REM REL.: 1
10060 REM (C) MGR, Asgard 11.02.1986
10070 REM
10080 REM
10090 REM Aufruf erfolgt durch
10100 REM >>> GOSUB SETPAL <<<
10110 REM Die Variablen enthalten:
10120 REM >>> PANR -> Palettennummer
10130 REM >>> COLOUR -> Farbe
10140 REM
10150 REM
10160 LABEL "SETPAL"
10170 PANR=PANR AND $F
10180 COLOUR=COLOUR AND $F
10190 OUT@ $F0,INT(PANR/4)+$40
10200 PANR=(PANR AND $3)*16+COLOUR
10210 OUT@ $F0,PANR
10220 RETURN
```

Einführung in die Grafik des MZ-800

```
10 REM ( Listing 3 )
20 REM *** Experimentierprogramm mit Paletten und 4 Bit-Planes ***
30 REM
40 REM Copyright (C) 1989 by MGR-Software, Asgard
50 REM written as Demo for MZ-800 with extended video RAM
60 REM Author -> Lars Hanke
70 REM rev. -> 24.05.1989 MGR, Asgard
80 REM
90 REM
100 INIT "CRT:M2":REM CLS UND 4 BIT-PLANES AKTIVIEREN
110 DIM BLK$(4)
120 BLK$(0)="B0"
130 BLK$(1)="B1"
140 BLK$(2)="B2"
150 BLK$(3)="B3"
160 LABEL "DRWNEU"
170 A$=""
180 FOR I=1 TO 15:A$=A$+" ";NEXT I
190 A$=A$+A$+"HH"
200 FOR I=15 TO 0 STEP -1
210 SYMBOL (I) 34,96,A$,1,1,0
220 A$=RIGHT$(A$,LEN(A$)-2)
230 NEXT I
240 CURSOR 0,0
250 PRINT "MGR-Software, Asgard"
260 PRINT "Paletten Experimentierprogramm"
270 PRINT
280 PRINT "Geben Sie nacheinander Block-"
290 PRINT "und Palettennummern an."
300 PRINT
310 CURSOR 4,10
320 PRINT "10:11:12:13:14:15:16:17:18:19:A:B:C:D:IE:IF"
330 GOTO "DATLES"
340 LABEL "ENDE"
350 INIT "CRT:M4"
360 END
370 LABEL "DATLES"
380 CURSOR 0,15
390 INPUT "Block Nr.: ";BNR
400 IF BNR>3 THEN GOTO "ENDE"
410 CURSOR 0,16
420 INPUT "Palette 0: ";P0
430 CURSOR 0,17
440 INPUT "Palette 1: ";P1
450 CURSOR 0,18
460 INPUT "Palette 2: ";P2
470 CURSOR 0,19
480 INPUT "Palette 3: ";P3
490 PANR=0+4*BNR:COLOUR=P0:GOSUB "SETPAL"
500 PANR=1+4*BNR:COLOUR=P1:GOSUB "SETPAL"
510 PANR=2+4*BNR:COLOUR=P2:GOSUB "SETPAL"
520 PANR=3+4*BNR:COLOUR=P3:GOSUB "SETPAL"
530 GOTO "DATLES"
540 REM
550 REM Unterroutine zum setzen der
560 REM Paletten unter BASIC
570 REM im Modus "CRT:M2" ohne CLS
580 REM
590 REM REL.: i
600 REM (C) MGR, Asgard 11.02.1986
610 REM
620 REM
630 REM Aufruf erfolgt durch
640 REM >>> GOSUB SETPAL <<<
650 REM Die Variablen enthalten:
660 REM >>> PANR -> Palettensnummer
670 REM >>> COLOUR -> Farbe
680 REM
690 REM
700 LABEL "SETPAL"
710 PANR=PANR AND $F
720 COLOUR=COLOUR AND $F
730 OUT@ $F0,INT(PANR/4)+$40
740 PANR=(PANR AND $3)*16+COLOUR
750 OUT@ $F0,PANR
760 RETURN
```

```
10 REM ( Listing 4 )
20 REM *** Experimentierprogramm mit BLINE und RESET ***
30 REM
40 REM Copyright (C) 1989 by MGR-Software, Asgard
50 REM
60 REM written as Demo for MZ-800 with extended video RAM
70 REM Author -> Lars Hanke
80 REM rev. -> 25.05.1989 MGR, Asgard
90 REM
100 INIT "CRT:M2"
110 PRINT "MGR, Asgard:"
120 PRINT "          RESET and BLINE experiments"
130 PRINT
140 INPUT"Bitmap.....: ";A
150 INPUT"Modus.....: ";M
160 INPUT"Hintergrundmap: ";B
170 INIT"CRT:M2"
180 BOX(8,0)150,50,270,150,B
190 BLINE(A,M)0,0,329,199
200 BLINE(A,M)329,0,0,199
210 RESET(A,M)160,75
220 RESET(A,M)160,125
230 RESET(A,M)105,100
240 RESET(A,M)215,100
250 FOR Y=90 TO 110
260 FOR X=260 TO 280
270 RESET(A,M)X,Y
280 NEXT X
290 NEXT Y
300 GET A$:IF A$="" THEN 300
310 IF A$<>"Q" THEN 100
320 INIT"CRT:M4"
330 END
```

Einführung in die Grafik des MZ-800

```
10 REM ( Listing 6 )
20 REM *** Experimentierprogramm mit PAINT Befehl ***
30 REM
40 REM Copyright (C) 1989 by MGR-Software, Asgard
50 REM written as Demo for MZ-800 with extended video RAM
60 REM Author -> Lars Hanke
70 REM rev. -> 23.05.1989 MGR, Asgard
80 REM
90 REM
100 READ A$,B$,X,Y,A,B,C,D,E:REM DATEN LESEN
110 IF A$="ENDE" THEN GOTO "ENDE":REM SCHLUSS ?
120 INIT "CRT:M2":REM CLS UND 16 FARBEN AKTIVIEREN
130 IF B$="PRE" THEN CIRCLE(4,0)160,100,70,.75:REM KREIS ZUERST ?
140 LINE(2,0)0,199,319,0
150 LINE(3,0)0,0,319,199
160 LINE(6,0)0,100,319,100
170 LINE(11,0)160,0,160,199
180 IF B$="POST" THEN CIRCLE(4,0)160,100,70,.75:REM KREIS ZUM SCHLUSS ?
190 BOX(15,0)15,20,300,180:REM UEBERLAUFEN VERHINDERN
200 SET(15,0)160,100:REM KREUZ ALS UNBEDINGTE BARRIERE
210 PAINT(14,0)X,Y,15,A,B,C,D,E:REM NACH DATEN MALEN
220 GET A$:IF A$="" THEN 220:REM AUF TASTE WARTEN
230 GOTO 100:REM UND NOECHEINMAL
240 REM
250 REM SCHLUSSROUTINE
260 REM
270 LABEL "ENDE"
280 INIT "CRT:M4":REM CLS UND 80 ZEICHEN AKTIVIEREN
290 END
300 REM
310 REM *** Datenbereich ***
320 REM
330 REM Die DATA Zeilen haben folgendes Format
340 REM 1. STRING = "ENDE" => letzter Datensatz
350 REM 2. STRING = "POST" => Kreis zuletzt zeichnen
360 REM "PRE" => Kreis zuerst zeichnen
370 REM 3.+4. X,Y => Koordinaten des Ausgangspunktes
380 REM 5.-9. A-E => Randfarben; 15 -> keine Randfarbe (mehr)
390 REM
400 REM
410 DATA "", "POST", 165,90,2,9,6,11,4
420 DATA "", "POST", 165,90,2,9,6,15,4
430 DATA "", "POST", 165,90,15,9,6,15,4
440 DATA "", "PRE", 165,90,2,9,6,11,15
450 DATA "", "POST", 165,90,15,15,15,15,4
460 DATA "ENDE", "NO", 0,0,0,0,0,0,0

10 REM ( Listing 5 )
20 REM *** Experimentierprogramm mit BLINE und RESET ***
30 REM
40 REM Copyright (C) 1989 by MGR-Software, Asgard
50 REM
60 REM written as Demo for MZ-800
70 REM Author -> Lars Hanke
80 REM rev. -> 25.05.1989 MGR, Asgard
90 REM
100 INIT "CRT:M1"
110 PRINT "MGR, Asgard:"
120 PRINT "          RESET and BLINE experiments"
130 PRINT
140 INPUT"Bitmap.....: ";A
150 INPUT"Modus.....: ";M
160 INPUT"Hintergrundmap: ";B
170 INIT"CRT:M1"
180 BOX(8,0)50,50,270,150,B
190 BLINE(A,M)0,0,329,199
200 BLINE(A,M)329,0,0,199
210 RESET(A,M)160,75
220 RESET(A,M)160,125
230 RESET(A,M)105,100
240 RESET(A,M)215,100
250 FOR Y=90 TO 110
260 FOR X=260 TO 280
270 RESET(A,M)X,Y
280 NEXT X
290 NEXT Y
300 GET A$:IF A$="" THEN 300
310 IF A$<>"0" THEN 100
320 INIT"CRT:M3"
330 END
```


Einführung in die Grafik des MZ-800

```
10 REM ( Listing 7 )
20 REM *** Experimentierprogramm mit PAINT Befehl ***
30 REM
40 REM Copyright (C) 1989 by MGR-Software, Asgard
50 REM written as Demo for MZ-800 with extended video RAM
60 REM Author -> Lars Hanke
70 REM rev. -> 24.05.1989 MGR, Asgard
80 REM
90 REM
91 REM Zeile 120 geaendert und Zeilen 171-178 neu eingefuegt
92 REM
100 READ A$,B$,X,Y,A,B,C,D,E:REM DATEN LESEN
110 IF A$="ENDE" THEN GOTO "ENDE":REM SCHLUSS ?
120 INIT "CRT:M2,B2":REM CLS UND 16 FARBEN AKTIVIEREN ***!***
130 IF B$="PRE" THEN CIRCLE[4,0]160,100,70,.75:REM KREIS ZUERST ?
140 LINE[2,0]0,199,319,0
150 LINE[9,0]0,0,319,199
160 LINE[6,0]0,100,319,100
170 LINE[11,0]160,0,160,199
171 REM
172 REM Einfuegen fuer Versuch 2
173 REM
174 PAL 9,2:REM Bitmap 9 als rot (wie Bitmap 2)
175 PAL 11,2:REM Bitmap 11 als rot (dto.)
176 REM
177 REM Ende des Zusatzes
178 REM
180 IF B$="POST" THEN CIRCLE[4,0]160,100,70,.75:REM KREIS ZUM SCHLUSS ?
190 BOX[15,0]15,20,300,180:REM UEBERLAUFEN VERHINDERN
200 SET[15,0]160,100:REM KREUZ ALS UNBEDINGTE BARRIERE
210 PAINT[14,0]X,Y,15,A,B,C,D,E:REM NACH DATEN MALEN
220 GET A$:IF A$="" THEN 220:REM AUF TASTE WARTEN
230 GOTO 100:REM UND NOCHEINMAL
240 REM
250 REM SCHLUSSROUTINE
260 REM
270 LABEL "ENDE"
280 INIT "CRT:M4":REM CLS UND 80 ZEICHEN AKTIVIEREN
290 END
300 REM
310 REM *** Datenbereich ***
320 REM
330 REM Die DATA Zeilen haben folgendes Format
340 REM 1. STRING = "ENDE" => letzter Datensatz
350 REM 2. STRING = "POST" => Kreis zuletzt zeichnen
360 REM "PRE" => Kreis zuerst zeichnen
370 REM 3.+4. X,Y => Koordinaten des Ausgangspunktes
380 REM 5.-9. A-E => Randfarben; 15 -> keine Randfarbe (mehr)
390 REM
400 REM
410 DATA "", "POST", 165,90,2,9,6,11,4
420 DATA "", "POST", 165,90,2,9,6,15,4
430 DATA "", "POST", 165,90,15,9,6,15,4
440 DATA "", "PRE", 165,90,2,9,6,11,15
450 DATA "", "POST", 165,90,15,15,15,15,4
460 DATA "ENDE", "NO", 0,0,0,0,0,0,0

10 REM ( Listing 8 )
20 REM *** Experimentierprogramm mit PAINT Befehl ***
30 REM
40 REM Copyright (C) 1989 by MGR-Software, Asgard
50 REM written as Demo for MZ-800
60 REM Author -> Lars Hanke
70 REM rev. -> 25.05.1989 MGR, Asgard
80 REM
90 REM
100 READ A$,B$,X,Y,A,B,C:REM DATEN LESEN
110 IF A$="ENDE" THEN GOTO "ENDE":REM SCHLUSS ?
120 INIT "CRT:M1":REM CLS UND 4 FARBEN AKTIVIEREN
130 IF B$="PRE" THEN CIRCLE[2,0]160,100,70,.75:REM KREIS ZUERST ?
140 LINE[2,0]0,199,319,0
150 LINE[1,0]0,0,319,199
160 LINE[2,0]0,100,319,100
170 LINE[1,0]160,0,160,199
180 IF B$="POST" THEN CIRCLE[2,0]160,100,70,.75:REM KREIS ZUM SCHLUSS ?
190 BOX[3,0]15,20,300,180:REM UEBERLAUFEN VERHINDERN
200 LINE[3,0]159,100,161,100:REM KREUZ ALS UNBEDINGTE BARRIERE
210 PAINT[3,0]X,Y,3,A,B,C:REM NACH DATEN MALEN
220 GET A$:IF A$="" THEN 220:REM AUF TASTE WARTEN
230 GOTO 100:REM UND NOCHEINMAL
240 REM
250 REM SCHLUSSROUTINE
260 REM
270 LABEL "ENDE"
280 INIT "CRT:M3":REM CLS UND 80 ZEICHEN AKTIVIEREN
290 END
300 REM
310 REM *** Datenbereich ***
320 REM
330 REM Die DATA Zeilen haben folgendes Format
340 REM 1. STRING = "ENDE" => letzter Datensatz
350 REM 2. STRING = "POST" => Kreis zuletzt zeichnen
360 REM "PRE" => Kreis zuerst zeichnen
370 REM 3.+4. X,Y => Koordinaten des Ausgangspunktes
380 REM 5.-7. A-C => Randfarben; 3 -> keine Randfarbe (mehr)
390 REM
400 REM
410 DATA "", "POST", 165,90,2,1,3
420 DATA "", "POST", 165,90,2,3,3
430 DATA "", "PRE", 165,90,1,3,3
440 DATA "ENDE", "NO", 0,0,0,0,0,0,0
```

Listings von Günter Pudritz

