

Title: „Heute schon geust?!“ Nr. 1

Document name: hsg1.pdf

Format: DIN A5

Scanning, rework: Ullrich von Bassewitz, uz@musoftware.de

PDF creation: 2009-04-14 22:53

Download URL: <ftp://ftp.musoftware.de/pub/uz/hsg/hsg1.pdf>

HEUTE SCHON GEUST ?!

protected widely throughout the world

hochoffizielles Organ der
CBM Selbsthilfe 600/700

INTERNATIONAL



Photoimitatur Coloniae A.D. MCMLXXXVII

NR.1

Inhaltsverzeichnis

Lee B. Leser	1
Software	2
Arbeitskreise	2
Regionaltreffen	3
Ausbau der Bank 15	4
Nachlese zur 1Mbyte - Erweiterung	5
Die Datasette streikt nie wieder	6
Neues aus dem CBM - Normenausschuß	8
Änderung von Devicenummern	10
Kopien wichtiger Unterlagen	12
Tips, Tricks und kleine Programme	13
Filetrans 600/700 zu älteren CBM-Rechnern	19
8000 Punkte-Grafik	20
Reparaturtips	22
Speichererweiterung 256k	23
Obersicht Editorfunktionen	24
Bildschirmmonitortips	25
8250 bekommt 8050-Format	26
Memory Map des CBM 600/700	27
Umbau 600 → 700 und ASCII → DIN	33
DIN/ASCII Universalkernal	34
DIN Tastaturbelegung	35
Das 64k - Banking des CBM 600/700	36
Krimskram	45
Flohmarkt	46
Materialbeschaffung (Sammelbestellungen)	46
Kommerzielle Bezugsquellen	47
Schlußbemerkungen	48
Impressum	49

Lee B. Leser,

wieder einmal haben Sie unsäglich lange warten müssen, bis Sie nun endlich HSG?I Nr. 1 in den Händen haben, und das, nachdem ausgerechnet ich zu einer Beschleunigung des Informationsflusses ermahnt hatte. Hoffentlich fühlen Sie sich ein wenig durch Umfang und Inhalt dieses Heftes entschädigt. Leider sehe ich mich einem Wust von Arbeit gegenüber, der schwer zu bewältigen ist. Eine sinnvolle Arbeitsteilung ist auch nicht so leicht zu organisieren. Immerhin kommt diese langsam zustande, und ich glaube, das schlimmste Überstanden zu haben, zumal da mir jetzt wenigstens die elende Kopiererei abgenommen wird. Es kann also weiter bei mir bestellt werden. Ich hoffe, durch die Kopieraktion, die mich lange Zeit in einem ungeahnten Maße beansprucht hat, die ärgsten Informationslücken geschlossen zu haben.

Auch die Fragebogenaktion, die dazu dienen soll, daß sich Mitbenutzer überregional themenorientiert zusammentun können, hat eine unerwartete Menge Arbeit beschert. Es war ja geplant, einfach die Vorderseiten zu kopieren und zu verbreiten. Dann stellte sich heraus, daß die Rückseiten oftmals die entscheidenden oder auch vertrauliche Informationen enthielten. So mußte alles Wichtige abgetippt werden. Hierfür wird Wolf Dieter Metzke mein Dank ewig nachschleichen.

Das größte Übel aber, das alle meine Zeitpläne über den Haufen geworfen hat, ist das Telefon, das ich nolens volens an manchen Tagen rund um die Uhr ohne Unterlaß zu bedienen hatte. Daher bitte ich inbrünstig, mich und alle Anderen, deren Anschriften in HSG?I verbreitet werden, nur in wirklich dringenden Fällen anzurufen.

Außerdem habe ich bisher nur wenige individuelle Anfragen beantworten können. Da dies das erste mal ist, das ich mit dem Computer wirklich arbeite, können Sie sich vorstellen, daß ich in CBM-spezifischen Fragen der Programmierung der falsche Ansprechpartner bin. Teilweise recht interessante Fragen zur Hardware werde ich versuchen, in den nächsten Heften zu beantworten. Mehr als auf eine Postkarte paßt, kann ich aus Zeitmangel einfach niemandem individuell antworten, es sei denn, daß ein schaltungstechnisches Vorhaben von mehr oder minder allgemeinem Interesse davon abhängt. Übrigens bin ich seit nunmehr fast einem halben Jahr fast ausschließlich mit "Club"-angelegenheiten befaßt.

Dem vielfach geäußerten Wunsch nach einer detaillierten Hardwarebeschreibung werde ich in den nächsten 3 - 4 Heften nachzukommen versuchen. In vielen diesbezüglichen Gesprächen bin ich in der Auffassung bestärkt worden, daß man bei der Beschreibung von Schaltungserweiterungen schon jetzt die Kenntnis der Grundschaltung voraussetzen sollte. Das Verlangen nach einer Einführung in Basic kann hier beim besten Willen nicht erfüllt werden. Dazu wird eine schier unüberschaubare Menge von Büchern feilgeboten, die geeignet sind, die für Anfänger wohl etwas dürftig geratenen Handbücher sinnvoll zu ergänzen. Die besonderen (Un-)Möglichkeiten des Basic 4.0 gehören allerdings hierher.

Dieses Heft geht an alle Interessenten, deren Anschriften mir teilweise auch zugetragen worden sind. Daher noch einiges Grundsätzliches.

Die CBM Selbsthilfe ist eine lose Vereinigung auf der Grundlage des bürgerlichen Rechts. Es wird kein Gewinn angestrebt, sondern lediglich eine gegenseitige Kostenersatzung erwartet. Für diese und einige künftige Zeitschriften wird eine Vorauszahlung von DM 20,- erwartet. Mal sehen, wie lange das reicht. Die erwähnte Fragebogenaktion kostet DM 5,-. Leider habe ich es bis jetzt nicht geschafft, ordentlich über die bislang eingegangenen Zahlungs-

gen Buch zu führen, da ich erst seit ein paar Tagen über ein entsprechendes Programm verfüge. Daher bekommen auf Verdacht einfach alle Einsender eines Fragebogens die Fragebogensammlung. Es sei mit Nachdruck darauf hingewiesen, daß die meisten Einsender ausdrücklich darauf hingewiesen haben, daß sie eine Weitergabe ihrer Daten an kommerzielle Interessenten untersagen. Ich hoffe, daß sich alle daran halten.

Die Kontonummer: Sylvia Rothe Sonderkonto PGA Köln, BLZ 370 100 50
Kontonummer 3783 36-507

Zu bemerken ist noch, daß HSG?! auf ziemlich schwachen Füßen steht. "Abonnenten" gibt es bis jetzt kaum über 100. Vielleicht tragen auch Sie zu einer Vergrößerung des Leserkreises bei.

Ich hoffe, Sie nicht noch einmal lange warten lassen zu müssen.



G. Schumacher

Software

Zunächst sei an dieser Stelle all jenen gedankt, die Programme geschickt haben. Dringend benötigt werden teilweise noch Bedienungsanleitungen und Informationen über die urheberrechtlichen Verhältnisse. Nachdem aus den hier eingegangenen Programmen weitgehend die namensgleichen entfernt waren, wies das Master-Directory immer noch etwa 950 Programme aus. Frank Vandecruys ist derzeit damit beschäftigt, per Programm auch hier noch doppelte Files aufzustöbern, eventuelle Unterschiede festzustellen, die auf dem 600/700 unbedingt lauffähigen auszusortieren und auf ihre Schutzwürdigkeit im Sinne des Urheberrechts zu überprüfen. Jedenfalls hat sich schon herausgestellt, daß eine unüberschaubare Fülle von Programmen verteilt werden können wird.

Was kommerzielle Software anbelangt, bitten wir um Erfahrungsberichte Ihrerseits. Günter Brühl, Steinerstr. 16, 7958 Laupheim, der die Softwareredaktion übernommen hat, hofft auf Ihre Mithilfe, und wird diese Berichte - möglichst mit Bezugsquellen und Preisen - zu einer wahrscheinlich recht umfangreichen Marktübersicht zusammenfassen. Das Angebot reicht vom Basiccompiler über die Basicerweiterung "Macro-Basic" mit wirklich ungewöhnlichen Leistungsmerkmalen, diverse Programmierhilfen bis zu allen wesentlichen Büroanwendungen incl. Fibu.

Arbeitskreise

Da nun endlich mit gleicher Post die gesammelten Fragebögen an alle Einsender eines solchen verteilt werden, ist zu hoffen, daß sich bald projektbezogene Arbeitskreise zusammenfinden werden. Hier schon mal die Anschriften einiger Ansprechpartner für bestimmte Bereiche:

Softwareredaktion HSG?!: Günther Brühl, Steinerstr. 16, 7958 Laupheim
Hardwareredaktion HSG?!: G. Schumacher, Palanterstr. 12-d, 5000 Köln 41
Furkamateure : Dirk Möll, Am Wallenbaum 27, 6600 Saarbrücken
Probleme mit Kernal : A. Stankewitz, Lindenstr. 15 Hinterhaus, 5000 Köln 1

Regionalgruppen

Hier in Köln hat es sich hervorragend bewährt, sich gelegentlich in einem größeren Kreis zu treffen. Wenn sich Leute mit den unterschiedlichsten Erfahrungen zusammenfinden, findet man hier für die Lösung seiner Probleme mit dem Computer doch in den meisten Fällen einen geeigneten Ansprechpartner. Außerdem ist ein Gedanken- und Erfahrungsaustausch über eine zentrale Stelle schlichtweg unmöglich. Die Treffen zu Köln finden in einem separaten Saal der Gaststätte Mohr Baedorf, Neumarkt 43, statt. Dies ist so zentral gelegen, daß auch auswärtige Teilnehmer keine Schwierigkeiten haben, es zu finden. Die nächsten Termine (jeweils Samstag, 15 Uhr): 14. und 28. März, 11. und 25. April.

Ich halte es für günstig, wenn sich möglichst viele Mitbenutzer zu solchen Treffen an wirklich zentralen Orten zusammenfinden. Um an den kölnischen Treffen teilzunehmen, haben sich beispielsweise einige Fahrgemeinschaften aus Aachen, Siegburg und den Niederlanden gebildet. Das ist bestimmt auch anderswo möglich. Einige Organisatoren solcher Treffen habe ich schon geworben. Man möge sich mit Angabe von Terminvorstellungen schriftlich, mit beigelegter Antwortpostkarte an die nachfolgenden Adressen wenden. Angebote und gesuchte Mitfahrgelegenheiten mögen in den Anschreiben vermerkt werden.

für die ganz weite Umgebung von Frankfurt (Mainz, Wiesbaden, Gießen usw.):
Moritz Mädler, Am Aussichtsturm 42, 6050 Offenbach

dito für Hamburg:
Jürgen Kaiser, Tralauer Kehre 4c, 2000 Hamburg 62

für das Ruhrgebiet:
Ralf Viehl, Mendelssohnstr. 12, 4370 Marl

für Bremen:
Rüdiger Lemke, Bruchhauser Str. 82, 2800 Bremen 61

für Berlin:
Detlef Kirch, Alt Gatow 41, 1000 Berlin 22

für Braunschweig:
Axel Schröder, Schöttlerstr. 16, 3300 Braunschweig

für Kassel, Göttingen:
Joachim Körnle, Gräfe Str. 31, 3500 Kassel

Wohlgerne haben die genannten sich lediglich bereit erklärt, ein erstes Treffen zustandezubringen. Solche Ansprechpartner werden noch für einige Gebiete gesucht. Wer dort ein regionales Treffen zu organisieren bereit ist oder an einem solchen teilzunehmen wünscht, möge sich nach dem oben beschriebenen Muster wenden an

Guido Thater, Gottfried - Keller - Str. 5, 5000 Köln 41.

Die dort eingehende Post wird dann gesammelt und an die jeweiligen Organisatoren weitergereicht. Bei der Auswahl von Regionen, in denen Treffen sinnvoll erscheinen, habe ich einfach nach der Übereinstimmung der beiden ersten Ziffern der Postleitzahl geschaut (mindestens 5*). Die Zahl in Klammern gibt die Anzahl Empfänger dieses Heftes in den betreffenden Gebieten an.

München (9), Würzburg (6), Nürnberg (5), Kiel (5), Herford/Osnabrück (11), Hannover/Celle/Hildesheim (32)

Leider habe ich erst später erfahren, daß die Bauteilenummerierung und die Pinbelegung im 600 und im 700 völlig voneinander verschieden sind. Somit kann die Bauanleitung nicht ohne weiteres für den 700 übernommen werden. Andreas Kesenheimer hat sich die Mühe gemacht, eine Vergleichsliste aus den Schaltplänen herauszuschreiben. Dann stellte sich heraus, daß beim 700 der Schaltplan wesentliche Abweichungen von der Schaltung enthält. Ambitionierte 700 - Benutzer müssen daher noch eine Weile auf eine Anleitung für ihren ~~Mbyte~~ - Ausbau warten.

Der Ausbau der Bank 15

G. Schumacher

Es ist einigermaßen mühsam, zumindest bis die einfach zu handhabende schnelle Lösung dieses Problems von Andreas zur Verfügung steht, Maschinenprogramme in anderen Speicherbänken als der Bank 15 laufen zu lassen. Zudem gibt es eine sehr große Zahl von Programmen, die nach Speicherplatz in der Bank 15 verlangen. Daher gehört zumindest der Ausbau des für Steckmodule vorgesehene Speicherplatz von \$2000 bis \$7fff in der Systembank meines Erachtens zur Grundausstattung eines CBM 600/700.

Außerdem sind im Bereich \$0000 bis \$1fff noch 6 k ungenutzt, von denen aber unter Umständen 4 k für ein Diskettenbetriebssystem reserviert bleiben sollten.

Viele ernsthaftes Assemblerprogrammierer sind daran interessiert, auch das Basic und das Kernal in einem Schreib- Lesespeicher zur Verfügung zu haben. Eventuell kann es ja auch sinnvoll erscheinen, je nach Anwendung verschiedene, auf bestimmte Anforderungen abgestimmte Basic- bzw. Kernalversionen zur Verfügung zu haben.

Leider kann an dieser Stelle nicht jede der zahllosen Möglichkeiten, entsprechender Änderungen aufgezeigt, geschweige denn detailliert beschrieben werden. So beschränke ich mich hier auf einige wenige in der Hoffnung, daß noch weitere Bauanleitungen hier eingehen.

Die einfachste Möglichkeit, Speicherplatz im Steckmodulbereich zu erhalten, ist natürlich das einstöpseln eines Steckmoduls. Eckart Knösel hat eine Platine ohne Batteriepufferung hierfür entwickelt. Natürlich läßt diese Platine auch die Verwendung von EPROMs zu. Selbst bestückt mit RAM oder EPROM ist so etwas mit maximal DM 60,- einschließlich guter Sockel und Gehäuse zu verwirklichen. Einige ähnliche Platinen mit Batteriepufferung sind ebenfalls entwickelt worden. Ich weiß aber nicht, ob jetzt schon eine solche zu einem moderaten Preis irgendwo erhältlich ist. Persönlich habe ich eher Probleme damit, wenn die Speicher ihren Inhalt nicht verlieren wollen.

Die preiswerteste Möglichkeit, zusätzlichen Schreib- Lesespeicher in der Bank 15 zu erhalten, steht den Besitzern eines auf ein Megabyte aufgerüsteten Rechners offen. Mit einem Aufwand von ca. 6,- Mark (West) ist es möglich, den gesamten Bereich von \$0000 bis \$7fff oder bis \$bfff wahlweise in das dynamische RAM einzublenden.