```

10 CLS:CURSOR10,1:COLOR3,0
20 BOX(2,0)75,1,180,20
30 PRINT(2) "Zahl erraten":PRINT
40 PRINT "    Ich denke mir eine Zahl
50 PRINT "    zwischen 1 und 100
60 PRINT:PRINT"Du sollst versuchen, sie zu erraten.
70 PRINT"Ich sage dir jedes Mal, wie du mit
80 PRINT"deiner geratenen Zahl liegst.
90 PRINT
100 L=INT(RND(1)*100+1)
110 G=0
120 PRINT:PRINT"O.K., ich habe eine Zahl. - fang" an
130 COLOR1,0
140 PRINT:INPUT"Welche Zahl habe ich gewaehlt? ":A
150 G=G+1:COLOR3,0
160 IFA=LGOTO210
170 IFA>LGOTO200
180 IFA<LGOTO190
190 PRINT"Zu niedrig. Versuche eine hoehere Zahl".GOTO130
200 PRINT"Zu hoch. Versuche eine tiefere Zahl":GOTO130
210 PRINT:PRINT"RICHTIG!!! Du hast"G"mal geraten.
220 IFG=5GOTO250
230 PRINT"Du solltest aber hoechstens
240 PRINT"5 Mal raten muessen!":GOTO260
250 PRINT"bravo!!!
260 PRINT
270 PRINT"Moechtest du noch einmal spielen? "
280 GET A$:IF A$=""THEN GOTO 260
290 IFA$="J"THENCLS.GOTO100
-
100 CLS:CURSOR1,0
110 COLOR3,0:END
110 PRINT"Berechnung von Ostern und Pfingsten
120 PRINT "    fuer die Jahre 1700 bis 2199
130 PRINT:INPUT"Jahr (vierstellig) ":J
140 B$=""Y=0
150 IF(J<1700)+(J>2199)THEN PRINT"Denkste!":GOTO130
160 IF(J<1699)*(J<1800)THEN M=23:N=3
170 IF(J<1799)*(J<1900)THEN M=23:N=4
180 IF(J<1899)*(J<2100)THEN M=24:N=5
190 IF(J<2099)*(J<2200)THEN M=24:N=6
200 X=INT(J/19):A=J-19*X
210 X=INT(J/4):B=J-4*X
220 X=INT(J/7):C=J-7*X
230 X=INT((19*A+M)/30):D=(19*A+M)-30*X
240 X=INT((2*B+4*C+6*D+N)/7):E=(2*B+4*C+6*D+N)-7*X
250 X=22+D+E:IFX>31GOTO270
260 A$="Maerz":B$="Mai":Y=X-12:GOTO320
270 A$="April":X=D+E-9:IFX=26THENLETX=19
280 IF(X=25)AND(D=28)AND(A=10)THEN LETX=18
290 IFX=12GOTO310
300 B$="Mai":Y=19+X:GOTO320
310 B$="Juni":Y=X-12
320 PRINT:PRINTJ:" Ostern = ";STR$(X):", ";A$
330 PRINT "    Pfingsten = ";STR$(Y):", ";B$
340 PRINT:PRINT"Nochmal: "
350 GETJ$:IFJ$=""GOTO350
360 IFJ$="J"GOTO130:ELSEEND

```

Listings von Günter Pudritz

```

100 CLS:SYMBOL(2)20,10,"PLANETENLANDUNG",2,2,0
102 CURSOR0,4:PAL1,3
104 PRINT" Sie können eine Landung auf einem der
106 PRINT"unten angegebenen Himmelskörper machen.
108 PRINT:PRINT"Die Spritmenge zum Bremsen liegt
110 PRINT" zwischen 0 und 100 Liter.
111 PRINT:COLOR1,0
112 PRINT"Bitte Himmelskörper wählen:
114 PRINT"1 = Erde
116 PRINT"2 = Mond
118 PRINT"3 = Mars
120 PRINT"4 = Vesta
125 COLOR3,0
130 GETX:IF(X<1)+(X>4)THEN130
140 ONXGOTO500,600,700,800
145 CLS:CONSOLE3,22
147 CURSOR10,0:PRINT(1)"Landung auf ";A$
150 G2=G1/36
160 G3=SQR(G2)*100:G3=INT(G3):IFG3<175THENG3=175
170 G4=G3*55:G4=INT(G4):IFG4<10000THENG4=10000
180 G5=G4*(LOG(G1)/20)+10000
190 A1=-6400:A2=5000:A3=15000:A4=10
200 B4=A4:B2=A2:N3=G3:N4=G4
202 CURSOR0,1:COLOR1,0
205 PRINT"Zeit Höhe Geschw. Sprit Impuls
206 CURSOR0,2
207 PRINT"(sec) (km) (km/h) (l) (l/sec)
208 COLOR3,0
210 T$=STR$(T1)
211 IFN3<1LETX3=INT(N3*1000)/1000:GOTO220
212 IFN3<10LETX3=INT(N3*100)/100:GOTO220
213 IFN3<100LETX3=INT(N3*10)/10:GOTO220
214 X3=INT(N3)
220 IFB2>1000LETX2=INT(B2):GOTO230
225 X2=INT(B2*10)/10
230 PRINTT$TAB(5)X3TAB(12)X2TAB(21)N4TAB(30);
240 INPUTF
250 IFF=0GOTO280
260 IF(F<0)+(F>100)GOTO320
270 T=N4/F:IFT<10THENB4=T
280 N4=N4-(F*B4)
285 V1=B3
286 T1=T1+B4
290 B5=(G2+((G2*N3)/(A5*-2)))-((F*G5)/(A3+N4))
295 B3=B2+(B5*B4)
298 N5=N3
300 N3=N3+(((B3+B2)/A1)*B4)
305 B2=B3
307 IFN3<0GOTO450:'gelandet
310 IFN4<=0GOTO400:'Sprit alle
315 GOTO210
320 BEEP:PRINT(2)"-->> Spritmenge nur 0 - 100":GOTO210
400 V2=SQR(B2^2+N3*G2*5650):PRINT:PRINT(1)"Sprit alle nach";T1;" Sekunden
410 V3=ABS(V2)*10000/3600
420 T1=T1+LOG((V3*N3*10000)/G1)
430 GOTO1000

```

5010 PRINT"Benötigte Zeit : ";T1;" Sekunden"
5015 COLOR3,0:PAL2,2:END

450 V2=SQR(ABS(N5/(26*B5)))*(26*B5)+V1:GOTO1000
460 T1=T1-(10-B4)
499 END
500 G1=980.7:A5=6371:A\$="Erde":GOTO145
500 G1=162:A5=1738:A\$="Mond":GOTO145
600 G1=374:A5=3380:A\$="Mars":GOTO145
800 G1=17.5:A5=195:A\$="Vesta":GOTO145
1000 COLOR2,0:PRINT:B\$="Sie sind "
1010 IFV2<20PRINTB\$;"gelandet":GOTO1100
1020 IFV2<100PRINTB\$;"abgestürzt":GOTO1140
1030 IFV2<250PRINTB\$;"mausetot":GOTO5000
1035 PRINT "Sie haben "
1040 IFV2<5000PRINT"einen neuen Krater erzeugt":GOTO5000
1050 IFV2>4999PRINT"den Planeten durchbohrt":GOTO5000
1100 IFV2<1PRINT"sanfte Landung - sehr gut":GOTO5000
1110 IFV2<5PRINT"Sie sind wohl Anfänger?":GOTO5000
1120 PRINT"Brutale Landung":GOTO5000
1140 IFV2<30PRINT"Sie können nicht mehr starten":GOTO5000
1150 IFV2<45PRINT"Sie sind verletzt, die Rakete brennt":GOTO5000
1160 PRINT"Es gibt keine Ueberlebenden"
5000 PRINT"Geschwindigkeit: ";INT(ABS(V2)*10)/10;" (km/h)

Listing von Oliver Braun

```
10 CLS
20 PRINT "VERSTECKSPIEL  HURKLE  "
30 PRINT
40 PRINT "AUTOR:OLIVER BRAUN  DATUM:6.2.89"
50 PRINT "===== "
60 PRINT:PRINT
70 N=10:G=10
80 PRINT "HURKLE IST IN EINEM KOORDINATENFELD"
90 PRINT "VON ";N;"MAL ";G;" PUNKTEN VERSTECKT"
100 PRINT "JEDER PUNKT WIRD DURCH ZWEI  "
110 PRINT "GANZE ZAHLEN BESTIMMT,"
120 PRINT "DIE DURCH EIN KOMMA GETRENNT SIND"
130 PRINT "AUSGANGSPUNKT IST DER PUNKT 0,0"
140 PRINT "VERSUCHE ZU ERRATEN,IN WELCHEM PUNKT"
150 PRINT "SICH HURKLE VERSTECKT HAT."
160 PRINT "ICH SAGE DIR  DANN;"
170 PRINT "IN WELCHE RICHTUNG DU GEHEN MUßT,"
180 PRINT "UM HURKLE ZU FINDEN.  VIEL GLÜCK !!!"
190 PRINT
200 FOR W=1 TO 10000:NEXT W
210 PRINT "BIST DU BEREIT -J-  " : INPUT A$
220 IF A$= "J" THEN 230 :ELSE 210
230 CLS
240 A=INT(N*RND(1))
250 B=INT(G*RND(1))
260 FOR K=1 TO 5
270 PRINT "RATE MAL, WO ES IST ?"
280 INPUT X,Y
290 IF ABS(X-A)+ABS(Y-B)=0 THEN 440
300 GOSUB 480
310 PRINT
320 NEXT K
330 PRINT "DAS WAR DEIN LETZTER VERSUCH"
340 PRINT "HURKLE HATTE SICH IN ";A;"","B" VERSTECKT !"
350 PRINT
360 PRINT "VERSUCHE ES DOCH NOCH EINMAL."
370 PRINT "HURKLE HAT SICH SCHON VERSTECKT !"
380 PRINT
390 PRINT "BIST DU BEREIT -J-  ",:INPUT B$
400 IF B$= "J" THEN 230
410 IF B$= "N" THEN CLS : PRINT"SCHADE":BEEP:BEEP:BEEP:END
420 GOTO 240
430 PRINT
440 PRINT "DU HAST HURKLE BEIM ";K;". "
450 PRINT"VERSUCH ENTDECKT !"
460 PRINT
470 GOTO 350
480 PRINT "EIN BISCHEN MEHR NACH  "
490 IF Y=B THEN 540
500 IF Y<B THEN 530
510 PRINT "SUED  "
520 GOTO 540
530 PRINT "NORD  "
540 IF X=A THEN 590
550 IF X<A THEN 580
560 PRINT "WEST  "
570 GOTO 590
580 PRINT "OST  "
590 PRINT
600 RETURN
610 END
```

256 Farben

Das folgende Programm soll eine Möglichkeit aufzeigen, mehr als nur 16 Farben mit dem MZ-800 darzustellen. Durch Mischung von zwei Farben erhält man eine neue. Aber wie mischt man zwei verschiedene Farben? Das Prinzip habe ich aus dem Maus-Programm abgeguckt, in dem es gut realisiert wurde.

Man setzt einfach zwei Pixel so dicht nebeneinander, daß sie für das Auge zu einer neuen Farbe verschmelzen. Wenn man das auf großer Fläche macht, ist die Täuschung nahezu perfekt. Es ist aber zu sagen, daß man dann selbstverständlich nicht mehr 320*200 Pixel hat, sondern nur noch 160*200. Aber das reicht ja für manche Zwecke auch noch gut aus.

Der Programmablauf sollte jedem klar sein (-Zeile 150 Bildschirmaufbau, -260 Abfrage der Tasten und entsprechende Farbveränderung). Das Programm läuft nur im Modus M2, also nur mit der Graphik-Erweiterung. Wer keine Graphik-Erweiterung besitzt, sollte sich schleunigst eine kaufen oder kann die Zeile 40 ändern in: INIT"CRT:M1". Es werden jetzt nur noch 16 verschiedene Farbkombinationen erzeugt, die allerdings nicht identisch sind mit den 16 SHARP-Farben. Wenn man mit PAL die einzelnen Farbenwerte verändert, so ergeben sich neue Kombinationen (es lassen sich also 16 aus 256 Farben auswählen, nicht mehr nur 4 aus 16).

Das Programm ließe sich in anderen Programmen nur schlecht benutzen, wegen der hohen Ausführungszeit. Diese Idee weiterzuführen wäre nur dann sinnvoll, wenn jemand diese Routinen in ML übersetzt. Sehr gut wäre ein Befehl PAINT für Farben. Meiner Ansicht nach wäre das ganz klar eine Bereicherung für unseren Rechner.

```
40 MO=15:INIT"CRT:M2"
50 CLS:COLOR MO,0:PAL
   1,1:Z1=1:PAL 2,5:Z2=5
```

```
60 SYMBOL 10,10,"SIMULATION
   VON 256 VERSCHIEDE-
   NEN",1,2
61 SYMBOL 50,25,"FARBEN AUF
   DEM MZ-800",1,2
70 BOX 100,55,120,65:BOX
   40,55,160,65
80 SYMBOL 105,70,"1 2",1,1
90 BOX 100,100,161,130
100 SYMBOL 103,135,"1 +
   2",1,1
110 PAINT [1] 101,56,MO
120 PAINT [2] 141,56,MO
130 FOR I=101 TO 129 STEP 2
131 FOR T=101 TO 159 STEP 2
132 SET [1] T,I:SET [2]
   T+1,I
133 NEXT T,I
140 FOR I=102 TO 129 STEP 2
141 FOR T=10 TO 159 STEP 2
142 SET [2] T,I:SET [1]
   T+1,I
143 NEXT T,I
160 CURSOR 0,19:PRINT"FARBE
   1 UND 2 AENDERN MIT >1<
   UND >2<!"
170 GET A$:IF A$="" GOTO
   170
180 IF A$ <> „1" GOTO 220
190 SOUND 34,1:Z1=Z1+1
200 IF Z1 > MO THEN Z1=0
210 PAL 11
220 IF A$ <> „2" GOTO 260
230 SOUND 44,1:Z2=Z2+1
240 IF Z2 > MO THEN Z2=0
250 PAL 2,Z2
260 GOTO 170
(Konvertiert von Edgar Lefgrün)
```

Funktionstastenbelegung in Textvariablen gespeichert

Manchmal kann es sehr nützlich sein, den Text der Funktionstasten vom Programm aus abfragen zu können, z.B. um den ursprünglichen Zustand am Ende eines Programms wiederherzustellen, wenn sie während eines RUN's neu belegt wurde. Die folgende Routine liest den Inhalt der 10 Tasten in das Array F\$(1...10).

Nach dem Ablauf (höchstens 2,5 sec.) können die Daten leicht ver-

waltet werden. Der Speicherbereich für diese 10 Tasten beginnt (im Basic 22046) bei \$12B2. Dort ist die Länge des Inhalts von F1 gespeichert. Es folgen 15 Daten; ist der String kürzer, so sind die übrigen Bytes entweder gefüllt mit \$0 oder mit anderen Zeichen, die von einer vorhergehenden Belegung stammen. Dieses ist für alle Funktionstasten gleich. Der Speicherbereich endet bei \$1351. Es ergibt sich automatisch die Möglichkeit, die Länge der Tastentexte zu manipulieren, ohne deren Inhalt zu löschen. Es ergibt sich folgende Formel: POKE\$12B2+(Funktionstaste-1)*16,Wert. Folgender POKE würde die Länge des Strings von F7 auf 3 Zeichen kürzen: POKE\$12B2+(7-1)*16,3. Probieren Sie ein bißchen damit herum, um sich damit vertraut zu machen. Eins ist zu beachten: Der Wert darf nie größer als 15 sein! Leider liegen die Speicherbereiche bei (fast) jedem Basic anders.

So wie das Listing abgedruckt ist, läuft es nur im Basic 22046, 52009 und 12016. Für andere Basic's muß (sofern aufgepäht) die Zeile 30 geändert werden in: Basic 52008: 30 FOR AD=\$12C0 TO \$1350 STEP 16, Basic 12013: 30 FOR AD=\$1322 TO \$13B2 STEP 16. Wenn Fragen oder Probleme auftauchen sollten, stehe ich gerne zur Verfügung. Jan Seng, Inbogenweg 3, 6144 Zwingenberg, 06251/77941

```
10 DIM F$(10)
20 NU=1
30 FOR AD=$12B2 TO $1343
   STEP 16
40 AN=PEEK(AD)
50 IF AN=0 GOTO 100
60 FOR OF=1 TO AN
70 F$(NU)=F$(NU)+CHR$(PEEK
   (AD+OF))
80 NEXT
90 NU=NU+1
100 NEXT
(EL-Konvertiert)
```

Bemerkung: Bei den 800er Sharp-Basicinterpretern gilt für die F-Tasten der gleiche Pegel.

Aufnahme von Daten mit RS 232C

Aufnahme von Daten mit der Schnittstelle RS 232C

Nachdem ich im Heft 5/6/88 eine Möglichkeit beschrieben hatte, wie man einen Analog/Digital-Wandler am MZ-700 betreiben kann, soll heute die dazu passende Software zur Erfassung der Daten vorgestellt werden. Zur Erinnerung: Eine aus Heft Elektor 5/85 aufgebaute Platine wurde an den Systembus des MZ-700 problemlos angepaßt. Mit dem MZ-800 funktioniert es natürlich genau so, nur die Verbindung ist eine andere. Die Kosten waren mit gut DM 50,- vergleichsweise gering.

Die Schaltung Platine kann dabei bis zu 8 analoge Spannungswerte (fast) gleichzeitig aufnehmen. Wem das nicht reicht, der sollte sich mit der im selben Heft vorgestellten Multiplex-Schaltung befassen.

Spannungswerte können von den verschiedensten Gebern erzeugt werden, im letzten Heft hatte ich Beispiele gegeben für eine Sauerstoffsonde (im Wasser gelöster Sauerstoff), Temperatur und Globalstrahlung. Moderne Meßgeräte arbeiten heute fast alle elektronisch und enthalten einen entsprechenden Geber, den wir mit einer kleinen, abgemagerten Schaltung dazu bringen können, die gewünschten Werte zu liefern. Schaltungsanwendungen sind meistens bei den Herstellern zu erhalten.

Ansprechen und Auslesen der Daten vom Port des MZ-700 ging mit dem S-BASIC so:

```
10 CLS
20 REM Prüfen der Eingänge
30 GET X$
40 REM Die Eingänge werden durch die entsprechende Taste des Computers angewählt
50 IF X$ = „" GOTO 50
60 X = ASC(X$) - 49
70 OUT#16, X
80 INP#16, A
```

```
90 PRINT A
100 GOTO 30
```

Die Variable A ist natürlich sofort weg, sobald sie durch das Auslesen einer neuen überschrieben wird. Also wird man mit einer indizierten Variable arbeiten, die man im RAM halten und später abspeichern kann.

Anhand eines konkreten Beispiels will ich aufzeigen, wie man die Daten erfassen, aufzeichnen und später mit dem Programm UNI-PLOT auf das Papier bringen kann.

Das Programm (LISTING 2) dient dazu, Globalstrahlung und Temperatur zu erfassen. Was das ist und wie die Diagramme aussehen, kann man in Heft 5/6/88 auf Seite 24 nachlesen.

Die Variablenamen sind dieselben wie im Programm UNI-PLOT.

1080 TY ist die Anzahl der Meßwerte (s.u.)

1090 WX(I) sind die Werte für Globalstrahlung, XW(I) die für die Temperatur. Diese beiden Naturgrößen nehme ich mit Sensoren auf, und zwar alle 10 Minuten (s.u.)

1100-1110 Eingabe der Uhrzeit. S-BASIC erlaubt es nicht, die Systemvariable TI\$ mithilfe der INPUT-Anweisung einzulesen. Das macht man auf dem Umweg über TS und Zuweisung in Zeile 1110.

1120 Da die Messungen um 0 Uhr eines Tages beginnen sollen, hängt sich der Rechner in Zeile 1140 zunächst auf, bis dieser Zeitpunkt erreicht ist.

1150 Im Programm muß der 'Klammeraffe' (@) statt des Dollarzeichens eingegeben werden. Kanal 0 des A/D-Wandlers wird angewählt.

1160 Einlesen des Meßwertes, E genannt.

1170 Die gemessene Spannung wird in den korrekten Wert der Globalstrahlung umgerechnet und der Feldvariablen WX(I) zugewiesen.

1180-1200 Dasselbe geschieht auf Kanal 3 mit der Temperatur.

1210 Anzeige der Werte zur Kontrolle.

1220 Warteschleife, um zu vermeiden, daß in derselben Minute zweimal gemessen wird.

1230 Mit dieser Bedingung wird erreicht, daß nur alle 10 Minuten eingelesen wird.

1240 Rücksprung, wenn Abbruchbedingung noch nicht erreicht ist.

1250-1270 Öffnen der Kassettendatei

1290 Dient nur der Aufzeichnungskontrolle. Bei kürzeren Meßzeiträumen kann man die Werte auch mit PLOT ON mitprotokollieren. Der Monitor kann dann entfallen. Bei einem langen Meßzeitraum würde der Plotterstift zu sehr einfrocknen.

1300-1320 Schreiben der Meßwerte auf das Band.

Mit diesem Programm werden jeden Tag 100 Meßwertpaare aufgezeichnet, d.h. bei einer Dimensionierung von 3500 also etwa 24 Tage. Mehr geht nicht!

Man könnte natürlich die Werte auch bei geöffneter Datei kontinuierlich wegschreiben (ohne Feldvariablen). Dann ist aber der Recorder ständig in Betrieb, Andruckwalze und Band werden belastet.

In Zeile 2050 und 2060 verwende ich die PRINTUSING-Anweisung, da ich nur eine Nachkommastelle an Genauigkeit brauche. Außerdem re-

Aufnahme von Daten mit RS 232C

duziert sich die Länge der Datei auf gut 9 kByte. Schreibt man die Werte unformatiert auf, bläht sich die Sache auf reichlich 33 kByte auf. Damit ist auch eine C 90-Kassette beinahe schon am Ende, denn Dateien werden ja im ASCII-Format und nicht im SHARP-Code aufgezeichnet. Das kann man gut hören, weil der Kassettenrecorder nach Aufzeichnung jedes Blockes kurz anhält.

Wer seine Floppy während der Meßphase nicht braucht, kann diese selbstverständlich zur Aufzeichnung verwenden. Nur sollte man dann keine sequentiellen-Dateien verwenden. Die Diskette läuft nach Öffnen eines BSD-Files durch WOPEN nämlich ständig, solange der Befehl wirksam ist. Und schließen kann man die Datei bis zur nächsten Messung auch nicht, weil man eine sequentielle Datei nicht einfach wieder öffnen kann, um Daten anzuhängen.

Also verwendet man eine RANDOM-Datei, muß allerdings später die Daten wieder in ein BSD-File umsetzen.

Gehen wir also davon aus, daß alle 7000 Daten auf der Kassette sind. Zur besseren Verarbeitung werden sie am besten erst einmal auf die Diskette übertragen. Für diese Operation kann man entweder BASIC benutzen oder eines der vielen Utilities, wie TRANS, DISK-FILER, COPYCMT usw.

Schwierigkeiten kann es geben, wenn man die Kassette mit dem MZ-700 aufgezeichnet hat, den Transfer aber mit dem MZ-800 vornehmen will. Unter Umständen können die Daten nicht eingelesen werden. Das S-BASIC verwendet einen ins RAM kopierten Monitor mit 1400 Baud, der 800er dagegen ausschließlich den ROM-Monitor mit 1200 Baud, auf den auch die meisten Utilities zugreifen. Am besten mal alles durchprobieren. Sonst

den Transfer mit dem 700er vornehmen.

Ist das geschehen, laden wir Disketten-BASIC MZ-22049 und ein Programm, mit dem wir aus der Gesamtdatei zwei machen, nämlich eine mit den Globalstrahlungswerten und eine andere mit den Temperaturen. Diese sollen ja auch getrennt geplottet werden; außerdem passen unter Disketten-BASIC so viele Daten nicht mehr ins RAM.

LISTING 3 zeigt, wie es gemacht werden kann.

Mit dem Programm UNI-PLOT kann nun die gewünschte Datei eingelesen werden. Man braucht nach Erscheinen des Directory dazu nur mit den Cursor in die entsprechende Zeile hochzufahren und bestätigt mit Return.

Beim Versuch, die Daten auszuploten, wird man zunächst scheitern, weil einige Parameter noch auf Null gesetzt sind. Man kann sie natürlich mithilfe des Programms LISTING 2 auch mit abfragen und in die Datei mit aufnehmen. Ich mache das grundsätzlich nicht, damit eine Datei immer eine reine Sammlung gleichartiger Daten bleibt, die sich später besser mit Turbo-PASCAL bearbeiten läßt und weil ich sie auch auf dem AT-Rechner unter MS-DOS weiter bearbeiten muß.

Nach Einlesen durch die Menü-Option 4 bricht man das Programm zunächst ab und gibt die gewünschten Zusätze im Direkt-Modus ein, z.B.