Wie auch in meiner letzten Bauanleitung ist ein Hueckepack - IC der Kern der Schaltung. Zunächst aber eine kurze Erklärung der Umschaltung zwischen dem dynamischen RAM und der Systembank.

Wird die Systembank angesprochen, liegen alle Adreßerweiterungsleitungen (BP 0 bis BP 4) auf High. Auf nämlichen Zustand werden sie durch ein Gatter von U 20 überprüft. Letzteres, ein nicht Und fühlt sich unter dieser Bedingung genügt, das Signal nichtSYSL0EN zu erzeugen, das sowohl den Hauptdekoder für die Systembank als auch den Datenbustreiber für die Systembank aktiviert, während über das zweite Gatter des selben Bausteins der Datenbus des Prozessors vom dynamischen RAM isoliert wird.

Will man nun die unteren 32 k statt in der Systembank im dynamischen RAM ansprechen, genügt es, die durch U 20 vorgenommene Umschaltung in diesem Bereich zu verhindern, indem man das Gatter praktisch erweitert, damit es nur noch aktiv wird, wenn auch A 15 high ist. In diese Erweiterung kann man ohne weiteres noch A 14 einbeziehen, um auch den Basicbereich in den dynamischen Speicher zu verlagern. Es erscheint sinnvoll, die Schaltung deaktivieren zu können, damit man wahlweise auch eventuelle Steckmodule weiterverwenden kann. Während des Rechnerbetriebes kann eine Umschaltung jedoch nur vorgenommen werden, wenn gleichzeitig der Resetknopf festgehalten wird. Anderenfalls zieht man dem Prozessor die Systeminformationen, die im "DUFFRAM" enthalten sind, unter sämtlichen 40 Beinen weg.

Zur Praxis

Man nehme einen 74S21, 74ALS21, 74F21 oder 74AS21. Die beiden letztgenannten sind mit ähnlicher Vorsicht zu behandeln wie MOS - Bausteine. Nachdem man die zu Pin 13 von U 20 führende Leiterbahn durchtrennt und in die Durchkontaktierung am anderen Ende dieser kurzen Leiterbahn einen Draht eingelötet hat, löte man huckepack auf U 20 einen Schaltkreis eines der oben genannten Typen, dessen Anschlüsse 1,2,4,5 und 6 abgespreizt sind und Pin 8 entfernt ist. Den Ausgang des neuen ICs (Pin 6) verbindet man mit Pin 13 beider ICs, den vorher erwähnten Draht mit zweien der freien Anschlüsse und die beiden anderen freien Anschlüsse mit den Mittelanschlüssen eines 2*um-Schalters mit Mittelstellung aus, den an in die Gehäuserückwand einbauen kann. A15 ist mit beiden Eingängen des neuen ICs, A14 mit einem Eingang des anderen ICs zu verbinden. A15 kommt von U62, Pin 12, A14 von U62 Pin 9. Zwischen den beiden für die Erweiterung verwendeten Gattereingänge und +5V (Pin 14 des selben Bausteins) sind nur noch Widerstände je 1kOhm notwendig. Nun kann man, sofern dynamisches RAM in der Bank 15 vorhanden ist, dort entweder den Bereich bis \$7fff oder den bis \$bfff ins RAM verlagern oder es auch sein lassen.

In der nächsten Ausgabe werden Schaltungen mit statischen RAMs für die Bank 15 und Möglichkeiten, auch das Betriebssystem ins RAM zu verlagern vorgestellt.

Nachlese zur 1 Mbyte - Erweiterung

G. Schumacher

Die Bauanleitung in unserer Nullnummer hat ein erfreuliches Echo gefunden, wenn auch viele Leser durchaus berechtigte Kritik bezüglich des Stils geübt haben. Jedenfalls funktioniert ein genauer Nachbau, abgesehen von dem Druckfehler, für den ich mich entschuldigen möchte. Er war durch die Unleserlichkeit meines Schaltplanes verursacht. Das Nicht-Oder wurde fälschlich mit U 94 statt U 90 bezeichnet.

Zu Punkt 14 des handwerklichen Teils gehört noch, daß vor dem Herstellen der angegebenen Verbindung die vorherige Verbindung von Pin 1 der Speicher mit einem festen Potential aufgehoben werden muß. Natürlich ist es auch wurscht, wie herum man die Keramikkondensatoren einsetzt.

Die Datasette streikt nie wieder

Einer der häufigsten Fehler, der bei der Datasette auftritt, ist ein verstellter Tonkopf. Dieser Fehler macht sich besonders dann bemerkbar, wenn mit Turbo- bzw. Supertape gearbeitet wird. Mit der hier beschriebenen Schaltung läßt sich extrem einfach, ohne jegliches Programm, der Tonkopf an jede Datenkassette anpassen.

Bauanleitung

Die vorliegende Bauanleitung, deren Bauteile zu einem Preis bis ca. 7,- DM zu haben sind, ersetzt ein Oszilloskop. Mit der Schaltung läßt sich eine Spannung -natürlich in gewissen Grenzen- auf ein Maximum abgleichen. Das Herz ist ein Operationsverstärker vom Typ LF 356, der als Komparator (Schwellwertschalter) betrieben wird. Außerdem hat dieser Operationsverstärker gegenüber anderen den Vorteil, daß seine Eingangsstufe aus einem Feldeffekttransistor besteht. Der Eingangswiderstand geht dadurch gegen unendlich und belastet das zu messende Signal in keiner Weise. Mit dem Trimpotentiometer läßt sich eine Schwellspannung einstellen, die laufend mit der analogen Sinusschwingung verglichen wird.

Ist der Momentanwert der Sinusschwingung kleiner als der vorgegebene Schwellwert, führt der Ausgang des LF 356 null Volt. Wird der Momentanwert größer, springt der Ausgang auf +5 Volt und regt dadurch eine Leuchtdiode an. Wird die Schwellspannung in den Scheitelpunkt der Sinusschwingung gelegt, geht die Zeitspanne, in der der Ausgang des Komparators +5 Volt führt, gegen ein Minimum. Daraus folgt, daß die Helligkeit der Leuchtdiode abnimmt, je näher die Schwellspannung an den Scheitelwert der Sinusschwingung rückt. Wird dagegen die Amplitude des Signals, also der Sinusschwingung vergrößert, wird die Helligkeit der Leuchtdiode wieder größer, da die Zeitspanne, in der der Komparator auf 5 V liegt, größer wird. Das ist eigentlich der ganze Abgleichvorgang. Mit dem Trimpotentiometer wird auf minimale Helligkeit und mit der Tonkopfeinstellschraube auf maximale Helligkeit abgeglichen.

Aufgebaut wird die Schaltung auf einer kleinen Lochrasterplatine. Wer sich eine Platine ätzen will, findet das Layout im Maßstab 1:1 in Bild 1. Wie die einzelnen Pins der Bauelemente miteinander verbunden werden, zeigt Bild 2. Achten Sie beim Zusammenbau auf die richtige Polarität der Leuchtdiode (Bild 3). Ist die Schaltung zusammengebaut, muß sie noch im Datasettengehäuse untergebracht werden. Öffnen Sie dazu die Datasette und bohren Sie an geeigneter Stelle ein Loch in das Gehäuseoberteil, so, daß die Leuchtdiode gerade in dieses Loch paßt.

Verbinden Sie die Anschlüsse "+" und "-" (Bild 2) mit den Motoranschlußklemmen. Dabei ist ebenfalls auf die Polarität zu achten. Im allgemeinen ist sie auf dem Motor gekennzeichnet. Sonst: rotes Kabel "+" und schwarzes Kabel "-". Der in Bild 2 mit "E" (für Eingang) gekennzeichnete Punkt muß über ein abgeschirmtes Kabel mit einem der beiden Lötunkte auf der Datasettenplatine (Bild 4) verbunden werden. Die Abschirmung ist an dem mit "-" gekennzeichneten Punkt (Bild 2) und an Masse zu löten. Bei den beiden Lötunkten (Bild 4) handelt es sich um den Ausgang des ersten bzw. zweiten Analogverstärker einer Commodore - Datasette. Geräte anderer Hersteller sind teilweise anders aufgebaut. Es kann vorkommen, daß die beiden in Bild 4 gekennzeichneten Analogverstärker in einem Gehäuse untergebracht sind. In diesem Fall ist der Punkt "E" mit dem Pin 8 dieses ICs zu verbinden. Bei einer Phonemark - Datasette ist es Pin 14.

Einfaches Einstellen

Bevor die Datasette wieder zusammengebaut wird, ist die Schaltung an die Datasettenelektronik anzupassen. Schalten Sie hierzu Ihren Computer ein, legen eine Programmcassette ein und betätigen die "Play"-Taste. Gehen Sie dann folgendermaßen vor:

1. Am Trimpotentiometer drehen bis die Leuchtdiode schwach flackert. Dadurch wird die Schwellspannung in den Scheitelpunkt der Sinusschwingung gelegt.
2. An der Tonkopfeinstellschraube drehen, bis die Helligkeit der Leuchtdiode ihr Maximum erreicht hat.

Letzteres ist nur dann erforderlich, wenn die Datasette nicht optimal eingestellt war. Der Einstellvorgang muß so lange wiederholt werden, bis eine Einstellung erreicht wird, bei der die Leuchtdiode erlischt so bald der Tonkopf verstellt wird. Bauen Sie nun Ihre Datasette wieder zusammen. Vorher aber den Computer ausschalten!

Wollen Sie nun ein Programm laden, das mit einem dejustierten Tonkopf oder einer anderen Datasette aufgenommen wurde, brauchen Sie nur noch nach dem Einschalten des Computers die Kassette einzulegen, die PLAY-Taste zu drücken und so lange an der Tonkopfeinstellschraube zu drehen, bis die Leuchtdiode ihr Helligkeitsmaximum erreicht hat.

Bild 1: Layout

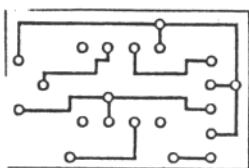


Bild 2: Bestückungsplan (Lötseite)

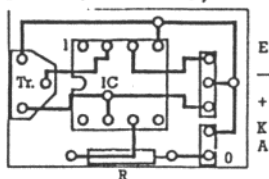


Bild 3: Leuchtdiode A=Anode=längeres Beinchen oder runde Seite

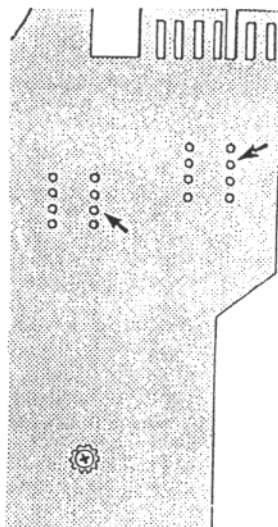
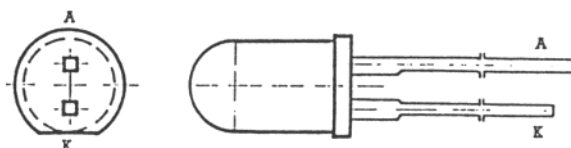


Bild 4: An einem der Punkte ist der Punkt * E * zu löten.

Stückliste

- 1 Widerstand 270 Ohm
- 1 Leuchtdiode
- 1 Operationsverstärker LF 356
- 1 Trimpotentiometer 10 k Ohm

Hier noch meine Adresse :

Peter Oevermann
Johannettentaler Str. 24

4930 Detmold 1
Tel. 05231/26831

Um ein wenig Licht ins Chaos zu bringen, ist eine gewisse Systematik angebracht. Aus diesem Grund haben wir uns überlegt, daß es sinnvoll ist, bestimmte Dinge zu "normen". Viele werden jetzt sicherlich denken, was soll der Quatsch, ich mache alles wie bisher, was die anderen machen ist mir egal. Doch alleine das Problem, einen Austausch von Public Domain Software zu organisieren oder Hardwareentwicklungen zueinander kompatibel zu halten, machen eine solche Maßnahme notwendig. Deshalb möchten wir Ihnen in dieser Ausgabe von HSG einige Vorschläge zu einer gleichen Vorgehensweise unterbreiten.

Als Erstes schlage ich vor, die Diskettenhandhabung zu standardisieren. So ist es sinnvoll, jede Diskette mit einer anderen ID-Nummer zu versehen, da es Programme zur Diskettenverwaltung gibt, die die ID-Nummern unterstützen.

Als Zweites sollte versucht werden, in den Filenamen möglichst viele Informationen zu integrieren. Deshalb schlage ich vor, den Filenamen mit einem Kürzel enden zu lassen, das drei Zeichen lang ist und Informationen über den Datentyp des Files beinhaltet, so wie es auch bei anderen Rechnern durchaus üblich ist. Diese Kürzel sollten mit einem Punkt von dem Filenamen abgetrennt werden. Die Art der Filekennzeichnung ist dann auch kompatibel zu CP/M und Supertape.

Folgende Kürzel stelle ich hiermit zur Diskussion:

```
:bas - File mit Basic-Text
:mas - File mit Macro-Basic-Text
:loa - Lade- bzw. Installationsprogramm
:src - Quelltext (leider leicht mit scr: zu verwechseln)
:ass - Assembler-Quelltext
:obj - Objektfile (compiliertes Programm)
:dat - Datenfile
:cod - Maschinencode
:txt - Textfile einer Textverarbeitung
:let - Brief (letter)
:fmt - formatierter Textfile
:scr - Bildschirminhalt (screen)
:sys - Betriebssystem und Erweiterungen (Bank 15)
:doc - Dokumentationsfile
```

Falls Ihnen das eine oder andere Kürzel noch fehlt, bzw. nicht gefallen sollte, so bitte ich Sie, mir dies mitzuteilen, um sich so an der Diskussion zu beteiligen.

An dieser Stelle möchte ich noch auf einen Trick hinweisen, wie Sie Programme aus dem Directory laden können, indem Sie vor den Directoryeintrag dload schreiben, ohne, daß Sie den Anhang löschen müssen.

Wenn Sie z.B. den Textfile "Beispiel" abspeichern wollen, geben Sie nach dem Namen ein SHIFT SPACE anstatt eines Anführungszeichens und danach einen Doppelpunkt ein! In dem Directory steht dann:

```
15 "Beispiel.txt": prg
```

Dies hat nun den Vorteil, das es möglich ist, im Directory mit dem Cursor in die entsprechende Zeile zu gehen und die Blockanzahl (hier 15) mit DLOAD zu überschreiben. Der Rechner lädt dann das Programm "Beispiel", ohne daß ": prg" gelöscht werden müßte.

Ein weiteres Anliegen von uns ist es, die Software einheitlich zu dokumentieren. Hierzu haben uns überlegt, daß es am sinnvollsten wäre ein Programm zu schaffen, mit dem es möglich ist, zu jedem Programm einen Datenfile einzurichten, auf dem dann alle wichtigen Informationen über das Programm abgelegt werden sollten. Folgende Informationen halten wir für relevant:

- Art des Files (insofern noch nicht im Kürzel enthalten)
- Name und Anschrift des Autors sowie Copyrightinweise (handelt es sich evtl. um Public Domain Software?)
- Welche Files werden für das Programm benötigt?
- relocatibel? bzw. erforderlicher Speicherbereich
- bei Basic-Programmen: Welche Basicversion?
- wurde Macro-Basic verwendet?
- lauffähig auf welchen Rechnern?
- ist das Programm 8432 (8032-Emulator) erforderlich?
- welche Datenspeicher unterstützt das Programm?
- welche Drucker unterstützt das Programm?
- welche weitere Hardware ist erforderlich?
- Kurze Erläuterung der Wirkungsweise des Programmes
- Kurze Erklärung des Programmes und Hinweise zur Benutzung

Es wäre schön wenn sich jemand finden würde, der Lust hat sich daran zu machen ein solches Programm zu entwickeln (siehe auch: Es gibt viel zu tun ...). Vielleicht hat der ein oder andere noch weitere Ideen und Anregungen, wie sich die Software sinnvoll und umfassend dokumentieren läßt.

LISTINGS-EINGEBEN: Wir haben einige interessante Listings, die darauf warten eingegeben bzw. umgeschrieben zu werden. Die Programme für den Apple erfordern eine gewisse Kenntnis des Apple-Basic.

Hier eine Liste der zur Verfügung stehenden Listings:

- * Formelsammlung zur Numerischen Mathematik mit Basic-Programmen (Microsoft MBasic)
- * Basic im Büro Band 4; Auftragsabwicklung und Dateiverwaltungssysteme (Applesoft II-Basic, DOS 3.3)
- * Basic aus der Praxis; Programm-Beispiel für kaufmännisch orientierte Berufe
- * Basic aus der Praxis Band 4; Typische Programm-Beispiele für Kaufleute (Microsoft MBasic)
- * Basic Sammlung Band 1 und 4 (für verschiedene Computer)
- * CBM Spaß und Spiele; Ausgewählte CURSOR Programme
- * Basic-Anwenderprogramme (Zahnprofile, Solaranlagen, Beleuchtung, Netztransformatoren, Ampel, Rohrnetz, Stundenplangestaltung; für verschiedene Computer)

Wie Sie sehen können gibt es eine Menge Arbeit und jeder sollte sich überlegen ob er mitarbeiten kann. Ohne Ihre Mitarbeit hat der 6xx/7xx nur einen Bruchteil des Wertes, den er haben könnte und hoffentlich auch bald hat.

Änderung der Device-Nummer bei verschiedenen Massespeichern

Jörg Linnig

Die Änderung der Device-Nummer ist in der Regel problemlos. Die Abfrage der Geräteadressen ist bei den meisten Geräten von Commodore (außer CBM 2031) gleich.

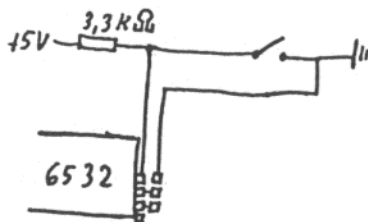
Verantwortlich für die Hardware-Selektion der Geräteadresse sind die Leitungen PBO - PB2 (Pin 24-22) des 6532. Dieser "Memory, I/O, Timer" - Baustein der 6500 Familie ist an den bezeichneten Pins gegen Masse geschaltet.

Wenn diese drei Pins gegen Masse geschaltet sind hat das Gerät die Device-Nummer 8. Um diese Adresse zu ändern müssen Leiterbahnen aufgetrennt werden. Die sich daraus ergebende Geräteadressen sind der folgender Tabelle zu entnehmen:

xx = Pin hat Kontakt mit der Masse
oo = Pin hat kein Kontakt mit der Masse

Ger.Nr.	1	Pin 22	1	Pin 23	1	Pin 24	1
1	8	1	xx	1	xx	1	xx
1	9	1	xx	1	xx	1	oo
1	10	1	xx	1	oo	1	xx
1	11	1	xx	1	oo	1	oo
1	12	1	oo	1	xx	1	xx
1	13	1	oo	1	xx	1	oo
1	14	1	oo	1	oo	1	xx
1	15	1	oo	1	oo	1	oo

Wer seine Geräteadresse ändern möchte ohne diese umschalten zu können, dem empfehle ich einfach die entsprechenden Beinchen des 6532 wegzubiegen, so daß sie keinen Kontakt mehr haben. Dies empfehle ich auch dann, wenn jemand einfach ersteinmal ausprobieren möchte, ob er auch alles richtig gefunden hat. Soll die Geräteadresse jedoch umschaltbar gemacht werden, so werden die Leiterbahnen aufgetrennt und ein Umschalter in folgender Art und Weise eingelötet:

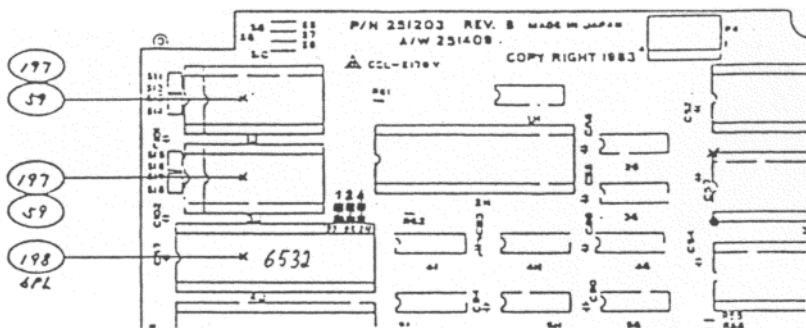


im Prinzip wäre es ausreichend einfach nur Leiterbahnen auf zu trennen und einen Umschalter einzulöten. Diese Vorgehensweise hat jedoch den Nachteil, daß es zu einem undefinierten Zustand kommen kann und das Laufwerk nicht mehr "weiß" ob es nun die Nummer 8 oder 9 oder überhaupt hat.

Um die Suche nach dem 6532 nicht zu einem Ostereiersuchen ausarten zu lassen hier nun einige Wegweiser für verschiedene Massenspeicher:

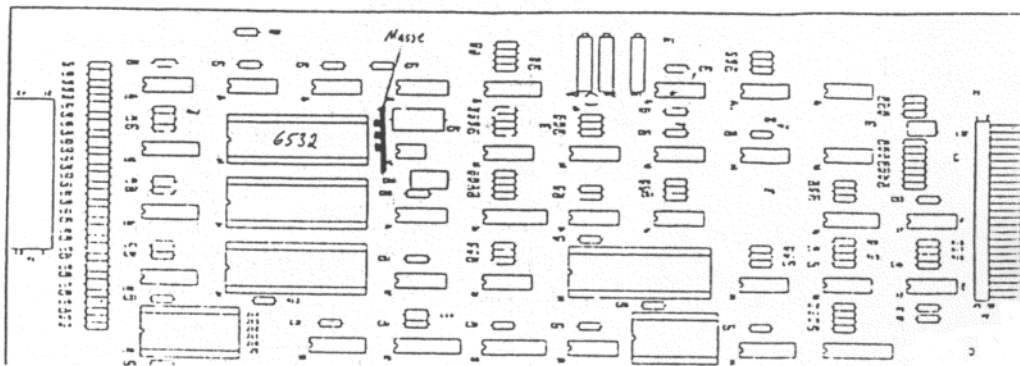
SFD 1001

Bei abgenommen Gehäusedeckel findet man den 6532 mit der Bauteilbezeichnung 4j am dritten Steckplatz von links. An seinem linken unteren Ende sind die drei Masseverbindungen. Sie sind nicht durchkontaktiert! Bei der SFD ist es besonders einfach die Leiterbahnen zu durchtrennen, denn die Lötunkte mit den dünnen Leiterbahnen befinden sich direkt neben dem 6532. Hier aber trotzdem noch einen kleinen Ausschnitt aus dem Bestückungslayout:



CBM 8280

Hier befindet sich der 6532 ebenfalls am dritten Steckplatz von links (wenn man von vorne schaut). Er hat die Bauteilbezeichnung 9g. Vorne rechts befinden sich die drei Lötunkte, die hier gegen Masse geschlossen sind. Sie sind durchkontaktiert und haben Verbindung mit den Pins 22-24. Wer sich dazu entscheiden sollte Leiterbahnen durchzutrennen, sollte meiner Meinung nach die Leiterbahnen auf der Rückseite der Platine durchtrennen, da hier das Layout der Leiterbahnen nicht so einfach zu durchschauen ist wie bei der SFD 1001. Wer die Lötunkte auf der Vorderseite durchtrennen möchte: der linke Lötunkt geht an Pin 22. Um die Suche weiter zu vereinfachen, habe ich hier ebenfalls einen Ausschnitt aus dem Bestückungslayout beigelegt:



CBM D9060/D9090

Bei den Festplattenlaufwerken hat der entsprechende 6532 die Bauteilebezeichnung 7G.

CBM 4040/8050/8250

Bei diesen Diskettenstationen hat der entsprechende 6532 die Bauteilbezeichnung UE1.

Wer die Gerätenummer an dem Diskettenlaufwerk CBM 2031 ändern möchte, dem empfehle ich das Commodore Handbuch: Disk System User Reference Guide.

Quellen:

- Disk System User Reference Guide.
- Informationen von Axel Mathias
- Ausprobieren

Kopien wichtiger Unterlagen

G. Schumacher

Folgende Unterlagen stehen zusätzlich zu der alten Liste als Kopie - außer dem Typenradkatalog nur für den privaten Gebrauch - zur Verfügung:

31. Servivehandbuch 8028 incl. relativ gut lesbarem Schaltplan (A3) DM 11,55

32. Typenradkatalog CUM 8028 DM 3,30
Demnächst werden durch unsere Sammelbestellungen über 40 verschiedene Typenräder zum Preis von ca. 30,- DM verfügbar sein. Allerdings muß mit langen Lieferzeiten gerechnet werden (mindestens 4 Wochen). Vorbestellungen werden schon angenommen.

33. CUM 6400 User's Manual DM 4,80

34. Schaltplan CUM 8088 - Coprozessorkarte DM 1,35

Kopien von Handbüchern u.ä. können auch gebunden werden. Aufpreis DM 5,--

Die ziemlich genau ausreichende Versandkostenpauschale beträgt DM 4,--

Alles kann bei mir zu den bekannten Bedingungen angefordert werden.

p.s.:

Gory van Beek hat mit damit angefangen, ein Kernallisting-CBM 500 zu dokumentieren. Denen, die daran weiter arbeiten wollen, können Kopien zu DM 16,50 angefertigt werden. Erstaunlich viele Mitbenutzer haben einen 500er. Vielleicht sollten sie sich auch in ihren besonderen Angelegenheiten um einen Erfahrungsaustausch bemühen. Übrigens: Wie erzeugt man Sprites?

Tips, Tricks und kleine Programme für den CBM 610 - 720

Folge 1

Günter Brühl 11/86

Ich möchte mit dieser Serie versuchen, für die bisher (und auch zukünftig) an mich herangetragenen Probleme der CBM 600/700 User in BASIC bzw. in Maschinensprache, Lösungsansätze aufzuzeigen.

Ich möchte aber keine kompletten Problemlösungen innerhalb dieser Zeitschrift bieten, da ich die Kreativität des Anwenders nicht sofort im Keim ersticken möchte.

Da niemand vollkommen ist, und ich diesen Anspruch auch für mich nicht erhebe, bitte ich die Leser, die meine Fehler ausbaden müssen, mich auf diese hinzuweisen und vielleicht auch Ihre jeweiligen Probleme damit etwas darzustellen.

Um nicht immer so viel schreiben zu müssen werde ich die Rechner der Serie CBM 610 bis 720 kurz 700'er nennen.

So, genug der Vorrede, ein erstes Problem:

Die Programmierung der RS-232

Der 700'er ist mit einer der Norm entsprechenden RS-232-Schnittstelle ausgerüstet. Diese Schnittstelle wird vom ACIA 6551 (Asynchroner Kommunikation Interface Adapter oder so ähnlich) bedient.

Dieser Baustein enthält folgende Funktionen:

- Generierung einer Baudrate (Bits/Sekunde) zwischen 50 und 19200 Baud
- Echo Mode
- Bidirektionaler Datentransfer
- Transferraten bis zu 125000 Baud bei externem Takt
- Programmierbare Wortlänge (5 - 8 Bit)
- Programmierbare Anzahl von Stoppbits
- Generierung und Erkennung von Parity
- Full and Half Duplex-Betrieb

Folgende Signale sind verfügbar:

Pin	Bezeichnung	Datenrichtung
1	Shield	- Schutzerde
2	TxD	- Transmit data Ausgabe
3	RxD	- Receive data Eingabe
4	RTS	- Ready to send Ausgabe
5	CTS	- Clear to send Eingabe
6	DSR	- Data set ready Eingabe
7	Ground	
8	DCD	- Data carry detect Eingabe
20	DTR	- Data terminal ready Ausgabe
24	XMIT CLK	- Transmit clock Ein- /Ausgabe

Ist diese Schnittstelle nicht einfach zu bedienen, steht in der Literatur nicht, daß man nur 3 Leitungen benötigt?

Im Prinzip ja, doch die Commodore Entwickler nahmens mal wieder viel zu genau. Der CBM 700 benötigt an allen Pin's, außer an Pin

24, den richtigen Signalpegel. Sollte Ihr anzuschließendes Gerät, was meistens der Fall ist, nicht über alle Leitungen verfügen, so fügen Sie bitte beim Stecker auf der CBM-Seite folgende Brücken ein:

- Pin 4 mit Pin 5
- Pin 6 mit Pin 8 und Pin 20

sonst läuft nämlich fast gar nichts.

Ein zweites Problem ist die Kreuzung der Datenleitungen und zwar muß fast immer Pin 2 (Txd) mit Pin 3 (Rxd) des Gegenüber verbunden werden (ist doch eigentlich logisch) und natürlich Pin 3 (Rxd) mit Pin 2 (Txd) des Gegenüber.

Sollten alle Signale beim Gegenüber vorhanden sein, wird es etwas komplizierter (Denksportaufgabe: wieviel Kreuzungsmöglichkeiten gibt es?). Jedoch läßt sich auch so kein Hardware-Handshake realisieren, da der CBM 700 auf den Überlauf des Puffers (Empfangen) bzw. auf das Setzen der Handschake-Signale durch das Gegenüber, nur mit dem Setzen der Status-Variablen 'st' reagiert und dann fröhlich weiterwurstelt. Natürlich kann man diese abfragen (siehe unten) und so wenigstens eine Software-Handshake realisieren.

Sollten sie mal eine Leitung falsch gelötet haben, so brauchen sie keine Befürchtungen zu haben, daß der Baustein gleich den Geist aufgibt. Meiner hält das schon seit 3 Jahren aus.

Nun zur Programmierung:

OPEN Kanalnummer, 2, Sekundäradresse, Bytestring öffnet den Kanal zur RS-232 (Adresse 2) und weist ihn der Kanalnummer zu.

Die Kanalnummer kann Werte zwischen 0 und 255 annehmen. Wenn eine Kanalnummer größer 127 gewählt wird, wird bei einem CR (Carriage Return) zusätzlich ein LF (Line Feed) geschickt.

Die Geräteadresse 2 ist unbedingt notwendig!

Die Sekundäradresse beeinflusst die Datenrichtung:

- 1 nur Senden
- 2 nur Empfangen
- 3 Senden und Empfangen

Wenn man zu diesen Werten 128 addiert werden die ankommenden bzw. gesendeten Zeichen von CDM-ASCII zu Standard-ASCII und umgekehrt konvertiert.

Der Bytestring setzt sich folgendermaßen zusammen:

Das erste Byte beeinflusst das Control Register, wobei die 3 gewählten Werte addiert werden müssen.