```
TY=3500 (Anzahl Meßwerte)
SY=1 (Schriftweite X-Achse)
TX=30 (max Wert in Y-Richtung)
SX=100 (Schriftweite X-Achsenbeschriftung)
BX$="E/klx" (Bezeichnung Y-Achse)
BY$="Tage" (Bezeichnung X-Achse)
UE$="GLOBALSTRAHLUNG" (Gesamtüberschrift)
```

Es ist ein Vorteil eines Interpreters gegenüber einem Compiler, daß Programme an jeder Stelle unterbrochen, Änderungen vorgenommen und fortgesetzt werden können. Bei der Entwicklung von Plotter-Programmen spart das außerdem viel Papier und Zeit.

Wer über den S-BASIC-Compiler verfügt, der kann die Meßdatenumenge noch erhöhen. In das Quellprogramm habe ich eine kleine Abfrage eingebaut. So kann man sich nach Starten des gut 19 kByte langen Maschinenprogramms langsam an die Obergrenze herantasten. Achtung! In LISTING 2 muß die Warteschleife in Zeile 1220 bei Verwendung des Compilers vergrößert werden, weil sie schneller durchlaufen wird.

Dirk Grube, Parnaßweg 7,
D-2320 Plön, ☎ 0 45 22/20 17

```
1000 ' LISTING 3
1010 '
1020 ' Programm teilt die
      Datei mit einem
1030 ' Messwertpaar in zwei
      einzelne auf
1040 '
1050 ' fuer BASIC MZ-22009
1060 '
1070 CLS:CLR
1080 DIR
1090 INPUT" Datei:" ;NA$
1100 ROPEN#1,NA$
1110 WOPEN#2,"GLOBAL"
1120 WOPEN#3,"TEMPERATUR"
1130 INPUT#1,WX,XW
1140 '
1150 PRINT#2,USING"##.#",WX
1160 PRINT#3,USING"##.#",XW
1170 IF NOT EOF(#1) GOTO
      1130
1180 CLOSE
1190 END
```

☛ Listing 2 finden Sie auf Seite 27!

Wordstar anpassen