Baudrate (Bit 0 bis 3): Bit 4 muß bei angegebener Baudrate gesetzt werden (zu den Werten 1 bis 15, 16 hinzuaddieren)

External Rate x 16	Dezimalwert, der zu addieren ist
50	0 (nur Empfangen)
75	1 + 16 = 17
109.92	2 "
134.58	3 "
150	4 usw ..
300	5
	6

600	7
1200	8
1800	9
2400	10
3600	11
4800	12
7200	13
9600	14
19200	15

Bit 5 und 6 beeinflussen die Wortlänge:

8 Bit	0
7 Bit	32
6 Bit	64
5 Bit	96

Bit 7 setzt die Stop-Bits:

1 Stop-Bit	0
2 Stop-Bits	128
1 Stop-Bit bei 8 Bit Wort und Parität	128
1.5 Stop-Bits bei 5 Bit Wort und keine Parität	128

Das zweite Byte beeinflußt die restlichen Parameter, wobei auf die implementierten Sonderfunktionen nicht eingegangen wird:

Bit 4 (Modus)

Normal	0
Echo	16 (Daten werden an Sender sofort zurückgeschickt - Duplex-Betrieb)

Bits 5,6,7 (Parität)

keine	0
ungerade	32
gerade	96
mark (transmitted)	160
space (transmitted)	224

Byte 3 und 4 sind nicht implementiert, werden jedoch benötigt.

Beispiel:

1200 Baud, 8 Bit, 2 Stop-Bits, Normal, gerade Parität, nur senden, Code-Konvertierung

```
open 2,2,129,chr$(24+0+128)+chr$(0+96)+"xx"
```

angesprochen wird sie nun mit print#2,; oder: input#2 ...

Nun noch zur Status-Variable 'st':

Diese Variable dient beim CBM zur Wiedergabe des I/O-Status. Gegenüber den sonstigen Werten kann sie bei der RS-232 folgende Werte annehmen:

0	alles okay
1	Paritätsfehler (nur beim Lesen)
2	Formatfehler (nur beim Lesen; z.B. falsche Wortlänge
4	Datenverlust (nur beim Lesen; Daten wurden schneller

angeliefert, als sie ausgelesen wurden)
 16 Eingabepuffer leer (nur beim Lesen)
 32 Fehler bei DCD 'Data Carrier Detect'
 64 Fehler bei DSR 'Data Set Ready'

Evtl. können mehrere Fehler gleichzeitig auftreten. In diesem Fall sind die einzelnen Fehlermeldungen auszublenden.

Wichtig: Beim ersten Auslesen der Variablen 'st' wird ihr Wert auf Null zurückgesetzt. Soll sie später noch verwendet werden, so ist ihr Wert einer anderen Variablen zuzuweisen. Des weiteren wird sie bei einer folgenden I/O-Operation (z.B. print) neu gesetzt.

Programmübernahme vom CBM 700 <-> CBM 8032

Da die BASIC-Programme des CBM 700 an die in den Speicherstellen 45 und 46 enthaltene Adresse in Bank 1 geladen werden, (CBM 8032 an 1025) wurde aus Kompatibilitätsgründen der dload-Befehl so gestaltet, daß er Programme grundsätzlich an die besagte Speicherstelle lädt, unabhängig davon, welche Adresse im File-Header eingetragen ist. Beim CBM 8032 lädt jedoch der dload-Befehl an die eingetragene Adresse. Dies führt bei der Übernahme von Programmen des 700'er zum Absturz des 8032 (Zeropage wird überschrieben). Man kann dies wie folgt vermeiden (im 700'er):

```
dsave "prog700"
poke 45,1: poke 46,4          Setzen der BASIC-Anfangszeiger
                                auf 1025
dload "prog700"
dsave "prog8032"
```

Übrigens: Mit poke 212,xx kann man sich einen schöneren Cursor generieren (z.B. xx = 72)

Viel Spaß beim probieren, bis zum nächsten Mal

P.S.: Anregungen bzw. Kritik nehme ich gern entgegen, solange sie nicht vernichtend ist!

Literaturhinweis

G. S.

Die einzige mir bekannte Zeitschrift, die sich bislang regelmäßig mit dem CBM 600/700 beschäftigt hat, ist das "MICRO MAG". Der Herausgeber bietet uns die 9 Hefte, die entsprechende Artikel enthalten, einschließlich der aktuellen Ausgabe vom Januar d. J. zum Paketpreis von DM 50,- an.

Hierin findet man u.a. einen einfachen Eprommer mit Programm, für den allerdings die Programmierspannung extern zugeführt werden muß, diverse Terminalprogramme zum Betrieb anderer Rechner über den CBM, Softwarerezensionen, einen Texteditor und einige Programmierhilfen.

Die Anschrift: Roland Löh, Hansdorfer Str. 4, 2070 Ahrensburg
 tel. 04102/55816

1. Für die Übertragung zwischen Computern (Dreidrahtbetrieb)

Pin	Belegung	Bezeichnung
1	nicht benutzt	
2	Pin 3 des Gegensteckers	(transmitted data / Daten senden)
3	Pin 2 des Gegensteckers	(received data / Daten empfangen)
4	Brücke zu Pin 5	(request to send)
5	Brücke zu Pin 4	(clear to send)
6	Brücke zu Pin 8 und Pin 20	(data set ready)
7	Pin 7 des Gegensteckers	(gemeinsame Masse)
8	Brücke zu Pin 6 und Pin 20	(data carrier detect)
9	nicht benutzt	
10	nicht benutzt	
11	nicht benutzt	
12	nicht benutzt	
13	nicht benutzt	
14	nicht benutzt	
15	nicht benutzt	
16	nicht benutzt	
17	nicht benutzt	
18	nicht benutzt	
19	nicht benutzt	
20	Brücke zu Pin 6 und Pin 8	(data terminal ready)
21 - 25	nicht benutzt	

2. Für den Anschluß eines Akustikkopplers

Pin	Belegung	Bezeichnung
1	nicht benutzt	
2	Pin 2 des Akustikkopplers	(transmitted data / Daten senden)
3	Pin 3 des Akustikkopplers	(received data / Daten empfangen)
4	nicht benutzt	
5	Pin 5 des Akustikkopplers	(clear to send)
6	nicht benutzt	
7	Pin 7 des Akustikkopplers	(gemeinsame Masse)
8	nicht benutzt	
9	nicht benutzt	
10	nicht benutzt	
11	nicht benutzt	
12	nicht benutzt	
13	nicht benutzt	
14	nicht benutzt	
15	nicht benutzt	
16	nicht benutzt	
17	nicht benutzt	
18	nicht benutzt	
19	nicht benutzt	
20	Pin 20 des Akustikkopplers	(data terminal ready)
21 - 25	nicht benutzt	

Programmbeschreibung

Als Besitzer eines älteren Commodore 3032 legte ich mir, durch den günstigen Preis angelockt, einen CBM 610 zu. Zu meinem Vergnügen liefen die meisten auf einer Floppy Typ 8050 abgespeicherten Basicprogramme des CBM 3032 auf den "Neuen". Umgekehrt jedoch schickten Versuche, die mit dem CBM 610 editierten Programme von der Floppy in den 3032 zu laden, diesen in einen nicht mehr ansprechbaren Zustand. Eine Analyse des Problems zeigte, daß der CBM 600/700 vermittelt eines Relativladers die für den Speicherbeginn ab 1025 des 3032 abgelegten Programme problemlos in den ab \$0003 beginnenden Programmspeicher laden kann. Der 3032 lädt die Programme jedoch absolut, die des 610 also in die 0-Page!

Um dies zu verhindern müssen die mit abgespeicherten Zeiger auf den Speicherbeginn und auf die folgenden Programmzeilen im Programmfile geändert werden. Dies leistet dieses Programm. Filetrans8050 wurde auf folgender Hardware erfolgreich getestet: Commodore 3032, 4032, 8032 und 610 sowie 8050 und 8250. Es ist möglichst komfortabel gestaltet und führt den Benutzer durch Dialoge. Da es bis auf den Befehl "Directory" in Zeile 50330 im Commodore-Basic 3.0 geschrieben wurde, kann es sicherlich als Vorbild für die Lösung ähnlicher Probleme auf anderen Rechnern dienen. Speziell für die Commodore-Rechner mit 80 Zeichen Schirmen sind die Befehle zum setzen der Fenster - "print chr\$(15)" in Zeile 50030 und "print chr\$(143)" in 50090.

Nach dem Eröffnen eines Lese- und eines Schreibkanals (bis Zeile 50110) wird der Zeiger (Byte 1 und 2 im Pgm-File) auf den ersten freien Platz im RAM gesetzt (\$0003 -> \$0401) und abgespeichert (Zeile 50120 & 50130). Sodann wird der erste Zeiger auf die nächste Programmzeile zwischengespeichert (in a1\$ und b1\$), durch Addition von 4 zum Highbyte (b\$) und Subtraktion von 2 vom Lowbyte (a\$) korrigiert und abgespeichert (Zeilen 50140-50160). Gleichzeitig wird ein Zähler initialisiert, der es erlaubt, den nächsten Zeiger auf die folgende Basiczeile zu finden. In einer Schleife werden nun die Inhalte der Programmzeilen unbeeinflusst übertragen (Zeilen 50170-50210) und der nächste Zeiger gesucht, korrigiert (50220-50290) und auch in den File#2 übertragen. Die restlichen Programmzeilen dienen dem Schließen der Kanäle sowie der Benutzerführung und der Fehlerbehandlung.

Die Anweisung "if f then" in Programmzeile 50310 entspricht übrigens dem "if f<>0 then" und spart 2 Byte. Sollten Sie sich beim Austesten des Programms mit dem auf der Diskette beigefügten Programmfile des CBM 610 - dem überarbeiteten QTH-Kennerprogramm aus IC 9/83, S.61 - wundern, woher in Zeile 300 das "out of memory error" stammt, hier ist die Erklärung: Es handelt sich nicht um einen Fehler des FILETRANS8050, sondern hier wird ein Token des CBM 610 für "ELSE" vom CBM 4032 falsch interpretiert. Die neu verfügbaren Befehle des CBM 600/700 werden also nicht erkannt und müssen von Hand korrigiert werden.

Benutzte Variablen:

Name	Bedeutung	Crossreference
FS	Einlesevariable für den Quellfilenamen und andere Variablen für die Fehlerbehandlung	50040, 50050, 50060, 50340, 50350
f, f\$, f1, f2, f3	Drivenummer für Zielfile	50060, 50110, 50310, 50440, 50315
q\$, q	Name für Zielfile	50070, 50110, 50120
y\$	Einlesevariablen für den Filetext	50080, 50100, 50110
a\$, b\$		50120, 50130, 50140, 50160, 50170
		50200, 50210, 50220, 50240, 50270
		50280, 50284, 50290
st	Status für I/O-Operationen	50060, 50110, 50120, 50140, 50170
		50220, 50310, 50370
a1\$	Zwischenspeicher für Zeiger (Lowbyte)	50160, 50210, 50270
b1\$	Zwischenspeicher für Zeiger (Highbyte)	50160, 50270
a2, b2	Zeichenzähler (low, high)	50160, 50180, 50220
a2\$, b2\$	Zeichenzähler für Vergleich	50190, 50210, 50240, 5260
p	Hilfsvariable für Sprungberechnung	50370, 50380
a	Hilfsvariable für Pointerkorrektur	50280, 50284, 50286, 50288

```

50000 rem"filetrans8050: Pgm zum Umformen von Pgm-Files des CBH800/100 zur
50005 rem"der Floppy 8050 fuer die CBM-Serie 3/4/8032 durch Zeichenweises
50010 rem"Uebertragen in einen neuen Pgm.File. V 1.2 22.08.1986
50015 rem"Hardware: CBM 3032,4032,8032,610,620; CBM 8050,8250, STD1001
50020 rem"(C) 1986 Wolf-Dieter Krueger, Schwewe 5b, 4519 Glandorf
50024 :
50025 rem ##### vorbereitung #####
50026 :
50030 print"b":print$pc(6)chr$(15):open15,8,15:rem clr/home & fehlerkanal offnen
50040 input"Name des Quellfiles (e=Ende): ";zf:open1,8,0,(zf):print
50050 if zf="e" or zf="end" then end
50060 open1,8,0,(zf):input#15,f,f$,f1,f2:f3=st:print:if f then 50370
50070 input"Zielfile auf Drive 0 oder 1 anlegen ";qf;q=val(qf)
50080 input"Name des Zielfiles (e=Ende): ";yf
50090 print"#####"spo(5)chr$(143):rem unter rechte fensterecke
50100 if yf="e" or yf="end" then end
50110 open2,8,1,qf+":(yf):input#15,f,f$,f1,f2:f3=st:print:if f then 50370
50114 :
50115 rem ##### setzen des zeigers auf den ram-anfang #####
50116 :
50120 get#1,a$:get#1,b$:if st then 50370
50130 a$=chr$(1):b$=chr$(4):print#2,a$;:printasc(a$):print#2,b$;:printasc(b$)
50134 :
50135 rem ##### setzen des ersten zeigers auf die naechste pgm-zeile ###
50136 :
50140 get#1,a$:get#1,b$:if st=64 then 50300
50150 if len(b$)=0 then b$=chr$(0)
50160 a1=a$:b1=b$:a2=04:b2=0:a$=chr$(asc(a$)-2):b$=chr$(asc(b$)+4):goto50290
50164 :
50165 rem ## suchen nach dem naechsten zeiger und uebertragung der pgm-zeile ##
50166 :
50170 get#1,a$:if st=64 then 50300
50180 a2=a2+1:ifa2=256thena2=0:b2=b2+1
50190 a2$=chr$(a2)
50200 if len(a$)=0 then a$=chr$(0)
50210 if a2$(a1) then print#2,a$;:print asc(a$):goto 50170
50214 :
50215 rem ##### korrektur und uebertragung des naechsten zeigers #####
50216 :
50220 get#1,b$:if st=64 then 50300
50230 a2=a2+1:ifa2=256thena2=0:b2=b2+1
50240 b2$=chr$(b2)
50250 if len(b$)=0 then b$=chr$(0)
50260 if b2$(b1) then 50290
50270 a1=a$:b1=b$:b1=chr$(asc(b$)+4)
50280 a=asc(a$)-2:if sgn(a)>0 then 50288
50284 b1=chr$(asc(b$)-1):if a=-1 then a=255:goto 50288
50286 if a=-2 then a=254
50288 a$=chr$(a)
50290 print#2,a$;:printasc(a$):print#2,b$;:printasc(b$):goto50170
50300 print#2,chr$(0):chr$(0);:print" 0":print" 0"
50310 f3=st:print"st="f3:print"#####"
50315 input#15,f,f$,f1,f2:close15:if f then 50370
50320 print"Der modifizierte File wurde erfolgreich auf Disk d"qf" abgelegt!";:pr
50330 directory d(q):print"###Soll noch ein Programm bearbeitet werden (j/n)? ";
50340 get zf:ifzf=""then50340
50350 print"q"zf:if zf="j" then run
50360 end
50365 rem ##### fehlerbehandlung #####
50370 print:p=f3:if p>2 then p=abs(p/64+2):print"Status="f3;
50380 on p goto 50400,50410,50420,50430
50390 print" Kein Floppyfehler. ":goto 50440
50400 print" Zeit bei Listener ueberschritten":goto50440
50410 print" Zeit bei Talker ueberschritten ":goto50440
50420 print" Ende oder Kennzeichnung (EOI ist low-> richtig)":goto50440
50430 print" Geraet nicht vorhanden"
50440 print:print"Floppyfehler: "f,f$:close15:for i=1to2000:next:run
50535 rem W(0) 1986 Wolf-Dieter Krueger, Glandorf

```

ready.

```

; #####
; #
; #          8000 punkte-grafik
; #          p.teuber
; #          am steinbrink 14b
; #          4352 herten
; #          tel.02366/43936
; #
; #####

```

```

*** der x wert wird in die speicherstelle 2045 gepokt ***
*** der y wert wird in die speicherstelle 2046 gepokt ***
*** ob ein punkt gesetzt oder geloescht werden soll ***
*** wird in die speicherstelle 2047 gepokt ***
*** 0 = punktloeschen ***
*** 1 = punktsetzen ***
*** z.b. poke2045

```

```

*** die aufloesung betraegt in x 160 punkte (0-159) ***
*** die aufloesung betraegt in y 50 punkte (0-49) ***
*** der nullpunkt liegt links unten ***
*** aufgerufen wird die routine mit sys1860 ***
*** die bildschirmadresse wird nach der formel ***
***  $53248 + ( 8 * ( 4 * ( y \text{ and } 254 ) + y \text{ and } 254 ) ) + ( x / 2 )$  ***
*** berechnet es ist das selbe wie ***
***  $53248 + 80 * \text{int} ( y / 2 ) + \text{int} ( x / 2 )$  ***

```

```

*** das assemblerlisting dokumentiert sich selber ***

```

```

;=$0744 ; start
i6509=$01
lobild=$f0 ; lowadresse vom bildschirm
hibild=$f1 ; highadresse vom bildschirm
xkoor=$07fd
ykoor=$07fe
puset=$07ff
;
;
0744 a9 0f lda #$0f
0746 85 01 sta i6509
0748 a9 31 lda #$31
074a 38 sec
074b ed fe 07 sbc ykoor ; y wert
074e c9 32 cmp #$32 ; vergleichen ob y p 50
0750 b0 1d bcs br001 ; wenn ja dann rucksprung nach basic
0752 29 fe and #$fe ; 2*int(y/2)
0754 0a asl ;
0755 0a asl ; *4
0756 ea nop
0757 85 f0 sta lobild ; y nach lowadresse speichern
0759 a0 d0 ldy #$d0 ; high vom bildschirm anfang laden
075b a2 09 ldx #$09 ; *10
075d 18 br003 clc
075e 65 f0 adc lobild
0760 90 01 bcc br002
0762 c8 iny ; ueberlaufkorrektur lobild
0763 ca br002.dex
0764 d0 f7 bne br003
0766 85 f0 sta lobild
0768 ad fd 07 lda xkoor x wert laden
076b c9 a0 cmp #$a0 vergleichen ob x p 160
076d 90 01 bcc br004
076f 60 br001 rts
0770 4a br004 lsr ; x/2

```

```

0771 90 03      bcc br005
0773 a2 10      ldx #$10          Tabellenauswahl
0775 18         clc
0776 65 f0      br005 adc lobild
0778 90 01      bcc br006
077a c8         iny
077b 85 f0      br006 sta lobild ; aktuelle bildschirmadresse speichern
077d 84 f1      sty nibild
077f ad fe 07   lda ykoor          ; y wert laden
0782 29 01      and #$01
0784 f0 04      beq br007
0786 8a         txa                ; tabellenauswahl
0787 09 20      ora #$20
0789 aa         tax
078a ad ff 07   br007 lda puset ; setzen oder loeschen
078d d0 01      bne br008
078f e8         inx
0790 a0 00      br008 ldy #$00
0792 b1 f0      lda (lobild)
0794 a0 08      ldy #$08
0796 d0 b3 07   br010 cmp tab
0799 f0 06      beq br009          zeichen gefunden ?
079b e8         inx
079c e8         inx
079d 88         dey                ; naechstes zeichen vergleichen
079e d0 f6      bne br010          ; fertig ?
07a0 60         rts                ; nicht gefunden
07a1 ad ff 07   br009 lda puset ; setzen/loeschen
07a4 f0 03      beq br011
07a6 e8         inx
07a7 d0 01      bne br012
07a9 ca         br011 dex
07aa a0 00      br012 ldy #$00
07ac bd b3 07   lda tab
07af 91 f0      sta (lobild)
07b1 60         rts                ; zurueck nach basic
07b2 00         brk
07b3          tab

```

```

.bytf#32 123 108 98 124 255 126 97 127 252 225 254 ; code tabell
.bytf#226 236 251 160 32 108 123 98 124 225 126 127 255
.bytf#254 226 251 97 252 236 160 32 126 123 97 108 127
.bytf#124 226 255 236 98 252 225 251 254 160 32 124 123
.bytf#255 108 225 126 226 127 251 98 254 97 236 252 160
.end

```

```

07b3 20 7b 6c 62 7c ff 7e 61
07b5 7f fc e1 fe e2 ec fb a0
07c3 20 6c 7b 62 7c e1 7e 7f
07cd ff fe e2 fb 61 fc ec a0
07d3 20 7e 7b 61 6c 7f 7c e2
07db ff ec 62 fc e1 fb fe a0
07e3 20 7c 7b ff 6c e1 7e e2
07eb 7f fb 62 fe 61 ec fc a0
07f3 00 ff 00 ff 00 ff 00 ff
07fb 00 ff 50 0a 01 aa 09 00

```


Fehler	Ursache
Senkrechter Strich in der unteren Ecke des Bildschirms	U19 (74S74) ist defekt (U19 darf kein LS-Typ sein)
IEEE-Bus wird nicht angesprochen	U1 (6526) ist defekt oder die Kontakte der Steckleiste sind oxidiert
Einige Zeichen der Tastatur gehen nicht	Kurzschluß des Netzleiles mit dem Stecker der Tastatur oder eine der Dioden CR4-CR19 ist defekt
Die Tastatur nimmt keine Zeichen an	U28 (G525) ist defekt oder der Tastaturanschluß steckt nicht richtig auf der Platine
RS-232 Schnittstelle geht nicht	U15 (G551) ist defekt oder die +12 V Versorgungsspannung fehlt
Kein Ton aus dem Lautsprecher	Anschluß des Lautsprecher ist nicht aufgesteckt
Kein Ton aus Lautsprecher und Audio-Buchse	U10 (G581) ist defekt
Kein Bild oder verzerrtes Bild	U7 (G545/G645) oder U28 (G525) sind defekt
Cursor erscheint aber keine Systemmeldung	Ein EPROM (U59 U60 U61) defekt oder nicht richtig eingesteckt
Rechner springt nach dem Einschalten in das Monitor-Unterprogramm	Ein EPROM (U59 oder U60) ist defekt oder nicht richtig ein- gesteckt
Leuchtdiode leuchtet bei Betrieb nicht	Anschluß der der Leuchtdiode ist nicht oder verkehrt herum aufgesteckt
Rechner und Leuchtdiode bleiben aus	Sicherung oder Netzteil defekt (Kurzschluß auf der Platine) U10 (G509) defekt
Steckmodul springt nicht an	Kontakte des Steckmodules sind oxidiert
Reset-Taste funktioniert nicht	Anschluß der Reset-Taste ist verkehrt aufgesteckt

Alle Angaben sind ohne Gewähr

Die 256k-Erweiterung ist auf der Platine bereits vorgesehen. Sie benötigen:

- Unbedingt einen potentialfreien Lötkolben (Lötstation) und eine hinreichende Lötterfahrung!
- 16 IC-Sockel 16-polig,
- 16 Dyn. Ram's 4164, 150 ns,
- 16 WIMA Kondensatoren 0.22 microfarad, 63V,
- 2 Elko's 2.2 microfarad, 10V, (sind manchmal schon vorhanden!)
- eine Entlötpumpe oder einen 0.8mm Bohrer zum Freimachen der Lötunkte.
- eine ruhige Hand und ca. 2 Stunden Zeit.

Zuerst öffnen Sie Ihr Gerät, 5 Schrauben im unteren Deckel, Garantie kann nicht erlöschen, weil's keine gibt. Klappen Sie vorsichtig den oberen Deckel auf und stellen ihn ab. Dann alle Kabelstecker von der Platine abziehen und die Platinschrauben lösen. Diese kann jetzt herausgenommen werden.

Legen Sie die Platine so vor sich hin, daß Sie den Bestückungsaufdruck lesen können. Im rechten Teil erkennen Sie 16 unbesückte IC-Plätze (U29-U36, bzw. U45-U52) samt den zugehörigen Entkopplungskondensatoren C66-C73 und C94-C101.

Die beiden Elko's gehören in C65 und C93.

An allen genannten Punkten müssen Sie die Lötunkte entlöten oder mit 0,8mm durchbohren. Achten Sie auf durch Lötzinn verursachte Brücken!

Haben Sie alle Bohrungen hergestellt, setzen Sie die IC-Sockel ein und verlöten diese sorgfältig. Danach setzen Sie die WIMA für jedes Ram ein, ebenfalls sorgfältig einlöten. Als letztes setzen Sie die beiden Elkos ein, achten Sie auf korrekte Polung, sie steht auf Elko und Platine. Sehen Sie sich Ihr Werk noch einmal selbstkritisch an, wenn Sie alles richtig gemacht haben, können Sie nunmehr die Ram's in die Sockel setzen, die Kerbe im IC muß übereinstimmen mit der Kerbe auf der Platinezeichnung!

Bauen Sie jetzt die Platine wieder ein, schließen alle Leitungen an, setzen den Deckel auf und dann wird's spannend:

Sie schalten ein. Sehen Sie den Cursor nicht gleich, schlecht, wieder ausschalten, irgendwo ist ein Fehler. Sehen Sie den Cursor, sollte sich der Rechner nach spätestens 8 sek. melden mit der üblichen 128-Einschaltmeldung. Nun können Sie sich schon beinahe freuen, vorher folgender Test:

Mit sys 60950 in den Monitor, v03 eingeben, m0000ffff eingeben, jetzt muß ein abwechselndes Muster von 00 und ff erscheinen. Den gleichen Test mit v04 machen. Ist alles zur Zufriedenheit, können Sie den Rechner endgültig zuschrauben.

Wollen Sie 256k unter Basic benutzen, so benötigen Sie allerdings noch zwei geänderte ROM's, da zwar das Betriebssystem den erweiterten Speicher kennt, Basic jedoch nicht. Benutzen können die neuen Segmente 03 und 04 nur aus dem Monitor. Wollen Sie ihn auch für Basic, dann stehlen Sie einem G20-Besitzer die IC's "basic-low 256k" und "basic-high 256k" (U59, U60).

Oder schicken Sie mir die beiden alten EPROM's, ich kann Ihnen das neue Basic für DM 10,- schießen.

Den kompletten Umbau, einschließlich aller Teile, kann ich notfalls für DM 200,- vornehmen, bitte senden Sie mir keine aufgrund Eigenlötarbeiten defekt gewordene Geräte, der Fehlersuchaufwand steht hier in keiner Relation mehr zum Anschaffungspreis.

Ich wünsche Ihnen viel Erfolg!!

Editor Funktionen

ESC = chr\$(27)+ = sys =chr\$(

A	chr\$(65)	59939		automatic insert
B	chr\$(66)	59835	chr\$(143)	set bottom
C	chr\$(67)	59936		cancel automatic insert
D	chr\$(68)	58989	chr\$(21)	delete line
E	chr\$(69)	59887		nonflashing cursor
F	chr\$(70)	59678		flashing cursor
G	chr\$(71)	59862		enable bell
H	chr\$(72)	59864		disable bell
I	chr\$(73)	58968	chr\$(149)	insert line
J	chr\$(74)	58674		move to the start of the line
K	chr\$(75)	58692		move to the end of the line
L	chr\$(76)	59107		enable scrolling
M	chr\$(77)	59109		disable scrolling
N	chr\$(78)	59909		normal screen
O	chr\$(79)			cancel insert+quote+reverse
P	chr\$(80)	59049	chr\$(150)	erase to the start of the line
Q	chr\$(81)	59028	chr\$(22)	erase to the end of the line
R	chr\$(82)	59894		reverse screen
S	chr\$(83)	59834		solid cursor
T	chr\$(84)	59833	chr\$(15)	set the top of the page
U	chr\$(85)			out of use
V	chr\$(86)	59069	chr\$(25)	scroll up
W	chr\$(87)	59083	chr\$(153)	scroll down
X	chr\$(88)	59769		cancel escape sequence
Y	chr\$(89)	59912		normal character set
Z	chr\$(90)	59897		alternate character set
		58509	chr\$(7)	ring bell
		58554	chr\$(4)	cancel last number
		58161	chr\$(29)	cursor left
		58759	chr\$(157)	cursor right
		58147	chr\$(145)	cursor up
		58132	chr\$(17)	cursor down
		58600	chr\$(20)	delete character
		58798	chr\$(148)	insert character
		58191	chr\$(19)	cursor home
		58186	chr\$(147)	screen clear cursor home
		58182	chr\$(18)	reverse on
		58180	chr\$(146)	reverse off
		58202	chr\$(137)	(un)set tab
		59738	chr\$(9)	use tab
		57933	chr\$(142)	graphics
		57937	chr\$(14)	text

Nach dem Umbau eines CBM 6X0 auf den 7X0er-Modus stellen die meisten von uns leider fest, daß sich das Bild auf dem Monitor nicht mehr einstellen lässt. Es erscheint ein Gewirr von Zahlen oder Strichen oder bestenfalls sieht man einige undeutliche Bilder nebeneinander.

Ursache ist die höhere Horizontal-Ablenkfrequenz, die im 7X0-er Modus etwa 18,4 kHz beträgt.

Wenn man Glück hat, ist dem Übel durch verdrehen des Zeilenfrequenz-Einstellers beizukommen. Das geht aber nicht immer, und dann muß gelötet werden.

In Reihe mit dem Zeilenfrequenz-Einstellregler ist meistens ein Widerstand vorhanden. Diesen gilt es sinnvoll zu verkleinern. Nie ganz weglassen. Normalerweise wird der Widerstand für eine höhere Frequenz verkleinert und umgekehrt. Oft kann man auch Erfolg haben durch Veränderung des frequenzbestimmenden Kondensators (meistens in der Größenordnung von 1 - 10 nF), von dem von dem in der 625-Zeilen-Ausführung manchmal 2 Stück parallel geschaltet sind, und von denen man den mit dem kleinsten Wert einfach auslöten muß um auf die höhere Ablenkfrequenz zu kommen.

Als Beispiel kann einmal der ITC-Monitor DM 1212 (Völkner) herangeführt werden. Dieser (grüne) Monitor ist preiswert (z.Z. DM 109,-) und für unsere Zwecke hervorragend geeignet.

In der Grundkonzeption ist der Monitor schon für die höhere Zeilenfrequenz von 18,4 kHz vorgesehen, umgekehrt läuft er so mit dem normalen CBM 610 nicht.

Abhilfe beim DM 1212 schafft das vergrößern von R21 von 2,2 kΩ auf 3 kΩ. Aber denken Sie daran sorgfältig zu arbeiten, sonst kann es passieren, daß der Zeilenendstufen-Transistor seinen Geist aufgibt weil keine Ansteuerung mehr da ist. Das ist weiter nicht so schlimm, denn dieser Transistor (Q1) kann durch einen überall erhältlichen Transistor BU 407 ersetzt werden. Am besten wird dieser dann gleich auf einen grösseren Kühlkörper montiert.

Anmerkung: Bei einigen anderen Monitor-Modellen kann es passieren daß die oberen Zeilen sich nach dem Umbau etwas nach links oder rechts verziehen. Das wird dann höhere Wissenschaft, allgemein gültige Regeln lassen sich hier schlecht aufstellen. Am besten holt man sich dann einen guten Bekannten dazu, der ausserdem noch Fernsehtechniker ist. Vielleicht kann der an einem Integrier- bzw. Differenzierglied etwas ändern.