```
1000 ' LISTING 2
1010 '
1020 ' Programm nimmt
      Messdaten auf und
      zeichnet
1030 ' sie nach Abschluss
      auf Kassette auf
1040 '
1050 ' fuer S-BASIC
1060 '
1070 CLS
1080 INPUT" Anzahl der
      Messungen: ";TY
1090 DIM WX(TY),XW(TY)
1100 INPUT" Zeit: ";T$
1110 TI$=T$
1120 B$="235959"'Beginn der
      Messungen
1130 CLS
1140 IFTI$<B$THEN 1140
1150 OUT@16,0
1160 INP@16,E
1170 E=E/51*6;WX(I)=E
1180 OUT@16,3
1190 INP@16,T
1200 T=T*.2222;XW(I)=T
1210 PRINTI;" ";TI$;WX(I);
      XW(I);I=I+1
1220 FORS=1TO140:NEXT
1230 IF (RIGHT$(TI$,3)<>
      "000") THEN 1230
1240 IF I<=TY THEN GOTO
      1150
1250 INPUT"? Dateiname ";
      NA$
1260 PRINT:PRINT CHR$(127);
      " RECORD.PLAY
1270 WOPEN NA$
1280 FORI=0TOTY
1290 PRINTI;
1300 PRINT/TUSING"##.#",WX
      (I)
1310 PRINT/TUSING"##.#",XW
      (I)
1320 NEXT
1330 CLOSE
1340 END
```

WORDSTAR anpassen

Von Edgar Lefgrün

In den letzten Wochen wurde der Autor mehrmals um Hilfe beim Anpassen von Wordstar 3.0 und Drucker aneinander auf dem SHARP MZ-800 unter CP/M-80 gebeten. Probleme bereiteten Unterstreichen, Durchstreichen, Fettdruck, Doppeldruck, alternative Schreibdichte, Exponentialschrift und Indexschrift. Das war aktueller Anlaß genug, den Drucker, das CP/M und den Drucker-Patch-Bereich des Programms einmal unter die Lupe zu nehmen und das Resümee niederzuschreiben.

Die Versuchsausdrucke wurden auf dem epsonkompatiblen BROTHER M-1109 des Autors ausgeführt und auch auf Anderen wie EPSON LQ-800 überprüft, die Patcharbeiten mit dem DISKEDIT.COM des CP/M durchgeführt. Dabei war eine Besonderheit im Verhalten von DISKEDIT zu beachten: Dateien aller Art, mit F/ile: Name und R/andom oder mit B/lock: und R/andom aufgerufen, werden ab 0000H gezählt. Im Speicher beginnt der Adreßpegel ab 0100H. Beim direkten Zugriff mit S/ektor wird jeder RECORD(Eintrag) ab 0000H gezählt. Die nachstehend erwähnte Tabelle des bereits erwähnten Drucker-Patch-Bereiches steht im RECORD 11D des Programms. Die Texte zur Prefix AP-Gruppe sind ab RECORD 33 in englischer und ab RECORD 36 in deutscher Sprache der WSMMSG.S.OVR-Datei zu finden.

Wagenrücklauf und Zeilenvorschub vom Betriebssystem

Bei den Funktionen Unterstreichen und Durchstreichen geschieht dies jeweils um eine Zeile zu tief. Beim Fett- und Doppeldruck kommt es zu einem zwei- oder dreimaligen Ausdruck in untereinanderlegenden Positionen. Hierbei liegt die Ursache nicht bei WORDSTAR, sondern am Drucker und am Betriebssystem.

Am Drucker steht der DIP-Schalter für CR (Auto LF) wahrscheinlich auf „ein“. Deshalb wird bei jedem (CR)Wagenrücklauf automatisch ein (LF)Zellenvorschub erzeugt. Weiterhin steht möglicherweise im der Abteilung PRINTER MODE des SETUP der Zeiger nicht auf ALL OUT. Nach dem Ausschalten beider erwähnten Möglichkeiten wurden bei der Erprobung die angesprochenen Funktionen einwandfrei ausgeführt.

(a) Positions-Überdruck-Methode (position-overstrike-method (POSMTH))

Beim Fettdruck und beim Doppeldruck wird jedes Zeichen mehrmals gedruckt. Der Fettdruck erfolgt durch einfaches Überdrucken. Beim Doppeldruck wird durch eine Verschiebung der Druckposition das Ausfüllen der Räume zwischen den Punkten des ersten Druckes bewirkt.

Zum Ausführen der Funktionen gibt es zwei grundsätzliche Methoden; die Wagenrücklaufmethode (CRM) und die Back-Space-Methode (BSM). Bei der CRM wird ein Text vom ersten bis zum zweiten Steuerzeichen gedruckt, gefolgt von einem Wagenrücklauf bis zur imaginären Position des ersten Steuerzeichens, danach eine Druckwiederholung bis zur imaginären Position des zweiten Steuerzeichens. Bei der BSM wird ein Zeichen, gefolgt von einem einspaltigen Rücklauf (Backspace) und einer Wiederholung, gedruckt. Die Methode wird durch den Wert eines Datums in der Tabelle des Drucker-Patch-Bereiches des Textprogrammes bestimmt:

POSMTH (0690H) 0590H, FFH oder 00H

Das Label POSMTH stellt eine symbolische Adresse im WS-Quelltext dar. Im Speicher ist es Adresse 0690H, auf der Diskette Position 0590H. Der Wert des Datums beträgt FFH für die CRM und 00H für die

Wordstar anpassen

BSM. Beim Fettdruck (boldface oder boldstrike (BLDSTR)) und beim Doppeldruck (doublestrike (DBLSTR)) ist die Anzahl der Druckwiederholungen durch die Daten in den angegebenen Positionen einstellbar:

BLDSTR (0691H) 0591H: 02H
DBLSTR (0692H) 0592H: 02H

(b) Wagenrücklauf und Zeilenvorschub von WORDSTAR

Die modernen Drucker bieten die Möglichkeit der Voreinstellung der Größe des Zeilenvorschubes und dessen Teilung. Am Drucker ist die Vorschubgröße mittels DIP-Schalter meist zwischen 1/6 und 1/8 Zoll wählbar oder wird beim Einschalten des Druckers fest eingestellt. Ein Vorschub sollte für WS 1/6 Zoll betragen. 1/8 Zoll bewirkt eine unschöne Platzverteilung auf dem Papierausdruck.

Die Teilung des Vorschubes geschieht in der Regel in n/216 Zoll. Die Teilung 36/216 stellt mit 1/6 Zoll einen Vollzellen-, 18/216 einen Halbzeilenvorschub dar. Die Zeichenkette zur Teilung wird an anderer Stelle besprochen. Für einen ganzen Zeilenvorschub ist deshalb das Vielfache eines Teilvorschubes auszugeben. Die Daten sind entsprechend in der Tabelle bei der symbolischen Adresse PSCRLF (print string for carriage return and line feed) einzutragen.

PSCRLF (0696H) 0596H: 02, 0D, 0A, 00, 00 bei 36/216 Teilung
PSCRLF (0696H) 0596H: 04, 0D, 0A, 0D, 0A bei 18/216 Teilung

(c) Wagenrücklauf

Für die unter (a) erwähnten Funktionen ist ein CR ohne LF erforderlich. Die dazu benötigte Zeichenkette steht ab PSCR (print string for carriage return). Der Anhang 00 dient einer nicht erklärten Ablaufverzögerung.

PSCR (06A1H) 05A1H: 02, 0D, 00

(d) Halbzeilenvorschub

Die Benutzung des unter (b) erwähnten Halbzeilenvorschubes ist auch für Anwendungen der Exponential (hochgestellter)- oder Index (tiefgestellter)-Schrift bei Druckern mit der Fähigkeit des halbzeiligen Abwärtsschreitens, nicht aber des Aufwärtsschreitens möglich. Sinnvoll ist dies bei Geräten, denen die Fähigkeit zum Drucken von Zeichen halber Größe fehlt. Das betrifft viele ältere Matrixdrucker aber auch Typenradrucker, für die dieser Vorschub eigentlich eingerichtet wurde. Die Zeichenkette ist unter PSHALF (print string for half line) abgelegt. Sie besitzt abstrakt die halbe Länge von PSCRLF.

PSHALF (06A8H) 05A8H: 02, 0D, 0A, 00, 00 bei Halbzeilenvorschub
PSCRLF (0696H) 0596H: 04, 0D, 0A, 0D, 0A bei Halbzeilenvorschub

(e) Spaltenrückschrift

Für die Anwendung der in (a) erwähnten Backspace-Methode ist die Ausgabe einer in PBACKS (print backspace) gespeicherten Zeichenkette an den Drucker nötig. Bei modernen Druckern ist die Kette einstellig und das Datum hat den Wert 08H. Vorstellbar sind jedoch auch Exoten, die andere Daten verlangen oder mit der Fähigkeit zu Halbspaltenrückschriften, die unter Umständen das gleiche Datum mehrmals verlangen.

PBACKS (06AFH) 05AFH: 01, 08

(f) Alternative Schriftart

Mit ^PA kann eine andere Schriftart wie Zeichenhervorhebung, Schmaldruck oder Breiindruck eingestellt werden. Die Zeichenkette zur Umstellung des Druckers ist in PALT (print alternate) abgespeichert. Die Beispielenkette bewirkt ein Hervorheben der Zeichen.

PALT (06B5H) 05B5H: 02, 1B, 45

(g) Normale Schriftart

Mit ^PN wird von der alternativen in die normale Schriftart gewechselt. Die Zeichenkette ist ab PSTD (print standard) gespeichert. Das Beispiel zeigt die Kette zum Rücksetzen der Zeichenhervorhebung.

PSTD (06BAH) 05BAH: 02, 1B, 46

(h) Exponentialschrift

Mit einem ersten ^PT wird die Zeichenkette ROLUP (roll up), mit einem zweiten ^PT die Zeichenkette ROLDOW (roll down) an den Drucker übermittelt. ROLDOWN setzt den Drucker aber nicht den Normalzustand zurück. Dazu wird die Kette in USR 1 vorgeschlagen.

ROLUP (06BFH) 05BFH: 03, 1B, 53, 00

(i) Indexschrift

Mit einem ersten ^PV wird die Zeichenkette ROLDOW, mit einem zweiten ^PV die Zeichenkette ROLUP ausgegeben. Der Drucker muß mit USR 1 in den Normalzustand gesetzt werden.

ROLDOW (06C4H) 05C4H: 03, 1B, 53, 01

Kommentar:

Das Setzen und Rücksetzen von Exponential- und Indexschrift ist hinsichtlich des logischen Schaltwerkes im Programm unglücklich gelöst, da ^PT und ^PV die Ausgabe der gleichen Zeichenketten bewirken können. Eine vernünftige Alternative wäre die Ausgabe von zwei unabhängigen Zeichenketten je Funktion. Leider ist eine Änderung des Programms zu diesem Zeitpunkt nicht möglich, da dem Autor kein Wordstar-Listing im ZILOG-Format vorliegt.

Benutzerdefinierte Funktionen

WORDSTAR bietet die Möglichkeit der Benutzung von vier anwenderbestimmten Funktionen. Diese werden mit ^PQ, ^PW, ^PE und ^PR eingeleitet. Die Zeichenketten sind in

Wordstar anpassen

die entsprechenden Benutzertabellen (user(USR)) einzutragen.

USR 1 (06C9H) 05C9H; USR 2 (06CEH) 05CEH; USR 3 (06D3H) 05D3H; USR 4 (06D8H) 05D8H.

(j) Rücksetzen von Exponential- oder Indexschrift

Mit ROLUP/ROLDOW ist kein „Aus-schalten“ sondern nur „Ein- und Um-schalten“ von Exponential- oder Index-schrift in sinnvoller Anwendung möglich. Daher ist das „Einbauen“ einer zusätzlichen Zeichenkette im Programm und im Text eine zusätzli-che Funktionsauslösung notwen-dig. Dafür geeignet ist der Befehl APQ.

REROL (06C9H) 05C9H: 02, 1B, 54

(k) Spaltenweiser Wagenrücklauf

In naturwissenschaftlichen Be-schreibungen tritt oft die Notwen-digkeit der Benutzung von Expo-nential- und Indexdruck in einer Volzeile gleichzeitig auf, d.h. Expo-nentialschreibweise in einer oberen Halbzelle und Indexdruck in einer unteren Halbzelle auf. Dazu ist bei-spielsweise ein Exponentialdruck, gefolgt von einem spaltenweisen Wagenrücklauf und ein anschlie-ßender Indexdruck nötig. Mit APW und USR 2 ist das realisierbar. Im Text ist für jeden Spaltenrücklauf ein APW einzutragen, in USR 2 steht die auslösende Zeichenkette. Sie erhält das Label CBS (column-backspa-ce).

CBS (06CEH) 05CEH: 01, 08

Farbbandwechsel

Wordstar bietet mit APY die Mög-lichkeit der Farbenumschaltung für Drucker mit einem zweiten Farb-band. Die Möglichkeit wird wohl hauptsächlich bei Typenrad-druckern für den professionellen Einsatz benutzt, Matrixdrucker für den Hobbybereich verfügen aus Kostengründen selten darüber. In der Tabelle stehen die imaginären Label RIBON (ribbon on) und RIBOFF

(ribbon off) ab (06DDH) 05DDH und (06E2H) 05E2H. Die Benutzung für ei-nen anderen Zweck ist angezeigt.

(l) US-Zeichensatz

Anstelle der Farbbandumschal-tung kann eine Zeichensatzum-schaltung realisiert werden. Die Zei-chenkette erhält die Marke PACSET (print-alternate-character-set). In dem Beispiel wird mit einem ersten APY der US-amerikanische Zeichen-satz eingeschaltet:

PACSET (06DDH) 05DDH: 03, 1B, 52, 00

(m) Ursprünglicher Zeichensatz

Beim Auftreten eines zweiten APY wird die Zeichenkette von PICSET (print-initial-character-set) an den Drucker ausgegeben. In dem Bei-spiel wird der ursprünglich einge-stellte deutsche Zeichensatz wie-der aktiviert:

PICSET (06E2H) 05E2H: 03, 1B, 52, 02

Drucker-Einstellung

Es gibt zwei grundsätzliche Mög-lichkeiten zur Betriebsarteneinstel-lung des Druckers: Hardware-Initia-lisierung und Software-Initialisie-rung. Die erstere geschieht bei mo-dernen Druckern meist mittels klei-nen Schiebeschaltern (DIP). Damit werden in der Regel sogenannte Hauptbetriebsarten ausgewählt (Modus I, Modus II) aber auch der gewünschte nationale Zeichensatz und das CR/LF-Verhalten. Bei der Bedienung eines Druckers von ver-schiedenen Rechnern ist unter Um-ständen jedesmal vor Druckbeginn eine Neueinstellung per Software notwendig. Nach Druckende eine Aufhebung der Einstellung wün-schenswert.

(o) Einstellen vor Druckbeginn

Vor Druckbeginn ist möglicherwei-se ein Wagenrücklauf und die Wahl eines nationalen Zeichensatzes notwendig. Wordstar bietet dafür die Ausgabe der Zeichenkette PSI-NIT (print string for initial) an. Die Bei-

spielkette zeigt die Auswahl des deutschen Zeichensatzes. Fährt der Wagen des Druckers vor Druckbe-ginn nicht selbsttätig an den linken Druckrand, ist das Zähldatum um eins zu erhöhen und ein CR anzufü-gen. Außerdem können die Daten für die Zeilenteilung von 18/216 hin-tenangestellt werden.

PSINIT (06E7H) 05E7H: 07, 1B, 52, 02, 0D, 1B, 33, 12 Halbzeilenteilung
PSINIT (06E7H) 05E7H: 03, 1B, 52, 02, 00, 00, 00, 00 ohne Teilung, ohne CR

(p) Einstellung nach Druckende

Nach Druckende besteht die Mög-lichkeit der Delinitialisierung des Druckers. Das heißt nichts anderes als: Aufheben der gegenwärtigen Einstellung. Die Zeichenkette trägt die Marke PSFINI (print string for final initial). Ein Beispiel ist eine Zeichen-kette zum Einstellen des 1/6 Zollzei-lenvorschubes nach einer Teilung.

PSFINI (06F8H) 05F8H: 03, 1B, 33, 24

(q) Durchstrichzeichen

Satzteile zwischen zwei APX werden durch ein unter dem Label SOCHR (strike-out-character) gespeicher-tes Zeichen durchgestrichen.

SOCHR (070BH) 060BH: 2DH

(r) Unterstrichzeichen

Satzteile zwischen zwei APS werden durch ein unter dem Label ULCHR (underline-character) gespeicher-ten Zeichen unterstrichen.

ULCHR (070CH) 060CH: 5FH

Demonstrationstext ohne besonde-re Druckerwirkung

H^{^V2}V^{^Q}SO^{^T4}T^{^Q} ist die Verh(ltnisformel der Schwefels{ure).

Die folgenden Worte stehen in ^Y[rechteckigen Klammern]^Y.

Der Zuname des Autors steht in rechteckigen Klammern und beinhaltet einen Umlaut: ^Y[^Ylefgr]n^Y]^Y.

Wordstar anpassen

Der Vektorraum $V^{T^2 \times T^2 \times Q^W \times V^o \times V^Q}$ beschreibt den geometrischen Punktraum $E^{T^2 \times T^2 \times Q}$ (Ebene). Eine Gleichung $f(x)$ die Kapazitätsdifferenz bei kapazitiver Ankopplung von Antennen kann so geschrieben werden: $D^{T^2 \times T^2 \times Q} = C^{VA \times V^Q} / 1 + (R^{VA \times V^Q} * o^{V \times \text{mega} \times V^Q} * C^{VA \times V^Q})^{T^2 \times T^2 \times Q}$.

1+10 PSCRLF (0696H) 0596H: 04, 0D, 0A, 0D, 0A bei 18/216 Teilung
 1+06 PSCR-- (06A1H) 05A1H: 02, 0D, 00
 1+06 PSHALF (06A8H) 05A8H: 02, 0D, 0A, 00, 00 bei Halbzeilenvorschub
 1+10 PSCRLF (0696H) 0596H: 04, 0D, 0A, 0D, 0A bei Halbzeilenvorschub

2D
 0+01 ULCHR- (070CH) 060CH: 5F

Beim Erstellen des vorstehenden Artikels wurden die Bücher WORDSTAR-TUNING und WORSTAR-Installationshandbuch unterstützend benutzt. Der Autor dankt Herrn Manfred Meyer und Herrn Werner Kelling, beide aus Neumünster, für die Unterstützung durch Überprüfen der Arbeitsergebnisse auf ihren Druckern.

Funktionstext mit besonderer Druckerwirkung
 H_2SO_4 ist die Verhältnisformel der Schwefelsäure.
 Die folgenden Worte stehen in [rechteckigen Klammern].
 Der Zunahme des Autors steht in rechteckigen Klammern und beinhaltet einen Umlaut: [Lefgrün].
 Der Vektorraum V^E beschreibt den geometrischen Punktraum E^2 (Ebene).
 Eine Gleichung für die Kapazitätsdifferenz bei kapazitiver Ankopplung von Antennen kann so geschrieben werden:
 $D^{T^2 \times T^2 \times Q} = C^{VA} / 1 + (R^{VA} * o^{V \times \text{mega} \times V^Q} * C^{VA})^{T^2 \times T^2 \times Q}$.

Zusammenfassung der Tabellendaten

Allen Zeichenketten, die mehr als ein Datum enthalten können, ist ein Längendatum, das die aktuelle Stringlänge angibt, vorangestellt. Bei Änderung der Kettenlänge ist der Wert des Datums entsprechend zu verringern oder zu vermehren. Die maximalen möglichen Längen der Ketten ist unterschiedlich. Sie sind in der nachfolgenden Tabelle in der Form „0+n“ oder „1+n“ in dezimaler Schreibweise aufgeführt. Die Länge von Einzeldaten ist als „0+1“ angegeben. Einige Labels oder Adressen sind doppelt aufgeführt.

0+01 POSMTH (0690H) 0590H, FF carriage-return-method CRM
 0+01 POSMTH (0690H) 0590H, 00 back-space-method BSM
 0+01 BLDSTR (0691H) 0591H: 02 0+01 DBLSTR (0692H) 0592H: 02 1+10 PSCRLF (0696H) 0596H: 02, 0D, 0A, 00, 00 bei 36/216 Teilung

1+05 PBACKS (06AFH) 05AFH: 01, 08 1+04 PALT-- (06B5H) 05B5H: 02, 1B, 45 1+04 PSTD-- (06BAH) 05BAH: 02, 1B, 46
 1+04 ROLUP- (06BFH) 05BFH: 03, 1B, 53, 00
 1+04 ROLDOW (06C4H) 05C4H: 03, 1B, 53, 01
 1+04 USR 1- (06C9H) 05C9H; 1+04 USR 2- (06CEH) 05CEH; 1+04 USR 3- (06D3H) 05D3H; 1+04 USR 4- (06D8H) 05D8H;
 1+04 REROL- (06C9H) 05C9H: 02, 1B, 54
 1+04 CBS--- (06CEH) 05CEH: 01, 08
 1+04 PACSET (06DDH) 05DDH: 03, 1B, 52, 00
 1+04 PICSET (06E2H) 05E2H: 03, 1B, 52, 02
 1+16 PSINIT (06E7H) 05E7H: 07, 1B, 52, 02, 0D, 1B, 33, 12 mit Teilung und CR
 1+16 PSINIT (06E7H) 05E7H: 03, 1B, 52, 02, 00, 00, 00, 00 ohne Teilung und CR
 1+16 PSFINI (06F8H) 05F8H: 03, 1B, 33, 24
 0+01 SOCHR- (070BH) 060BH:

Die deutsche Schreibmaschinentastatur für Wordstar (Betriebssystem CP/M 80 auf dem MZ-800)

Von Edgar Lefgrün

Angeregt von dem Listing WSTASTEN des Autor im Heft 2/89, äußerten Leser den Wunsch nach einer möglichst freiprogrammierbaren Tastatur und einer der deutschen Schreibmaschine entsprechenden Belegung. Die Ursache des Wunsches nach einer Maschinentastatur ist in der Fähigkeit des Blindschreibens zu sehen. Die Tastaturbelegung ist in erster Linie vom BIOS des CP/M abhängig. Demnach muß eine Umprogrammierung zweckmäßigerweise im Betriebssystem erfolgen. Eine vollkommene Freiprogrammierung ist zwar mit entsprechendem Aufwand an Speicherplatz möglich; ist aber nicht notwendig, da nur ein Teil der Belegung geändert werden braucht.

Wordstar anpassen

Die Grundlagen

Stark vereinfacht kann die Kode-Erzeugung so beschrieben werden: Auf eine der zehn Spaltenleitungen des Tastatursystems wird mittels Maschinenbefehl über den Tastatur-Controller ein Spannungs-Tiefpegel-Signal gegeben und die neun Zeilenleitungen als Ausgänge nacheinander abgetastet. Eine gedrückte Taste erzeugt auch dort ein Low-Signal. Durch gezieltes Pegeln bestimmter Spalten und Abtasten bestimmter Zeilen wird z.B. festgestellt, ob eine „normale Taste“ gleichzeitig mit einer Sondertaste z.B. SHIFT+A oder SHIFT+BREAK gleichzeitig oder nur eine Standardtaste oder Sondertaste gedrückt ist. Dann wird daraus der Tastaturcode geformt und in den Anzeigekode der ROM-Tabelle von 0BEAH-0C69 umgerechnet. Dieser Anzeigekode wird auch als Bildschirmcode bezeichnet. Der Wert eines Bildschirmcodes ist durch die Position seines Bitmusters im CG-ROM und CG-RAM bestimmt.

Der Bildschirmcode wird zum Zweck der verträglichen Datenbearbeitung und Ausgabe an äußere Geräte und Systeme in den ASCII-Kode umgesetzt. Von außen eingehende Daten im ASCII-Kode müssen zur Darstellung auf dem Bildschirm in dessen Kode umgesetzt werden. Zur Umsetzung sind im BIOS verschiedene Tabellen vorhanden. Die ab Adresse F36CH und F37EH sind zum Umsetzen der Tastenkodes der Spalten eins und acht im Normal- oder Shiftmodus. Die Tabelle ab F3D6H ist zum Umsetzen vom Bildschirm- in den ASCII-Kode und umgekehrt.

Das Prinzip dieser Umsetzung kann so beschrieben werden: Die Position in der Tabelle repräsentiert beginnend ab F3D6H mit 00H den ASCII-Kode, das an dieser Stelle stehende Datum den Anzeigekode.

Ein Beispiel: Die Adresse F3F7H ist in der Tabelle die laufende Nummer 21H und enthält den Wert 61H. 21H als ASCII-Kode für das Ausrufungszeichen ist umsetzbar in 61H als Bildschirmcode für das Ausrufungszeichen oder umgekehrt.

Die Tastenkodes der Spalten eins und acht werden mit Ausnahme der Blank-, Graph-, Tab- und Alpha-Tasten „direkt“ in die in den erwähnten Tabellen stehenden ASCII-Kodes umgesetzt. Für die Spalten zwei bis sieben muß eine zusätzliche Umsetzroutine und eine zusätzliche Tabelle eingefügt werden. In dieser Tabelle stehen die umzusetzenden Bildschirm- und ASCII-Kodes hintereinander.

Das Umprogrammieren

Da eine Umprogrammierung der ROM-Dekodierung nicht ohne weiteres möglich ist, muß zur variabler RAM-Dekodierung die nachfolgende Routine ab DE60H und die anschließende Tabelle in das BIOS eingefügt werden.

Einzufügende Routine zur variablen Umsetzung vom Anzeigekode in den ASCII-Kode