Speziell der Monitor DM 1212 braucht für seine H-Synchronisierung und für die Video-Endstufe invertierte Signale bei Betrieb an den CBM 600er Computern. Dazu müssen zwei Brücken auf der Hauptplatine umgelegt werden. Und zwar müssen die Anschlüsse 2 und 9 von U13 (610) an Masse gelegt werden. Vorher sind unbedingt Leiterbahnen aufzutrennen. Links vom Quarz Y1, oberhalb der 60-poligen Steckerleiste befindet sich eine Gruppe von 3 x 3 Lötungen, von denen im Originalzustand die oberen Punkte mit

den mittleren durchverbunden sind. Das linke und das mittlere dünne Bähnchen sind vorsichtig durchzutrennen und dann die mittleren Lötungen mit den unteren durch 2 Drahtbrücken zu verbinden. Damit sind dann das H-Sync-Signal und das Videosignal invertiert. Die Signalform des "Composite-Video bleibt unverändert erhalten.

Wenn Sie das alles hinter sich gebracht haben, dann müsste der Monitor DM 1212 nach anlegen von 15 Volt Gleichspannung spielen.

Man kann den Kontrast beim DM 1212 noch etwas hochkitzeln. Das geht zwar auf Kosten der Bandbreite, aber auf dem Bildschirm ist kaum eine Verschlechterung zu erkennen. Wem der Kontrast bei diesem Monitor also zu gering ist, der kann durch vergrößern des 820 Ω / 2 Watt Widerstandes (R 205) auf der Bildröhren-Anschlussplatte auf 1200 Ω den Kontrast erhöhen.

Übrigens, wenn es Ihnen nicht gefällt nur mit dem bereits eingebauten Kontrastregler (ist kein Helligkeitsregler) auszukommen, dann kann an den DM 1212-Anschlüssen 2,3 und 4 ein 100-k Ω -Poti (Schleifer an Anschluss 4) angeschlossen werden. Die Grundeinstellung nimmt man so vor, daß das externe Poti auf max. Helligkeit eingestellt wird und der Brightness-Grundeinsteller auf der Platine soweit verdreht wird, daß das Hintergrundraster noch gerade gut sichtbar ist. Mit den externen Potis die Sie dann für Kontrast und Helligkeit zur Verfügung haben, lässt sich das Bild dann angenehm einstellen.

Jeder von uns wird wissen, daß ein Trafo ein magnetisches Streufeld hat. Am kleinsten ist dieses Streufeld bei guten Ringkernen oder bei den sogenannten Philbert-Trafos. Wenn das Bild auf dem Monitor einen Bauchtanz vollführt, dann ist in aller Regel das Streufeld des falsch platzierten Trafos die Ursache. Da hilft nur ein streuärmerer Trafo, oder das Ding so zu verdrehen und weiter wegzusetzen, daß der Bauchtanz nicht mehr stört. Eine gut gesiebte 15 Volt Gleichspannung für den Monitor bleibt selbstverständlich Voraussetzung.

Noch etwas: Versuchen Sie beim Kauf eines Monitors immer ein Schaltbild zu ergattern. Das erleichtert bei späteren Reparaturen und Umbauten die Arbeit ungemein.

G250 umschaltbar nach G050-format.

Max Wanrooij

Das Gehäuse öffnen. Rechts am Forderseite ist eine L-förmige Platine. Darauf montiert eine G530 Chip.

Wenn Pin 18 nach aussen gebogen ist, und via eine Schalter geleitet Ja oder Nein mit Erde verbunden wird dann ist

Nein eine G050 und Ja eine G250.

Softwarelösung:

open15,8,15

print#15, 'M-W';chr\$(172);chr\$(16);chr\$(1);chr\$(1)

print#15, '09'

close15

Zurückstellung im G250-modus: das letzte chr\$(1) in zweite Zeile abändern in chr\$(2) und das Programm wiederholen.

Erläuterung:

Diese Memory-Map beschreibt die Benutzung der verschiedenen RAM-Speicherplätze. Die Darstellung der Speicherbenutzung durch das BASIC ist aus Jim Butterfield, "Machine-Language", 1984 entnommen. (Ein sehr gutes Buch, das zur einen Hälfte aus einem sehr schönen Assembler-Lehrgang besteht und zur anderen einen Anhang bietet, der über fast alle Commodore-Computer Systeminformationen bereithält, auch zum CBM 600/700). Da ich mich mit dem BASIC-Interpreter nicht besonders gut auskenne, sind die darüber gemachten Angaben nicht von mir überprüft. Die Beschreibungen der RAM-Belegung durch das KERNAL sollten aber vollständig und richtig sein. Über Ergänzungen und Berichtigungen, die vielleicht doch noch anzubringen sind, freuen wir uns.

Alle Bänke:

0000	0	6509 execution register, Bank der Programm-
		ausführung
0001	1	6509 indirection register, Bank der Datenzu-
		griffe

Bank 0

0000-ffff	0-65535	frei (nur bei 1 Megabyte-Ausbau verfügbar)
-----------	---------	--

Bank 1

0002-f000	2-61439	BASIC Programmtext
fa5e-fb00	61440-64512	Eingabepuffer in BASIC

Bank 2

CBM x20:		
0002-fbff	2-65535	BASIC Feldvariablen
CBM x10:		
0002-fbff	2-64511	BASIC Numerische, Feld- und String-Variablen
fc00-ffff	64512-65535	Funktionstexte

Bank 3 (nur CBM x20)

0002-7fff	2-32767	frei
8000-ffff	32768-65535	BASIC Numerische Variablen

Bank 4 (nur CBM x20)

0002-fbff	2-64511	BASIC Stringvariablen (top down)
fc00-ffff	64512-65535	Funktionstexte

Bank 5 bis Bank 14 frei, je nach Speicherausbau benutzbar

Bank 15 (Systembank)

0002-008f BASIC

02-04	2-4	USR Sprungadresse: x10: JMP \$9ba7
		x20: JMP \$9aed
05-08	5-8	IIS, Systemuhr (h, m, s, 1/10 s)

0009-000B	9-11	Print Using format pointer
000C	12	Search character
000D	13	Scan-between-quotes Flag
000E	14	Input point; number of subscripts
000F	15	Catalog line counter
0010	16	Default DIM flag
0011	17	Type: 255 = string, 0 = integer
0012	18	Type: 128 = integer, 0 = floating point
0013	19	Crunch flag
0014	20	Subscript index
0015	21	Input = 0; get = 64; read = 152.
0016-0019	22-25	Disk status work values
001A	26	Current I/O device for prompt suppress
001B-001C	27-28	Integer value
001D-001F	29-31	Descriptor stack pointers
0020-002B	32-43	Miscellaneous work pointers
002D-002E	45-46	Start-of-BASIC pointer
002F-0030	47-48	End-of-BASIC pointer
0031-0032	49-50	Start-of-Variables pointer
0033-0034	51-52	End-of-Variables pointer
0035-0036	53-54	Start-of-Arrays pointer
0037-0038	55-56	End-of-Arrays pointer
0039-003A	57-58	Variable work pointer
003B-003C	59-60	Bottom-of-Strings pointer
003D-003E	61-62	Utility string pointer
003F-0041	63-65	Top of string memory pointer
0042-0043	66-67	Current BASIC line number
0044-0045	68-69	Old BASIC line number
0046-0047	70-71	Old BASIC text pointer
0049-004A	73-74	Data line number
004B-004C	75-76	Data text pointer
004D-004E	77-78	Input pointer
004F-0050	79-80	Variable name
0051-0053	81-83	Variable address
0054-0056	84-86	For-loop pointer
0057-0058	87-88	Text pointer save
005A	90	Comparison symbol accumulator
005B-005D	91-92	Function location
005E-0060	94-96	Working string vector
0061-0063	97-99	Function jump code
0064-006E	100-110	Work pointers, values
006F	111	Exponent sign
0070	112	Acum string prefix
0071	113	Acum#1: exponent
0072-0075	114-117	Accum#1: mantissa
0076	118	Accum#1: sign
0077	119	Series evaluation constant pointer
0078	120	Accum#1 hi order (overflow)
0079-007E	121-126	Accum#2
007F	127	Sign comparison, Acc#1 versus #2
0080	128	Acc#1 low-order (rounding)
0081-0084	129-132	Series, work pointers
0085-0087	133-135	BASIC text pointer
0088-0089	136-137	Input pointer
008B-008E	139-142	DOS parser work values
008F	143	Error type number

0090-00ff KERNAL

90-92	144-146	Zeiger auf Filenamen
93-95	147-149	Zeiger für LOAD/SAVE
96-98	150-152	LOAD/SAVE, letzte Adresse (end address)
99-9b	153-155	LOAD/SAVE, erste Adresse (start address)
9c	156	I/O Status (SI), 00 : OK
		#00 : Fehler
9d	157	Länge des Filenamens
9e	158	Aktuelle Logische Filenummer
9f	159	Aktuelle Gerätenummer
a0	160	Aktuelle Sekundäradresse
a1	161	Eingabegerät, normal = 1 (Tastatur)
a2	162	Ausgabegerät, normal = 3 (Bildschirm)
a3-a5	163-165	Soll-Ladestart-Adresse
a6-a8	166-168	Zeiger auf Eingabebuffer RS 232
a9	169	Stoptastenflag nach IRQ: 3f: keine Taste
		3e: Stoptaste
		3d: / (Nummernblock)
		3b: -
		37: +
		2f: enter
aa	170	IEEE deferred flag
ab	171	IEEE deferred character
ac-ad	172-173	Zeiger bei Bankumschaltroutine
ae	174	Monitor: Register bei BRK: PC high
af	175	Monitor: Register bei BRK: PC low
b0	176	Monitor: Register bei BRK: SR
b1	177	Monitor: Register bei BRK: AKKU
b2	178	Monitor: Register bei BRK: XR
b3	179	Monitor: Register bei BRK: YR
b4	180	Monitor: Register bei BRK: SP
b5	181	Monitor: Register bei BRK: \$01 (Bankregister)
b6	182	frei
b7-b8	183-184	Monitor: IRQ bei BRK
b9-ba	185-186	Monitor: Zeiger 1
bb-bc	187-188	Monitor: Zeiger 2
bd	189	Monitor: Zähler
be	190	Monitor: Verschiedenes
bf	191	Monitor: Gerätenummer
c0-c1	192-193	Startadresse Funktionstastentext
c2-c3	194-195	Zeiger auf aktuelles Funktionstastenwort
c4-c5	196-197	Zeiger für Funktionstastenänderungen,
		Zeiger für Scrolling
c6-c7	198-199	Zeiger für Funktionstastenänderungen
c8-c9	200-201	Zeiger in Bildschirmspeicher
ca	202	Aktuelle Zeilennummer des Cursors
cb	203	Aktuelle Cursorposition in Zeile \$ca
cc	204	Text/Graphikflag, #00: Text, #00: Graphik
cd	205	Letzte Taste bei IRQ (für Tastaturrepeat)
		Keine Taste: \$ff
ce	206	Letzte Cursorzeile (vor Input)
cf	207	Letzte Cursorspalte (vor Input)
d0	208	New character flag (für franz. Zeichen)
d1	209	Anzahl der Tasten im Tastaturbuffer
d2	210	Gänsefußchen flag; 00= normal, \$01= *-Modus
d3	211	Anzahl characters während Gänsefußchenmodus

d4	212	Cursor-Flag: bit 6&5: 00: Blinken aus 01: Kein Cursor 10: Blinken mit 1/16 11: Blinken mit 1/32 Feldrate (normal)
d5	213	Länge der Input-Zeile
d6	214	Anzahl der 'Tasten' im Funktionstastenbuffer
d7	215	Zähler für Tastenwiederholung
d8	216	Zähler bis zur ersten Tastenwiederholung
d9-da	217-218	Verschiedene
db	219	Aktueller Ausgabecharacter (Print)
dc	220	Oberste aktuelle Zeile (Fenster)
dd	221	Unterste aktuelle Zeile (Fenster)
de	222	Aktuelle erste Spalte (Fenster)
df	223	Aktuelle letzte Spalte (Fenster)
eo	224	Flag bei Tastatur-IRQ: \$ff=keine Taste \$7f=Taste gedrückt \$6f=mit shift \$5f=mit control
e1	225	Zeiger auf Wert in Tastaturtabelle
e2-e5	226	Bitmap für doppelte Zeilen
e6	230	(Verschiedenes bei Bank 0X-Kernal)
e7-ff	231-255	frei
0100	256	Zwischenspeicher Hex nach Bin (Monitor)
0100-010a	256-266	Numeric to ASCII work area (Basic)
0100-01fe	256-510	Stack
01ff	511	Stackpointer bei Bankumschaltroutinen
0200-02ff BASIC		
0200-020f	512-527	Bereich für Filenamen
0210-0226	528-550	Disk command work area
0227-022f	551-559	frei
0230-0254	560-595	Kassettenbuffer (Super-Tape)
0255-0256	597-598	Verschiedenes
0257	599	Bankadresse Zwischenspeicher
0258	600	Output logical file
0259	601	Sign of TAN
025a-025d	602-605	Pickup subroutine; Verschiedenes
025e-0276	606-630	PRINT USING working variables
0277-027f	631-639	frei
0280-0281	640-641	Vektor: Error routine (8555; x10) (854d; x20)
0282-0283	642-643	Vektor: Warm start (85cd) (85c5)
0284-0285	644-645	Vektor: Crunch token (88c2) (88a9)
0286-0287	646-647	Vektor: List (89f4) (89db)
0288-0289	648-649	Vektor: Command dispatch (8754) (874c)
028a-028b	650-651	Vektor: Token evaluate (96b1) (969c)
028c-028d	652-653	Vektor: Expression eval (95c4) (95af)
028e-028f	654-655	Vektor: CHRGOT (ba2c) (b98e)
0290-0291	656-657	Vektor: CHRGET (ba32) (b994)
0292-0293	658-659	Vektor: Float - fixed (ba1e) (b980)
0294-0295	660-661	Vektor: Fixed - float (9d39) (9ca5)
0296-0297	662-663	Vektor: Error trap
0298-0299	664-665	Error line number
029a-029b	666-667	Error exit pointer
029c-029d	668-669	Stack pointer save
029e-029f	670-671	Temporary TRAP, DISPOSE bytes
02a0-02a5	672-677	Temporary INSTR\$ bytes
02a6-02a7	678-679	Bank offset
02a8-02ff	680-767	frei