```
DE60:ACHANGE: ;Beginn der einzufügenden Umsetzroutine
DE60:CD 06 F5 CALL F506 ;Unterprogramm-Manipulator aufrufen
DE63:06 14 LD B,14H ;x=Tabellenelementanzahl/2
DE65:21 73 DE LD HL,DE73 ;Zeiger auf das Element vor den Tabellenanfang
DE65:LOOP: ;Beginn der Suchschleife
DE68:23 INC HL ;Zeiger auf Vergleichselement korrigieren
DE69:BE CP (HL) ;Zeigerziel mit dem Akkuinhalt vergleichen
DE6A:23 INC HL ;Zeiger auf Ersatzelement korrigieren
DE6B:28 05 JR Z,EXECHG ;Vergleichselement ist gefunden
DE6D:10 F9 DJNZ LOOP ;nicht gefunden, weiter suchen
```

```
DE6F:C3 9F F3 JP F39F ;nicht vorhanden, zur normalen Umsetzroutine
DE72:EXECHG: ;Umsetzen vom Anzeigekode in den ASCII-Kode
DE72:7E LD A,(HL) ;Ersatzelement aus der Tabelle kopieren
DE73:C9 RET ;Operation erfolgreich beendet
```

Nachfolgend die Umsetztabelle. Das jeweils vorstehende Element ist ein Anzeigekode, der in den nachstehenden ASCII-Kode umgesetzt wird. In der darauf folgende Erläuterung steht „nd“ für „nicht darstellbar“. Die dazugehörigen Symbole sind in dem Textsystem des Autors derzeit nicht installiert. Die Bedeutung kann den Tabellen entnommen werden. Zwei Tasten sind für Sonderzeichen freigehalten. Es wurde der ASCII-Kode 5F als Platzhalter eingesetzt. Bei Ersatz durch andere Daten sollten an deren ASCII-Position in der Anzeigekodetabelle die dazugehörigen Displaykodes eingetragen werden. Folgendes sei angemerkt: ASCII-Kodes einzusetzen, die nicht auf dem Drucker und dem Bildschirm als annähernd gleiche Bitmuster darstellbar sind, erscheint allgemein nicht zweckmäßig. Die Gründe dafür sind offensichtlich und bedürfen hier keiner weiteren Erläuterung.

```
DE74:TABLE: ;Tabellenanfang
DE74:55, 5D ; @, UE
DE76:A4, 7D ; ` , ue
DE78:52, 2B ; nd, +
DE7A:BC, 2A ; nd, *
DE7C:51, 3B ; ^, ;
DE7E:57, 3A ; nd, :
DE80:2A, 7E ; -, sz
DE82:2B, 3F ; =, ?
DE84:63, 40 ; #, @
DE86:54, 23 ; nd, #
DE88:40, 27 ; nd, '
DE8A:59, 5F ; nd, _
DE8C:80, 5F ; nd, _
DE8E:60, 3D ; Pi, =
DE90:19, 5A ; Y, Z
DE92:1A, 59 ; Z, Y
DE94:99, 7A ; y, z
DE96:9A, 79 ; z, y
```


Wordstar anpassen

```
DE98:67, 2F ; ', /
DE9A:A5, 50 ; nd, ^
DE9C:50, 5E ; ^, '
Darstellung der einzufügen-
```

den Umsetzroutine als Hex-Dump.

```
DE60: CD 06 F5 06 13 21 73
DE 23 BE 23 28 05 10 F9 C3
DE70: 9F F3 7E C9 -- -- --
```

Darstellung der einzufügen-

den Umsetztabelle als Hex-Dump. DE70: -- -- -- -- 55
5D A4 7D 52 2B BC 2A 51 3B
57 3A

```
DE80: 2A 7E 2B 3F 63 40 54
23 40 27 59 5F 80 5F 60 3D
```

```
DE90: 19 5A 1A 59 99 7A 9A
79 67 2F A5 50 50 5E 00 00
```

Die Sondertasten mit Anhang

Wie bereits erwähnt, stehen die ASCII-Kodes für die Spalten eins und acht in RAM-Tabellen ab F36C und F37E. Die 90 in Spalte 1 steht an Stelle des Blank-Tasten-Kodes. Eine Änderung ist nicht zweckmäßig, da die Taste im SETUP neu belegt werden kann, und der Tastenkod bereits an anderer Stelle „abgefangen“ wird. 09 steht an Stelle des Codes der ALPHA-Taste. Auch dieser Tastenkod wird bereits an anderer Stelle ausgewertet.

Spalte 1 Normalmodus:

Taste: BLANK F36C:90; BLANK/
Taste: GRAPH F36D:5F; GRAPH/
Taste: POUND F36E:3E; größer-als-Operator/
Taste: ALPHA F36F:19; ALPHA/
Taste: TAB F370:09; TAB/
Taste: Semikolon F371:5C; O-Umlaut/
Taste: Doppelpunkt F372:5B; A-Umlaut/
Taste: CR F373:0D; Wagenrücklauf

Spalte 8 Normal-Modus:

Taste: INST F374:18; INST/
Taste: DEL F375:10; DEL/
Taste: CURUP F376:12; CURUP/
Taste: CURDWN F377:11/
Taste: CURRGT F378:13; CURRGT/
Taste: CURLFT F379:14; CURLFT/
Taste: Fragezeichen F37A:2F; Teiler-

strich/
Taste: Teilerstrich F37B:2D; Bindestrich

Spalte 1 Shift-Modus:

Taste: BLANK F37E:90; BLANK/
Taste: GRAPH F37F:5F; GRAPH/
Taste: Zeiger nach unten F380:3C; kleiner-als-Operator/
Taste: ALPHA F381:5F; ALPHA/
Taste: TAB F382:09; TAB/
Taste: Plusoperator F383:7C; o-Umlaut/
Taste: Multiplikationsoperator F384:7B; a-Umlaut/
Taste: CR F385:0D; Wagenrücklauf

Spalte 8 Shift-Modus:

Taste: CLR F386:16; CLR/
Taste: HOME F387:15; HOME/
Taste: CURUP F388:12; CURUP/
Taste: CURDWN F389:11/
Taste: CURRGT F38A:13; CURRGT/
Taste: CURLFT F38B:14; CURLFT/
Taste: Zeiger nach rechts F38C:60; gespiegeltes Auslassungszeichen/
Taste: Zeiger nach links F38D:5F; Unterstrich

Darstellung der geänderten Umsetztabelle für die Tastaturspalten 1 und 8 als Hex-Dump

```
F36C: 90 5F 3E 19 09 5C 5B 0D ;
Spalte 1 Normal-Modus
F374: 18 10 12 11 13 14 2F 2D ;
Spalte 8 Normal-Modus
F37E: 90 5F 3C 5F 09 7C 7B 0D ;
Spalte 1 Shift-Modus
F386: 16 15 12 11 13 14 60 5F ; Spalte 8 Shift-Modus
```

Integration

Die Umsetzroutine muß in die normalen Routinen zur Tastaturkodierung eingebunden, der Wertebereich zur Umkehrung in Großbuchstaben erweitert und die Anzeigekodes der Umlaute und des SZ in die Tabelle an die Positionen der „verdrängten Zeichen“ eingesetzt werden. Soll auch die GRAPH-Taste eine andere Verwendung finden, ist die Umsetzung in den ESC-Kode zu unterbinden. Adressen und Werte sind nachfolgend aufgeführt.

Integrieren der Umsetzroutine in das BIOS: F364: CD 60 DE.
Erweitern des Wertebereiches zur Umsetzung bei CAPS-Lock: F307: FE 5E.

Einsetzen der großen Umlaute in die Bildschirmkodetabelle: F431: AE AF AD.

Einsetzen der kleinen Umlaute und des SZ in die Tabelle: F451: 9B AC AB AA.

Unterdrücken der Umsetzung des Codes der GRAPH-Taste in den ESC-Kode: F1FA: C3 00 F2.

Danach kann in die Tabellenpositionen F36D und F37F der Kode für den Unterstrich 5F als Platzhalter für eine eventuelle spätere sinnvolle Belegung eingesetzt werden.

ASCII-Kodes größer 7FH

Ursprünglich war diese Artikel wesentlich umfangreicher. Eine Umstellung von Umlauten auf ASCII-Kodes für kompatible PC's und Drucker ähnlich der Darstellung in Heft 2/89 („Sharp-Kode auf der Konsole und IBM-Kode für den Drucker“ von Manfred Meyer und Edgar Lefgrün) war bereits entwickelt, erprobt und beschrieben. Auf der Konsole und Drucker funktionierte alles wunderbar. Auch Sonderzeichen wie Pi und Degree waren darstellbar. Dann kam die Erprobung mit Wordstar und das Stauen. WS verwandelte ASCII-Kodes, deren Wert größer als 7FH waren, in Steuerzeichen. Der erweiterte Macintosh-Kode 8AH für a-Umlaut wurde in 0AH, den Wordstar-Kode für den Zeilenvorschub, umgewandelt. Der Wert 80H wurde subtrahiert.

Es ist fraglich, ob eine weitere Umprogrammierung des Textprogrammes möglich und sinnvoll ist. Es wird aber weiter daran gearbeitet.

Zum vormaligen Artikelkonzept gehörte auch die nachfolgende Druckcodeumsetzung. Die Tabelle wurde dem neuen Konzept angepaßt. Sinnvoll ist die Anwendung im

Wordstar anpassen

Zusammenhang mit dem internationalen Zeichensatz.

Die Druckzeichenumsetzung

Bei Kodeunterschieden zwischen Rechner und Drucker ist eine Umsetzung in den Druckercode notwendig. Das BIOS enthält eine Routine von F3BB-F3D5 und eine Tabelle von F4D6-F4F5 zum Umsetzen vom ASCII- in den Sharp-Druckercode. Beides ist für die Art der in diesem Fall benötigten Umsetzung vom Macintosh- in den erweiterten internationalen Kode nicht geeignet und wird deshalb durch nachfolgende Routine und Tabelle ersetzt. Zur Umsetzung muß das Sharp-/Centronics-Flag in E095 auf „1“ gesetzt sein. Das geschieht im Normalfall mit dem SETUP-Programm.

Maschinenroutine für eine Druckzeichenumsetzung