0300-03ff KERNAL

0300-0301	768-769	IRQ Vektor (fbc9)
0302-0303	770-771	BRK Vektor (ee21)
0304-0305	772-773	NMI Vektor (fcaa) (RT1)
0306-0307	774-775	OPEN Vektor (f6bf)
0308-0309	776-777	CLOSE Vektor (f5ed)
030a-030b	778-779	Connect-input Vektor (f549)
030c-030d	780-781	Connect-output Vektor (f5a3)
030e-030f	782-783	Restore default I/O Vektor (f6a6)
0310-0311	784-785	Input Vektor (f49c)
0312-0313	786-787	Output Vektor (f4ee)
0314-0315	788-789	Stoptastenabfrage Vektor (f96b)
0316-0317	790-791	GET Vektor (f43d)
0318-0319	792-793	Abort all files Vektor (f67f)
031a-031b	794-795	Load Vektor (f746)
031c-031d	796-797	Save Vektor (f84c)
031e-031f	798-799	Monitor Befehlsvektor (ee77)
0320-0321	800-801	User escape (e01f)
0322-0323	802-803	User control (e01f) nicht benutzt!
0324-0325	804-805	IEEE send LSA Vektor (f274)
0326-0327	806-807	IEEE send ISA Vektor (f280)
0328-0329	808-809	IEEE receive byte Vektor (f30a)
032a-032b	810-811	IEEE send character Vektor (f297)
032c-032d	812-813	IEEE send untalk Vektor (f2ab)
032e-032f	814-815	IEEE send unlisten Vektor (f2af)
0330-0331	816-817	IEEE send listen Vektor (f234)
0332-0333	818-819	IEEE send talk Vektor (f230)
0334-033d	820-829	File logical addresses table
033e-0347	830-839	File device table
0348-0351	840-849	File secondary addresses table
0352-0354	850-852	Bottom of system memory
0355-0357	853-855	Top of system memory
0358-035a	856-858	Bottom of user memory
035b-035d	859-861	Top of user memory
035e	862	IEEE timeout; 00= enable
035f	863	Load-verify flag; \$00= load, \$80= verify
0360	864	Anzahl offener Files
0361	865	Textausgabeflag; bit 7 =1, Meldungen ausgeben bit 6 =1, Fehlermeldungen ausgeben
0362	866	frei
0363-0366	867-870	Verschiedenes bei RS 232 & abort all files
0367-0368	871-872	frei
0369	873	Timer toggle, CIA-IRQ-Register nach IRQ
036a-036b	874-875	Kassetten-Vektor (fe5d, Fehlermeldung) (bc5d, Super-Tape)
036c-036e	876-878	frei
036f-0371	879-881	Sollstartadresse bei LOAD, \$xxxxfff: keine Verschiebung
0372-0374	882-884	frei
0375	885	Kassettenmotor-flag; 00= läuft
0376-0377	886-887	RS 232 control, command
0378-0379	888-889	frei
037a	890	RS 232 status
037b	891	RS 232 handshake input
037c	892	RS 232 input pointer

037d	893	RS 232 arrival pointer
037e-037f	894-895	frei
0380-0382	896-898	Top of memory pointer
0383-0396	899-918	Längen der Funktionstastenwörter
0397	919	Reverse flag; 00= normal, \$80= reverse
0398	920	Länge der aktuellen Zeile
0399	921	Zwischenspeicher für auszugebenden Character
039a	922	Autoinsertflag; 00= normal, \$80= autoinsert
039b	923	Scrollingflag; 00= scrolling ein, \$80= aus
039c	924	Verschiedenes für Bildschirmberechnungen
039d	925	Zähler für Funktionstastenwortausgabe
039e	926	Scrollingflag; aktuelle Zeile scrollen (=80) nicht scrollen (=00)
039f	927	Bell mode flag; 00: an, #00: aus
03a0	928	Zwischenspeicher für Bankadresse
03a1-03aa	929-938	Bitmap der Tabulatormarken
03ab-03b4	939-948	Tastatureingabebuffer
03b5-03b6	949-950	Funktionstastenvektor (e92a ASCII) (e957 DIN)
03b7-03f7	951-1015	frei
03f8-03f9	1016-1017	Warmstart-reset-Vektor (bbcc x10 Basic) (bb39 x20 Basic) (ee00 Monitor)
03fa-03fb	1018-1019	Warmstart-Testmaske (5aa5)
03fc-03fd	1020-1023	frei
0400-07ff	1024-2047	frei, aber meistens belegt
0800-0fff	2048-4095	Zusätzlicher RAM (Coprozessorkarte)
1000-1fff	4096-8192	Zusätzlicher RAM oder ROM (von Commodore für Floppybetriebssystem geplant)
2000-7fff	8192-23767	RAM oder EPROM im Cartridge-Connector
8000-bfff	32786-49151	BASIC-EPROMs
c000-cfff	49152-53247	Im Normalsystem nicht verfügbar, ev. irgendwann nachladbarer Zeichensatz
d000-d7ff	53248-55295	Bildschirmspeicher
d800-d801	55296-55297	Videocontroller (CRTC 6545)
d900-d9ff	55552-55807	Beliebiger I/O (Expansion-Connector)
da00-da1c	55808-58836	Sound interface device (SID 6581)
db00-db0f	56064-56079	Complex interface adapter (CIA 6526), von Commodore für Coprozessor vorgesehen), Chip-select an Expansion-Connector)
dc00-dc0f	56320-56335	Complex interface adapter (CIA 6526), IEEE-Daten, Kassetten-Read, Userport
dd00-dd03	56576-56579	Asynchronous communication interface adapter (ACIA 6551), RS 232
de00-de07	56832-56839	Tri port interface adapter (TPIA 6525), IRQ-Sammler, IEEE-Steuersignale, Kassetten- Steuersignale, Userport
df00-df07	57088-57095	Tri port interface adapter (TPIA 6525), Tastatur, Bildschirmmodi (PAL-NTSC-CRTC)
e000-ffff	57344-65535	KERNAL EPROM

Nachdem nun allgemein bekannt ist, daß sich die "Firmware" CBM610/710 von der der CBM620/720 lediglich durch den vom Basic benutzten Speicherplatz unterscheidet, und man zum Behufe des Umbaus X10 in X20 nach dem Ausbau des Speichers lediglich die beiden Basic-EPROMs auswechseln muß, sei darauf hingewiesen, daß ein Umbau eines 600ers in den Modus des 700ers einen durchaus erstrebenswerten Fortschritt darstellt. Der einzige Unterschied in der Funktionsweise beider Geräte liegt in der Bildschirmdarstellung. Hier bringt der 700er es auf eine Zeichenauflösung von immerhin 9*14 Punkten, was einer Gesamtauflösung von 720*350 Punkten entspricht.

Bei dieser wirklich beachtlichen Bildqualität sollte man übrigens im Falle des Umstiegs auf einen moderneren Computer die Weiterverwendung des 700 als Textterminal - gegebenenfalls mit weiteren Nebenfunktionen - ernsthaft in Erwägung ziehen.

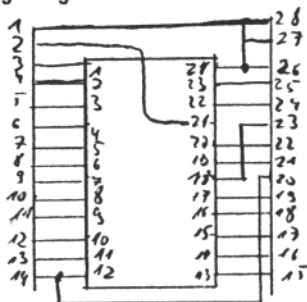
Zur Praxis

Der Umbau 6X0 ASCII in 7X0 ASCII erfordert neben einem guten Monitor lediglich ein neues Zeichengenerator-EPROM und die Umpolung eines Anschlusses am Tastaturportbaustein.

Der Zeichengenerator U25 ist zu entfernen und ein EPROM vom Typ 2732 einzusetzen. Die Leiterbahn zwischen Pin 33 und 34 von U28 wird durchtrennt und Pin 32 von U28 über einen Widerstand auf 5 V gelegt. Beim Einschalten fragt das ASCII-Kernal diesen Anschluß ab und nimmt nun die Videogenerator - Initialisierung des CBM 700 vor (siehe auch: Bildschirmmonitor - Tips in diesem Heft).

Die Umrüstung eines CBM 600 oder 700 von ASCII in DIN erfordert in jedem Fall ein neues Kernal. Hiervon gibt es gleich zwei Versionen, da wegen der komplizierteren Tastaturdekodierung (Akzente) kein Platz mehr vorhanden ist, um entsprechend dem ASCII - Kernal zwei verschiedene Videoformate darin unterzubringen. Der Aufwand einer DIN - Umrüstung erhöht sich weiter dadurch, daß für den Zeichengenerator ein 8 kbyte - EPROM benötigt wird wegen des unverzichtbaren zweiten Zeichensatzes. Es gibt zwar solche EPROMs, die mit dem Zeichengenerator - ROM pinkompatibel sind, meines Wissens aber nicht mit der erforderlichen kurzen Zugriffszeit, dafür aber sehr teuer. Es bleibt also nichts, als einen 2764 mit einem Zwischensockel zu verwenden. (Beschalung siehe Bild 1) Natürlich gibt es auch zwei verschiedene DIN Zeichengeneratoren (700 und 600).

Bei der Verwendung des genannten Adapters im CBM600 gibt es ein mechanisches Problem. Steckt man den Adapter in einen Sockel, drückt das Netzteil so stark darauf, daß er teilweise aus dem Sockel gehebelt wird. Lötet man den Adapter direkt ein, wird man ihn später vielleicht wieder entfernen wollen um einer in Entwicklung befindlichen preiswerten Grafik und oder oder um frei definierbarer Zeichensätze willen. Man kann aber auch Distanzhülsen in das Gehäuseoberteil kleben (ca. 5mm), wobei man noch, den Nebeneffekt einer besseren Wärmeableitung im ganzen Gerät erzielt.



DIN ASCII Universalkernal

Wie Sie sicher schon Ihrem Handbuch entnommen haben, ist im DIN - Kernal die Umschaltung auf einen zweiten Zeichensatz vorgesehen. Hier wird weitgehend die Tastaturbelegung nach DIN verwendet. Hauptsächlich die Abweichungen im Zeichensatz sind von dieser Umschaltung betroffen. Unerfreulich ist hier, daß im ASCII - Grafikmodus nach dieser Umschaltung völlig wirres Zeug auf dem Bildschirm erscheint. Dieser Umstand macht einen sehr großen Teil der verfügbaren Programme, die ja meist in ASCII geschrieben sind, unbrauchbar. Daher haben wir (Andreas Stankewitz und ich) ein doppeltes Kernal entwickelt, das sowohl ein der DIN - Norm entsprechendes als auch ein ASCII - Kernal enthält, ebenfalls mit normgerechter Tastaturdekodierung. Im Betrieb kann zwischen den Betriebssystemen vermittelt eines Schalters gewechselt werden, wenn man während des schaltens den Resetknopf gedrückt hält.

Es ist anzunehmen, daß ein beträchtlicher Teil unserer Mitbenutzer künftig vornehmlich mit der DIN - Tastatur arbeiten wollen wird. Daher wird Dirk Möll demnächst Aufkleber drucken lassen.

Wer bezüglich der Tastaturbelegung Sonderwünsche hat, möge sich vertrauensvoll an Wolfgang Taube, Trainau, Graitzer Gasse 1, 8627 Redwitz wenden. Er weiß, wie es geht, und hat einen fast viersseitigen Artikel zur Veröffentlichung eingereicht. Da dieses Gewußt wie von weniger allgemeinem Interesse sein dürfte, und wir der oben beschriebenen 2-Normlösung den Vorzug geben, haben wir vom Druck desselben abgesehen. Im übrigen sucht Wolfgang Taube noch Kontakt zu Leuten, die sich mit ihm um die Programmierung verschiedener Druckertreiber für die Zeichensätze des 8028 kümmern.

ANFRAGEN: Wir gehen förmlich unter in Anfragen. Die meisten derer, die irgendwelche Fragen an uns gerichtet haben warten noch bis heute auf eine Antwort. Dies hat nichts damit zu tun, daß wir keine Lust hätten die Fragen zu beantworten. Es ist vielmehr so, daß uns die Zeit fehlt. Aus diesem Grund suchen wir Ansprechpartner, die sich bereit erklären Anfragen aus einem bestimmten Sachgebiet zu beantworten. Es wäre schön, wenn sich möglichst viele hierzu melden würden. Vielleicht könnte so etwas auch in den Regionalgruppen organisiert werden, denn viele Fragen könnten dann mit einem preiswerten Telefongespräch geklärt werden. Das Spektrum der Gebiete und Anforderungen ist groß, es reicht von Fragen über das Basic bis zu Fragen zur Maschinenprogrammierung. Fragen, die von allgemeinem Interesse sind sollen im HSG veröffentlicht werden.

Folgende Gebiete sind uns eingefallen:

- * Basic 4.0
- * Emulation von 8032 Programmen in den 6xx/7xx
- * Hardware 6xx/7xx
- * Diskettenlaufwerke (SFD 1001, 8280, 1541/71 etc.)
- * Drucker (8028, 6400 etc.)
- * Betriebssystem, Assembler
- * etc.

Im Folgenden ist die Zeichenbelegung der Tastatur des von uns geänderten DIN-KERNALS beschrieben. Die Tastaturbelegung ist unabhängig vom CHAR-ROM. In Verbindung mit dem (originalen) ASCII-CHAR-ROM erscheinen daher bei Benutzung der Umlauttasten Graphikzeichen. Erst mit einem DIN-CHAR-ROM erscheinen die Umlaute auch auf dem Bildschirm (mit dem Club-EPR0M oder dem ROM von Commodore).

Die oberste Tastenreihe (Funktions-, Cursor- und Steuertasten) ist der Belegung identisch mit der ASCII-Version (entsprechend der Beschriftung der Tastenköpfe).

normal

esc	1	2	3	4	5	6	7	8	9	0	B	'	<	del	?	ce	*	/	
tab	q	w	e	r	t	z	u	i	o	p	ü	+	[retu]		7	8	9	-	
	shl	a	s	d	f	g	h	j	k	l	ö	#	[rn]		4	5	6	+	
	shift	y	x	c	v	b	n	m	,	.	-		shift	C=	1	2	3	[ent]	
	ctrl	-----space-----														0	.	00	[er]

mit shift

esc		~	S	\$	%	&	/	(	)	=	?	`	>	ins	£	µ	f	π
stb	Q	W	E	R	T	Z	U	I	O	P	Ü	*	[retu]		[	]	√	Σ
	shl	A	S	D	F	G	H	J	K	L	Ü	Ä	[rn]		1	τ	1	1
	shift	Y	X	C	V	B	N	M	;	:	_		shift	C=	[	]	f	[ent]
	ctrl	-----space-----															00	[er]

Die Codes aller Steuertasten werden durch Shift mit einem gesetzten Bit 7 versehen. Aus RETURN = \$0d wird \$8d, aus SPACE = \$20 wird \$a0 usw.

CTRL ergibt zusammen mit den Buchstabentasten (und nur diesen) direkt SteuerCodes. CTRL-m erzeugt z.B. \$0d = RETURN.

Mit Shift zusammen wird wieder das Bit 7 gesetzt.

CTRL-A ergibt also \$81, CTRL-B = \$82 usw.

Zur Tastaturbelegung von Commodore für DIN (die war leider nicht vollständig entsprechend der DIN-Norm für Dateneingabegeräte, typisch Commodore, so daß wir sie geändert haben) gehörten nicht die mathematischen Zeichen μ , π , $\sqrt{\quad}$ usw., obwohl sie im Zeichensatz-ROM vorhanden sind. Die 'letzten' Graphikzeichen des DIN-Zeichensatzes sind in den zuerst verkauften CHAR-EPR0Ms des Clubs noch ein paar Drucker Sonderzeichen wie ^ , ^ usw. zum Opfer gefallen. Diese Änderung hat sich aber leider nicht bewährt, da doch relativ viele Programme die Graphikzeichen benutzen (z.B. Protex).

Einleitung

In diesem Artikel will ich verschiedene Aspekte des Bankings im CBM 600/700 beschreiben. Der Anfang ist, so hoffe ich, auch für die Anfänger des Programmierens nachvollziehbar. Aus diesem Grunde sind die kurzen Assemblerbeispiele auch überausföhrlich dokumentiert, so daß sich eingefleischte Assemblerprogrammierer vielleicht an die Stirn tippen möchten, aber ich finde die Allgemeinverständlichkeit wichtiger.

Zum Ende hin wird der Artikel dann anspruchsvoller. Hier ist dann eine fundierte Kenntnis von Assemblerprogrammierung Voraussetzung, da die Bankumschalt Routinen durchaus nicht trivial sind.

In jedem Fall sind aber ein paar Grundkenntnisse in allgemeiner EDV von Nöten. Begriffe wie Adressbus, Prozessorregister, Programmcode usw. gehören mit dazu.

Grundlagen

Wie alle 8-Bit-Prozessoren ist der 6509 - ein Bruder des 6502 - bei Speichergrößen, die 64k-Byte übersteigen, erst einmal grundsätzlich überfordert. Das liegt an dem nur 16 Bit breiten Adressbus, dessen Breite wiederum von der internen Prozessorstruktur abhängt ($2^{16} = 65536 = 64k$).