```
F3BB:PCHANGE: ;Beginn des ge-
    änderten Unterprogramms
F3BB:79 LD A,C ;auszugebendes
    Zeichen in den Akku laden
F3BC:21 D6 F4 LD HL,F4D5H ;Zeiger
    auf den Tabellenanfang
F3BF:46 LD B,(HL) ;Tabellennängen-
    datum laden
F3C0:23 INC HL ;Zeiger auf erstes
    zu prüfendes Tabellenelement
F3C0:LOOP: ;Beginn der Prüf-
    schleife
F3C1:BE CP (HL) ;Tabellenelement
    mit Akkuinhalt vergleichen
F3C2:28 05 JR Z,EXECHG ;Element
    ist vorhanden
F4C4:23 INC HL ;Zeiger korrigieren
F4C5:23 INC HL ;Zeiger auf näch-
    stes Tabellenelement
F4C6:10 F9 DJNZ LOOP ;nächstes
    Element prüfen
F3C8:C9 RET ;Zeichen ist nicht in
    der Tabelle enthalten
F3C9:EXECHG: ;Beginn der Um-
    setzroutine
F3C9:23 INC HL ;Zeiger auf Ersatze-
    lement korrigieren
F3CA:4E LD C,(HL) ;Ersatzelement
    aus der Tabelle kopieren
F3CB:C9 RET ;Operation erfol-
    greich beendet
```

ASCII-Kode-Tabelle für die Druckzeichenumsetzung

Adresse:Vergleichselement, Ersatzelement; Kommentar
 F4D6:07 ;x=Tabellenelementanzahl/2
 F4D7:5B, 8E ;AE
 F4D9:5C, 99 ;OE
 F4DB:5D, 9A ;UE
 F4DD:7B, 84 ;ae
 F4DF:7C, 94 ;oe
 F4E1:7D, 9A ;ue
 F4E3:7E, E1 ;sz

Transientes oder resistentes Patchen

Es gibt zwei grundsätzliche Arten, die die Veränderung von Programmen in hexadezimaler Form betreffen; Die andauernde und die vorübergehende Art. Bei der andauernden wird die Änderung auf dem Programmquelldatenträger, in diesem Fall die Systemdiskette, vorgenommen. Bei der vorübergehenden wird mit einem Monitor von

Hand oder zweckmäßigerweise mit einem eine Änderung im RAM erzeugendes, für diesen Zweck besonders gefertigtes Programm gearbeitet.

Hilfe zum Einarbeiten der Änderungen auf der Diskette

Das feste Einarbeiten der Patches geschieht auf der Diskette mit DISKEDIT. Es ist zu beachten, DISKEDIT kann den wirklichen Adreßpegel nicht anzeigen. Das BIOS beginnt auf der Diskette mit der Adresse DE00 auf Spur 3, Sektor 0. Ein logische Sektor, auch als RECORD (Eintrag) bezeichnet, beinhaltet 128 Bytes. Um eine Erklärung der hexadezimalen Rechenregeln zu vermeiden, sind nachfolgend RAM-Adresse, Spur, Sektor und die Adresse innerhalb des Eintrages und zusätzlich ein Kommentar angegeben.

RAM-Adresse: Spur/Sektor: Eintragsadresse; Kommentar

	①	②	③	④	⑤	⑥	⑦	⑧	⑨	⑩
⑪ D ₇	Blank key	Y	Q	I	A	1	~	INST	BREAK ESC	F ₁
⑫ D ₈	GRAPH	Z	R	J	B	2	~	DEL	CTRL	F ₂
⑬ D ₅	⌘	@	S	K	C	3	-	⌘		F ₃
⑭ D ₄	ALPHA		T	L	D	4	SP	⌘		F ₄
⑮ D ₃	TAB		U	M	E	5	Q	⌘		F ₅
⑯ D ₂	;		V	N	F	6	9	⌘		
⑰ D ₁	:		W	O	G	7	.	7		
⑱ D ₀	CR		X	P	H	8	.	/	SHIFT	

Wordstar anpassen

DE60: 03/00: 0060; Umsetzroutinenbeginn
 DE74: 03/00: 0074; Umsetztabellebeginn
 DE80: 03/01: 0000; Tabellenfortsetzung
 E095: 03/04: 0016; Sharp-/Centronics-Flag
 F1FA: 04/07: 007A; Graph-Tastenumsetzung unterdrücken
 F307: 04/10: 0007; Wertebereichsvergleich
 F364: 04/10: 0064; Integrationspunkt
 F36C: 04/10: 006C; Tabellenbeginn Spalte 1 Normalmodus
 F374: 04/10: 0074; Tabellenbeginn Spalte 8 Normalmodus
 F37E: 04/10: 007E; Tabellenbeginn Spalte 1 Shift-Modus
 F380: 04/11: 0000; Fortsetzung Spalte 1 Shift-Modus
 F386: 04/11: 0006; Tabellenbeginn Spalte 8 Shift-Modus
 F3BB: 04/11: 003B; Beginn der Druckumsetzroutine
 F431: 04/12: 0031; Bildschirmcodes, große Umlaute
 F451: 04/12: 0051; Bildschirmcodes, kleine Umlaute, sz
 F4D6: 04/13: 0056; Druckzeichenumsetztabelle

Nach dem Einfügen der Tastaturumsetzung BIOS auf Diskette darf das SETUP+.COM von sds-Computer nicht mehr in der ursprünglichen Form benutzt werden. Das SETUP+ fügt die gleiche Routine mit anderem Zähldatum und eine andere

Tabelle für Umlaute und sz an gleichen Adressen ein. Auch in die Tabellen der Spalten eins und acht erfolgt zumindest ein nachgewiesener Eingriff.

Anregungen zum transienten Patchen

Programmgesteuerte zeitweilige Patches im BIOS-Bereich des RAM's sind auf vielfältige Weise erzeugbar. Einige Möglichkeiten sollen erwähnt werden.

- Das Patchen des BIOS vor dem Aufruf von WORDSTAR per Programm.
- Anschließend Aufruf von WS durch das Programm.
- Aufruf des Patchers durch WORDSTAR.
- Rücksetzen des BIOS durch Programmaufruf eines Patchers beim Beenden von WS. Programmieren kann solch ein Patcher, z.B. in C-Basic, Turbo-Pascal oder Assembler. Es könnte im Overlay-Bereich von WS ablaufen. In ML programmiert könnte es in WS integriert oder angefügt werden. Eine noch andere Lösung ist das Arbeiten mit SUBMIT und DDT oder ZSID.

Der Verfasser hofft, das Thema dem Leserinteresse entsprechend sorgfältig und ausführlich behandelt zu haben. Er dankt Herrn Hans-Peter

Laackmann aus Tettnang für die Anregung zu dieser Arbeit. Als Arbeitsunterlage wurde benutzt: BIOS-Listing von Rainer Schäfer, sds-Computer.

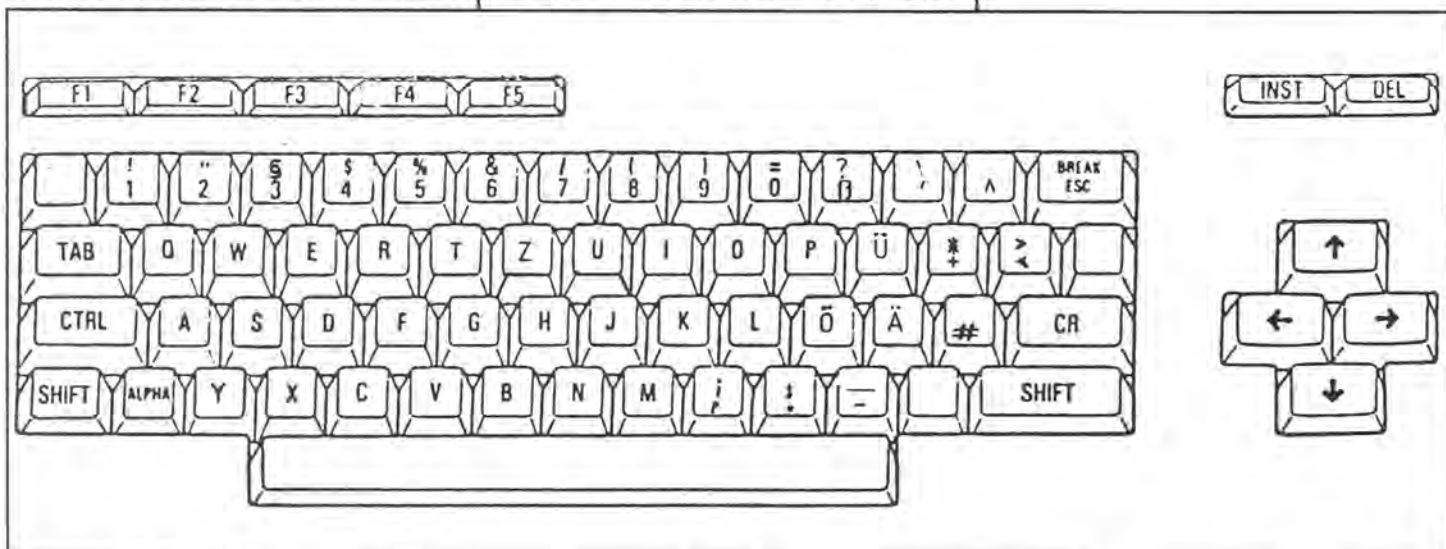
Wagenrücklauf und Zeilenvorschub

Von Edgar Lefgrün

Zur korrekten Nutzung der Unterstrich- und Durchstrichfunktionen beim Textverarbeitungsprogramm WORDSTAR, ist das Ausschalten des automatischen Zeilenvorschubes (AUTO-LF) am Drucker notwendig.

Das führt zu Komplikationen bei der Handhabung der Basic-Interpreter. Diese sind zur Ausgabe von CR(\$0D) bei Zeilenende voreingestellt. Eine bedarfsweise Hardware-Umstellung ist umständlich und bewirkt einen mechanischen Verschleiß des DIP-Schalters. Eine fernwirkende Umstellung per Software ist bei den dem Autor bekannten Druckern nicht möglich.

Das Problem ist zwar bei den 800er Interpretern bei eingeschalteten und ON-LINE befindlichen Drucker mit dem Befehl INIT"LPT:M0,S2,\$A" lösbar, bei den 700er Interpretern oder ausgeschalteten Drucker nicht möglich oder führt zum Programmabbruch.



Wordstar anpassen

LSD	MSD															
	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
0	0,000	SP	P	O	□	↑	π	□	□	p	□	□	□	□	SP	
1	0001	A	Q	1	□	↑	□	□	a	q	□	□	□	□	□	
2	0010	B	R	2	□	□	□	□	b	r	□	□	□	□	□	
3	0011	C	S	3	□	□	□	□	c	s	□	□	□	□	□	
4	0100	D	T	4	□	□	□	□	d	t	□	□	□	□	□	
5	0101	E	U	5	□	□	□	□	e	u	□	□	□	□	□	
6	0110	F	V	6	□	□	□	□	f	v	□	□	□	□	□	
7	0111	G	W	7	□	□	□	□	g	w	□	□	□	□	□	
8	1000	H	X	8	□	□	□	□	h	x	□	□	□	□	□	
9	1001	I	Y	9	□	□	□	□	i	y	□	□	□	□	□	
A	1010	J	Z	□	□	□	□	□	j	z	□	□	□	□	□	
B	1011	K	€	□	□	□	□	□	k	ä	ü	□	□	□	□	
C	1100	L	□	□	□	□	□	□	l	□	□	□	□	□	□	
D	1101	M	□	□	□	□	□	□	m	□	□	□	□	□	□	
E	1110	N	□	□	□	□	□	□	n	□	□	□	□	□	□	
F	1111	O	□	□	□	□	□	□	o	□	□	□	□	□	□	

Eine einfache Methode erscheint ein Eingriff per POKE in die Interpreter zu sein. Bezeichnungen, Adressen und Daten sind nachfolgend aufgelistet.

MZ-2Z016 POKE\$1093,\$A/
 MZ-5Z009 POKE\$1093,\$A/
 MZ-2Z046 POKE\$1093,\$A/
 MZ-1Z013 POKE\$1677,\$A/
 MZ-5Z008 POKE\$0F1E,\$A

Aufzuführen sind noch die Patches für drei vielbenutzte Programme.

DISKEDIT 83 von Bruno Volkmer:
 21EF:3E,0D ändern 3E,0A

UNIMON von Eckart Schatter:
 CCB7:0D ändern 0A

FD-Assembler von Kersten + Partner:
 C037:3E,0D ändern 3E,0A

Sollten Sie liebe Leserin, werter Leser, negative Erfahrungen mit den Änderungen machen oder die Liste durch Daten für andere Interpreter oder Programme ergänzen können, schreiben Sie uns bitte!

LSD	MSD															
	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
0	0000	SP	O	@	P	□	□	□	□	q	n	□	□	□	□	
1	0001	↓	!	I	A	Q	H	□	□	a	□	□	□	□	□	
2	0010	↑	"	2	B	R	I	□	□	e	z	Ü	□	□	□	
3	0011	→	#	3	C	S	□	□	□	w	m	□	□	□	□	
4	0100	←	\$	4	D	T	K	□	□	s	□	□	□	□	□	
5	0101	□	%	5	E	U	□	□	□	u	□	□	□	□	□	
6	0110	□	&	6	F	V	□	□	□	t	i	□	□	□	□	
7	0111	□	'	7	G	W	□	□	□	g	□	□	□	□	□	
8	1000	□	(8	H	X	□	□	□	h	ö	□	□	□	□	
9	0001	□)	9	I	Y	□	□	□	i	□	□	□	□	□	
A	1010	□	*	□	J	Z	□	□	□	j	□	□	□	□	□	
B	1011	□	+	□	□	□	□	□	□	□	□	□	□	□	□	
C	1100	□	,	□	□	□	□	□	□	□	□	□	□	□	□	
D	1101	CR	□	□	□	□	□	□	□	□	□	□	□	□	□	
E	1110	□	>	□	□	□	□	□	□	□	□	□	□	□	□	
F	1111	□	□	□	□	□	□	□	□	□	□	□	□	□	□	

Listing von Hans-Peter Auel

```
1 'Programm UNERASE zum Retten gelöschter Programmfiles. H.-P. Auel 15.5.89
2 'für FD-Interpreter MZ-22046
3 '
4 'Wenn der Interpreter ein File von der Diskette löscht, ändert er nur den
5 'Typ des Files auf 00. Ebenso gibt er die vom File belegten Sektoren wieder
6 'frei. Solange noch nichts auf die Diskette geschrieben wurde, kann man also
7 'die Files retten, die Information ist ja noch vorhanden.
8 'Das vorliegende Programm sucht nach gelöschten Programmen und überprüft ob
9 'die Sektoren schon überschrieben wurden. Ist das nicht der Fall, wird das
10 'File zurückgeholt.
11 'Eine Besonderheit der BSD und BRD Dateien verhindert die Rekonstruktion
12 'des Sektorverzeichnisses. Diese Files werden nicht in geschlossenen Blöcken
13 'sondern als einzelne Sektoren in verketteter Liste abgespeichert.
14 'Diese werden nur im Directory eingetragen und nicht im Sektorverzeichnis.
15 'Dadurch sind sie zwar lesbar, aber nicht gegen Überschreiben geschützt.
16 'Ebenso sind sie nicht mehr löschar. Die Datenfiles müssen kopiert werden,
17 'den Eintrag im Directory kann man umbenennen oder mit einem Diskmonitor
18 'entfernen.
19 '
20 DIRFD1:'Nur zum Einlesen desselben ,festlegen des Laufwerkes und
30 INIT"CRT:M3": 'auch das Sektorverzeichnis einlesen
40 S=$27F0:' erster Directory Eintrag
50 VV=0
60 PRINT"Gelöschte Files suchen:";
70 IF S>$2FB0 THEN 670 :' Letzter DIR Eintrag ?
80 IF PEEK(S)<>0 THEN S=S+32:PRINT"*";:GOTO70:'Prog nicht gelöscht
90 L=PEEK(S+$14)+PEEK(S+$15)*256
100 IF L=0 THEN S=S+32:GOTO70
110 PRINT:PRINT"Programm :",
120 FORI=iTO 16
130 A=PEEK(S+I):IF A=13THEN I=16:GOTO150
140 PRINTCHR$(A);
150 NEXTI:PRINT
160 PRINT"Anfang:",:A=PEEK(S+$16)+PEEK(S+$17)*256:PRINTHEX$(A)
170 PRINT"Ende:",:L=PEEK(S+$14)+PEEK(S+$15)*256:PRINTHEX$(A+L-1)
180 PRINT"Länge:",INT(L/1024*100+.5)/100;" kB"
190 PRINT"Start:",:G=PEEK(S+$18)+PEEK(S+$19)*256:PRINTHEX$(G)
200 ST=PEEK(S+$1E)+PEEK(S+$1F)*256
210 PRINT"Von:", "Track";INT(ST/16), "Sektor";FRAC(ST/16)*16
220 SE=ST+L/256
230 PRINT"Bis:", "Track";INT(SE/16), "Sektor";INT(FRAC(SE/16)*16)
240 PRINT
250 PRINT"ist gelöscht. Zurückholen ?";
260 GET A$:IF A$="" THEN 260
270 PRINTA$
280 IF A$="N" THEN S=S+32:GOTO 50
290 IF A$<>"J" 260
300 VV=1
310 V=$2FD6+INT(ST/8)-6:' Anfang des Sektorverzeichniss + erster Sektor -Offset
320 AS=INT(L/256):IF L MOD 256 <> 0 THEN AS=AS+1 :'Anzahl der Sektoren/Programm
330 FOR I=0 TO AS-1
340 IF PEEK (V+INT((I+FRAC(ST/8)*8)/8)) AND INT(2^(FRAC((I+FRAC(ST/8)*8)/8)*8))
< 1 THEN I=5000
350 NEXTI
360 IF I=AS THEN 420 :' Sektoren noch in Ordnung
370 PRINT
380 PRINT"Es wurde schon ein anderes Programm auf dieser Diskette gespeichert, n
achdem"
```

Listing von Hans-Peter Auel

```
390 PRINT"das Programm gelöscht wurde! Es ist daher unmöglich das Programm zurück zu"
400 PRINT"bekommen. Die Sektoren sind dadurch überschrieben worden. Schade !"
410 PRINT"Oder waren es Daten ?"
420 PRINT
430 PRINT"Welchen Filetype hatte das Programm : "
440 PRINT"      OBJ      [1]"
450 PRINT"      BTX      [2]"
460 PRINT"      BSD      [3]"
470 PRINT"      BRB      [4]"
480 PRINT
490 GET A:IF A<1 OR A >4 THEN 490
500 PRINT A
510 IF I=5000 THEN S=S+32:GOTO 50
520 POKE S,A
530 IF A < 3 THEN 590
540 PRINT"Das Datenfile ist jetzt wieder lesbar. Bitte kopieren Sie es sofort auf eine"
550 PRINT"andere Diskette, da das Sektorverzeichniss nicht angepasst wird. Geben Sie dem"
560 PRINT"File auf DIESER Diskette mit RENAME bitte einen anderen Namen, da der Eintrag"
570 PRINT"nicht löscher ist."
580 WAIT 5000:S=S+32:GOTO70
590 FOR I=0 TO AS-1
600 POKE V+INT((I+FRAC(ST/8)*8)/8) , PEEK (V+INT((I+FRAC(ST/8)*8)/8)) OR VAL(STR$(2^(FRAC((I+FRAC(ST/8)*8)/8)))) : ' Sektoren als belegt kennzeichnen
610 NEXT I
620 FR=PEEK($2FD2)+PEEK($2FD3)*256+AS : ' Anzahl belegter Sektoren
630 POKE$2802,FR MOD 256,INT(FR/256) : ' errechnen u. eintragen
640 S=S+32:GOTO 70
650 '
660 'Sektorverzeichniss zurückschreiben, wenn es verändert wurde
670 IF VV THEN ELSE END
680 A$="FD1:" :USR($1A45):USR($35F0):USR($3702) : 'Directory auf Disk schreiben
690 END
700 '
710 '
1000 ' Hier stehen die Änderungen mit Zeilennummer + 1000
1002 'für die FD-Interpreter MZ-52008 und MZ-52009
1030 CLS : 'auch das Sektorverzeichniss einlesen
1040 S=$2020:' erster Directory Eintrag
1070 IF S>$27B0 THEN 670 : ' Letzter DIR Eintrag ?
1310 V=$2806+INT(ST/8)-6:'Anfang des Sektorverzeichniss + erster Sektor -Offset
1620 FR=PEEK($2802)+PEEK($2803)*256+AS : ' Anzahl belegter Sektoren
1680 A$="FD1:" :USR($1A3D):USR($2E2A):USR($2F37) : 'Directory auf Disk schreiben
```


Listing von Karsten Schulte

```

program wertetabelle; { fuer ganzrat. Funktionen mit Proz. "function" }
var c:array[0..10] of real;
    n:integer;
    a,b,delta:real;

(* Vorlage: Fachliteratur *)
(* Dieses Programm hat Karsten Schulte GravensteinerStr.16 2223-Meldorf *)
(* geschrieben. Ich suche HILFE!!!! Wie kann man die Eingabe von Buchstaben*)
(* beim Grad der Funktion, den Koeffizienten und den Intervallgrenzen *)
(* abfangen ??????*)

procedure lies_koeffizienten;
var i:integer;
begin
  clrscr;
  writeln;writeln;
  writeln('      Ermittlung der Wertetabelle einer ganzrationalen Funktion ');
  writeln('-----');
  writeln;
  write('      Geben Sie den Grad (n_max=10) des Polynoms ein ! n = ');readln(n);
  writeln;
  writeln('      Und nun die Koeffizienten !');writeln;
  for i:=n downto 0 do begin
    write('      c',i,' = ');readln(c[i])
  end
end;

procedure lies_daten;
begin
  writeln;writeln;
  writeln('      Geben Sie Intervallgrenzen und Schrittweite an !');writeln;
  write('      a --> ');readln(a);
  write('      b --> ');readln(b);
  write('      Schrittweite --> ');readln(delta)
end;

procedure schreib_Tabelle;
var x:real; k:integer;

function f(x:real):real;
var i:integer;
    y:real;
begin
  y:=0;
  for i:=n downto 1 do y:=(y+c[i])*x;
  y:=y + c[0];
  f:=y
end;

begin { schreib Tabelle }
  clrscr;writeln;
  writeln('      =====');
  writeln('      | WERTETABELLE DER FUNKTION : |');
  writeln('      =====');
  WRITELN;
  write('      f(x) = ');
  for k:=n downto 1 do write(c[k]:5:2,'*x^',k,' + ');writeln(c[0]:5:2);
  writeln;
  writeln('      x':6,'|':5,'f(x)':9);
  writeln('      -----|-----');
  x:=a;
  while x<=b do
    begin
      writeln('      ', x:8:5,'|':3,f(x):9:5);
      x:=x+delta
    end
end; (* schreib Tabelle *)

(* -----*)

begin
  lies_koeffizienten;
  lies_daten;
  schreib_tabelle
end.-----*)

```

Kleinanzeigen

Kleinanzeigen

Um Ihnen zu helfen, wenn Sie etwas suchen, kaufen oder verkaufen wollen, bieten wir Ihnen den Kleinanzeigenmarkt. Für bis zu fünf Zeilen zahlen Sie nur DM 10,-. Händler zahlen nur DM 20,-. Wenn Sie Gewerbetreibender sind, beachten Sie bitte, daß das aus Ihrer Kleinanzeige aus wettbewerbsrechtlichen Gründen deutlich hervorgehen muß. Jede Zeile darf bis zu 27 Anschläge haben. Bitte beachten Sie, daß Kleinanzeigen nur bei Vorkasse berücksichtigt werden können.

5,25" MZ-1F19 inkl. Contr.+P-CP/M+D-Basic 310, --/64kB Ram 60, --/MZ800 Maus+Cad 60, --/10 QD 25, --/div. Softw. z.B. Textwr. 20, --INFO 089/464936

WordStar Patches, meine Anzeige im 700/800 Magazin 5/6 88. Zuschrift aus Italien, bitte noch einmal melden, habe die Adresse verloren. Theo Dame, Postfach 1431, D-5760 Arnsberg

Verkaufe MZ-800, Quick-Disk, Datarecorder, Epson LX80 komplett DM 500, -- oder gegen Gebot. Gert Bezner, 7430 Metzingen, 07123/41284 oder 4612

Verkaufe MZ-800 CMT inkl. Grafikerweiterung; 5,25" Floppy-Laufwerk für den MZ; Sharp Plotter CE-516P 4-farbig; ca. 65 Disketten voll mit Programmen für nur VB DM 600, -- Tel. 08134/562

Verkaufe MZ-821 + Video RAM 150, --; RAM-Disk 100, --; 3,5" Floppy + Controller + Disketten + Bücher + Diskettenbox 450, --Centronics-Kabel 25, -- Dave Effelsberg, Hermann-Troll-Weg 13, 4400 Münster Tel. 0251/615148

Verk. MZ-821 (V.RAM), Joyst. (1X16), RS232 (MZ1BI03), FD-Karte (MZ1E05), Literatur u. Prg. K. H. Hertel, Königsberger Str. 24, 6600 Saarbrücken

Suche für MZ-700 mit K+P-Floppy MZ-700 Basic-Interpreter-Diskette. Stefan Rommeis, Kreuzbergstr. 9, 6409 Dippers

Buch „Die beiden Datenbanksysteme dBase II und III“ von A. Janson, Franzis Verlag, sucht Manfred Meyer, Unterjörn 71, 2350 Neumünster, 04321/32370

Verkaufe MZ-800 mit Quick-Disk, Datarecorder, Video-RAM, 64K Speichererweiterung komplett für VB DM 500,00. MZ-Verlag Harald Schicke

Hilfe!!!! Es gibt keine Basiclistings mehr!!!
Wer verkauft mir sein Listing vom 800'er Disketten-Basic MZ-2Z046, oder leiht es mir?
Oliver Brendel Erlkamerstr.26 8150 Holzkirchen,
☎ 08024 / 7396

Fehlerkorrektur

Beim Konvertieren des Programms TBE-Scroll in Heft 2/89 unterlief mir ein schwerer Fehler. Die Zeilen 150-220, 400-440 und 530-550 sind verstümmelt oder fehlen. Ich bitte Sie um Entschuldigung. Ich hoffe, ich habe die nachfolgende Ergänzung richtig wiedergegeben. Der Text wurde rekonstruiert, da die Daten auf der Diskette bereits überschrieben wurden. Die Begriffe (:, :) werden im Programmtext bedeutungsgleich mit den eckigen Klammern benutzt. Edgar Lefgrün

```
150 DATA $21,$00,$80,$11,$28,$00,$3A,$FA,$FE,$FE,$00,$28,$04,$47,
    $19,$10
160 DATA $FD,$22,$FC,$FE,$21,$00,$00,$3A,$FA,$FE,$47,$3A,$FB,$FE,
    $90,$FE
170 DATA $00,$28,$04,$47,$19,$10,$FD,$22,$FE,$FE,$C9,$DB,$E0,$3E,
    $00,$D3
180 DATA $CE,$3E,$01,$D3,$CD,$D3,$CC,$2A,$FC,$FE,$11,$40,$9F,$01,
    $28,$00
190 DATA
    $ED,$B0,$2A,$FC,$FE,$01,$28,$00,$09,$ED,$5B,$FC,$FE,$ED,$4B,$FE
200 DATA
    $FE,$ED,$B0,$21,$40,$9F,$01,$28,$00,$ED,$B0,$3E,$02,$D3,$CD,$D3
210 DATA
    $CC,$2A,$FC,$FE,$11,$40,$9F,$01,$28,$00,$ED,$B0,$2A,$FC,$FE,$01
220 DATA
    $28,$00,$09,$ED,$5B,$FC,$FE,$ED,$4B,$FE,$FE,$ED,$B0,$21,$40,$9F
400 FOR X=72 TO 122 STEP 10
410 SYMBOL (:INT(RND*3):) 12,X „TBE Software“,3,1
420 NEXT
430 BOX (:1:) 50,90,270,110,2
440 SYMBOL (:0:) 60,93,„SHARP MZ-800“,2,2
```

530 ' und beide Werte müssen zwischen 0 und 199 liegen.
540 ' Anschließend wird die Initialisierungsroutine ab \$FE00 aufgerufen:
550 USR(\$FE00):'INIT

... auf die richtigen Partner kommt es an
Teamwork

**Sharp Personal Computer
&
MZ-Verlag Harald Schicke**



**Bücher und
Qualitätssoftware
für MZ-700 + MZ-800
aus dem MZ-Verlag**

**Hier erhalten Sie
Lebenshilfe:**

- Bio-Partner
- Familienplanung

**Hier wird gespielt
und gelernt:**

- MZ-700-Klavier
- Jumper Jimmy
- Der verflixte Kohl
- Primus
- Chemie Null
- Schachprogrammierung

**Hiermit verdienen
Sie Geld:**

- Busigraph
- Plotgraph
- Textverarbeitung
- Polydat
- Painter
- Stockresearch
Ihr persönliches

**Hiermit steigen
Sie ein:**

- Einstelgerbuch
für den MZ-800
- BASIC 12013
Deutschkommentiertes
BASIC-Listing
- BASIC-Trainer