Seitdem Speicher mit mehr als 64k-Byte Adressraum nicht mehr ausschließlich in das Ressort von Großrechnern gehören (noch gar nicht so lange her), sind viele Wege beschritten worden, den Prozessoren das Verwalten von mehr als sechzehn Adressleitungen beizubringen. Die Lösungsmöglichkeiten reichen und reichen von der umständlichen Erzeugung verschiedener Chipselectsignale durch I/O-Ports, welche für das Ansprechen bestimmter (Speicher-) Bausteine verantwortlich sind, bis zur Schaffung neuer Prozessoren. Der zur Zeit wohl populärste Prozessor mit mehr als 16 echten Adressleitungen ist der MC 68000, der in Computern wie Atari ST, Apple Macintosh und Commodore Amiga zu finden ist. In Großrechenanlagen sind Prozessoren mit mehr als 16 Adressleitungen allerdings schon länger gang und gäbe. Demgegenüber ist der 8086, der in den IBM PCs und seinen Nachbauten Anwendung findet, noch mit einer prozessorinternen Bankingstruktur behaftet.

Commodore ist mit seinem 6509 einen eigenen Weg gegangen, um aus dem 64k-Käfig auszubrechen.

Im Grunde ist der 6509 ein 6502-Prozessor, dem noch - on Chip - ein vier Bit breiter Ausgabeport beigelegt ist. Dieser Port wird nun auf zwei verschiedene Weisen angesprochen. Hierzu sind zwei Register vorgesehen. Die vier Leitungen werden im CBM 600/700 als Adressen A16 - A19 benutzt. Damit stehen insgesamt 20 Adressleitungen zur Verfügung. Die Steuerung der neuen Adressleitungen durch den Prozessor geschieht wie im folgenden beschrieben ist.

Wie schon erwähnt, sind zwei Register zur Ansteuerung des Adressports vorhanden. Sie werden mit den Adressen \$0000 und \$0001 angesprochen (diese Register sind in der Literatur häufig mit 'execution-register' und 'indirection-register' bezeichnet). Die Adressierung läuft prozessorintern ab, so daß ein extern angeschlossener Speicher bei diesen Adressen nicht benutzt wird.

Ein Schreiben in das 0-Register bewirkt eine sofortige Änderung der Pegel der Portleitungen, was wiederum eine Änderung der Adressleitungen A16 - A19 nach sich zieht. Im Klartext heißt das, daß die Programmausführung auf eine andere RAM-Bank umgeschaltet wird. Das bedeutet aber, daß in der dann adressierten Bank sinnvoller Programmcode stehen muß, sonst stürzt das System ab. Der Programcounter des Prozessors wird wie normal weiter hochgezählt. Ein kleines Programmbeispiel, hoffentlich auch für nicht Assemblerprogrammierer verständlich, mag dies verdeutlichen:

Programmstart in Bank 15 (\$0f)

```
0400 sei      Interrupts ausschalten, Erklärung s. Listing
0401 lda #S02 Den Akkumulator mit dem Wert S02 = 2 laden
0403 sta S00   In das 6509 Register zur Programmausführung
               schreiben
```

Jetzt wird auf die Bank S02 umgeschaltet

```
0405 xxx      Hier muß nun der nächste Programmcode stehen
```

Diese Art und Weise des Bankwechsels ist aber nur in den seltensten Fällen zu gebrauchen, da in beiden Banken an ungefähr der gleichen Stelle Programmcode stehen muß. Außerdem können so keine Unterroutrinen angesprungen werden, da diese mit RTS (return from subroutine) enden und nicht, wie es notwendig wäre, um zur Ausgangsbank zurückzukehren, mit der Befehlsfolge wie in unserem Beispiel.

Das viele Programme nur in Bank 15 laufen liegt daran, daß sich hier das Kernl und die I/O-Bausteine befinden, die von anderen Banken aus eben nur sehr Unständlich angesprochen werden können.

Eine etwas bequemere Art des Bankwechsels bieten die Bankumschaltroutinen des Kernals, die weiter unten beschrieben werden.

Nicht so schwierig wie das Wechseln der Programmausführung von einer Bank zur anderen ist die Adressierung von Daten über die Bänke hinweg. Hierfür ist das Register \$0001 vorgesehen. Wird ein Wert in dieses Register geschrieben, so wird dieser erst dann auf den Port gelegt, wenn die Prozessorbefehle LDA (\$xx),y oder STA (\$xx),y benutzt werden. Erst wird anhand der 0-Page-Adressen und des Y-Registers die gewünschte Speicheradresse berechnet. Wenn diese dann auf dem Adressbus gelegt wird, erscheint gleichzeitig an den Port-Adressleitungen die in Register \$0001 festgelegte Bankadresse.

Die Praxis soll wieder an einem Beispiel verdeutlicht werden:


```

0400 lda #S45          in $f0-$f1 die Adresse $6745 schreiben
0402 sta $f0
0404 lda #S67
0406 sta $f1
0408 lda #S01          S01 in das Prozessorregister $0001
040a sta $01          schreiben
040c ldy #S12          Y- (Index-) Register mit $12 laden
040e lda ($f0),y       Akkumulator mit Wert von Adresse $6757
                        ($6745+$12) aus Bank $01 laden.
0410 sta $f2          in Adresse $00f2 der Programmbank
                        schreiben
0412 brk              Programm abbrechen

```

Mit dem Monitor läßt sich nun nachprüfen, daß in \$00f2, Programmbank (im einfachsten Fall Bank 15) der selbe Wert steht wie in Adresse \$6757, Bank \$01.

Durch diese Umlenkung ist es also relativ einfach möglich, Daten aus anderen Banken als der Programmbank zu lesen und dort welche hineinzuschreiben. Hiervon macht z.B. der Basicinterpreter, besonders die 256k-Version, viel Gebrauch. Die ganze 'Arbeit' läuft in Bank 15, der Systembank ab, während der Basic-Programmtext in Bank 1 steht und die Variablen in Bank 2 (bis Bank 4).

Die Bankumschaltroutinen des Kernals

Um die Ausführung von Unterprogrammen in anderen Banken als der Systembank (Bank 15) möglich zu machen, sind im Kernal die Bankumschaltroutinen vorgesehen.

Wenn man in Basic vor einem SYS-Aufruf den Bankbefehl mit einer Banknummer ungleich 15 verwendet, so werden diese Bankumschaltroutinen benutzt. Es kann also auch ein Assemblerunterprogramm in jeder beliebigen Bank von Basic aus benutzt werden.

Bei der Bankumschaltung werden alle Informationen, die für eine korrekte Bearbeitung des Prozessorbefehls RTS benötigt werden, in die entsprechende Bank übertragen. Das braucht allerdings seine Zeit, so daß mit der Benutzung der Umschaltroutinen die Programme nicht gerade beschleunigt werden.

Ein Unterprogramm in einer Bank x kann auch ein weiteres Unterprogramm in einer Bank y mittels der Bankumschaltroutinen aufrufen. Es darf nur nicht eine Bank angesprochen werden, die schon in der Kette von sich aufrufenden Unterprogrammen vorgekommen ist, da der Stack einer Bank mit einem Programmaufruf in diese neu initialisiert wird.

Also nicht:

```

Unterprogramm (UP) in Bank 1 aufrufen,
von dort aus UP in Bank 2 anspringen,
von dort aus UP in Bank 1 anspringen,
RTS in Bank 2,
RTS in Bank 1,
RTS in Startbank.

```

Die vierte Zeile (RTS in Bank 1) würde nicht mehr richtig ausgeführt.

Um mit den Bankumschalt Routinen arbeiten zu können, muß ein Teil dieser in die anzuspringende Bank kopiert werden. Wenn sie ganz kopiert werden, kann von der dann programm ausführenden Bank wiederum eine andere Bank angesprungen werden. Sie sind im Kernal von Adresse \$fe9c bis \$ff3d zu finden (siehe auch Listing am Ende des Artikels). Bevor die Bankumschalt routine des Kernals aufgerufen wird, müssen die Register des Prozessors mit der Startadresse des Unterprogramms geladen werden:

Akku = Bankadresse des UP's

XR = Lowbyte der Startadresse

YR = Highbyte der Startadresse

Die Bankumschalt routine wird mit JSR \$ff6c (Vektor in der Sprungleiste auf \$fe9c) angesprungen

Ein Unterprogramm aufruf in die Bank 1 mit der Programmadresse \$3456 geschieht also folgendermaßen:

```
0400 lda #01      Bankadresse
0402 ldx #56      Startadresse low
0404 ldy #34      Startadresse high
0406 jsr $ff6c    Bankumschalt routine aufrufen
```

UP in Bank 1 wird ausgeführt

```
0409 xxx          Hauptprogrammfortsetzung
```

Die Unterprogramme in den Banken ungleich 15 können so zwar auch auf das Kernal zugreifen, aber eben nicht mit dem gewohnten JSR über einen der Kernal-Vektoren in der Sprungleiste am Speicherende. Das hat auch zur Folge, daß keine Register übergeben werden können. Um dies zu erreichen sind erweiterte Umschaltroutinen von Nöten, die jetzt beschrieben werden sollen.

Kernal-Ersatzroutinen

Bei den Kernal-Ersatzroutinen zeigen die Vektoren der Kernal-Sprungleiste mittels JSR-Befehlen auf spezielle Umschaltroutinen. Anhand der Rücksprungadresse wird die Zieladresse im Kernal in der Bank 15 ermittelt. Dann wird auf die entsprechende Routine über die normalen Bankumschaltroutinen verzweigt. Die Register bleiben bei dieser ganzen Prozedur erhalten und werden an die Kernalroutinen übergeben. Auch nach einem RTS werden die Register nicht geändert.

Mit diesem Verfahren ist es also möglich, alle Routinen des Kernals, die über die Sprungleiste erreichbar sind, zu benutzen. Die Sache hat nur einen gewaltigen Haken: Die Bankumschaltroutinen sind sehr, sehr langsam und bremsen entsprechend die Kernalzugriffe ab.

Die Verringerung der Geschwindigkeit ist so groß, daß ein Maschinesprache-Monitor nicht mehr schneller läuft als sein Gegenstück in Basic!

Der Vorteil der Kernal-Ersatzroutinen ist aber unbestritten, der, daß endlich genug Platz zum Programmieren vorhanden ist, da man nicht mehr an die Systembank gebunden ist.

Um das Dilemma mit der zum Einschlafen reizenden Geschwindigkeitsverringerung zu umgehen gibt es nur eine Möglichkeit: Es muß ein neues Kernal geschaffen werden, das von einer beliebigen Bank aus die I/O-Zugriffe erledigt. Von

diesem neuen Kernal gibt es zwei Ausführungen. Die eine ist ein 8032-kompatibles Kernal, im 8432-Programm zu finden. Die andere ist bei mir in Entwicklung und fast fertig. Dieses Kernal ist ca. 99% kompatibel zum CBM 600/700 ASCII-Kernal und nur um 15/16 langsamer. Eine DIN-Version wird folgen (siehe auch Beilage).

Es soll aber trotzdem eine Kernal-Ersatzroutine vorgestellt werden. Dies mag der Erweiterung des allgemeinen Know-Hows dienlich sein, zumal solche Routinen doch in einigen Programmen benutzt werden (z.B. Makro-Basic).

Der Vollständigkeit halber möchte ich hier noch das Programm BRAKE-SYS erwähnen. Es dient ebenfalls dazu, die Kernalaroutinen in der Bank 15 anzuspringen. Allerdings wird hierzu nicht der JSR \$....-Befehl benutzt, sondern eine Folge von BRK und der Adresse der anzuspringenden Routine. Dieses Programm hat den Nachteil, daß man nicht mehr mit einem normalen Assembler arbeiten kann, da ein solcher nicht über einen 'BRK \$....'-Befehl verfügt. Aber immerhin eine originelle Idee, die Bankumschaltung so durchzuführen.

Ich beschreibe hier eine Erweiterung der von Commodore vorgeschlagenen Routine.

Erweitert ist die Routine in zwei Punkten:

1. Das Datenrichtungsregister \$0001 ist nach einem Kernalaufwurf unverändert,
2. Der Aufruf von Unterprogrammen in anderen Banken ist möglich.

Da das Listing ausführlich dokumentiert ist, will ich jetzt keine weiteren Worte darüber verlieren.

Ersatzkernel-Routine

			Hier zeigen alle JSRs der Sprungleiste hin
fe49	08	php	Register retten
fe4a	78	sei	
fe4b	48	pha	
fe4c	8a	txa	
fe4d	48	pha	
fe4e	98	tya	
fe4f	48	pha	
fe50	ba	lsx	Platz in Stack reservieren
fe51	ca	dex	
fe52	ca	dex	
fe53	ca	dex	
fe54	9a	txs	
fe55	a0 06	ldy #S06	Stackinhalt drei Bytes nach unten verschieben
fe57	bd 04 01	lda S0104,x	
fe5a	9d 01 01	sta S0101,x	
fe5d	e8	inx	
fe5e	88	dey	
fe5f	d0 f6	bnc \$fe57	
fe61	a9 76	lda #S76	RTS-Adresse zur Kernelroutine in Stack ablegen (\$fe77 -1)
fe63	9d 01 01	sta S0101,x	
fe66	a9 fe	lda #Sfe	
fe68	9d 02 01	sta S0102,x	
fe6b	a5 01	lda S00	aktuelle Bank
fe6d	9d 03 01	sta S0103,x	
fe70	a9 0f	lda #S0f	Zielbank = S0f (15) = Kernel
fe72	85 01	sta S01	
fe74	4c b9 fe	jmp \$feb9	Bankumschaltung durchführen
			nach RTS aus Bank 15 (Kernel) altes Datenrichtungsregister aus Stack holen (liegt unter RTS-Adresse)
fe77	08	php	
fe78	85 ac	sta Sac	
fe7a	68	pla	
fe7b	85 ad	sta Sad	
fe7d	68	pla	
fe7e	85 01	sta S01	
fe80	a5 ad	lda Sad	
fe82	48	pha	
fe83	a5 ac	lda Sac	
fe85	28	plp	
fe86	60	rts	zum Hauptprogramm
			IRQ/BRK Einsprung Akku retten
fe87	85 ac	sta Sac	Statusregister holen
fe89	68	pla	
fe8a	48	pha	
fe8b	85 ad	sta Sad	merken
fe8d	a5 01	lda S01	Datenrichtungsregister retten
fe8f	48	pha	
fe90	a5 ad	lda Sad	Statusregister
fe92	20 5e ff	jsr \$fff5e	IRQ oder BRK in Bank 15 ausführen
fe95	85 ac	sta Sac	Akku merken
fe97	68	pla	altes Datenrichtungsregister zurückholen
fe98	85 01	sta S01	
fe9a	a5 ac	lda Sac	Akku
fe9c	40	rti	

fe9d 85 01	sta \$01	Kernal Bankumschalt routine Bankadresse in Datenrichtungsregister UP-Startadresse +2 in Stack
fe9f 8a	txa	
fca0 18	clc	
feal 69 02	adc #S02	
fea3 90 01	bcc \$fea6	
fea5 c8	iny	
fca6 aa	tax	
fca7 98	tya	
fea8 48	pha	
fca9 8a	txa	
feaa 48	pha	
feab 20 19 ff	jsr \$fff19	Sac+ = \$0100; YR = \$ff
feac a9 fe	lda #Sfc	Stackstart in Bank X init.
feb0 91 ac	sta (\$ac),y	
feb2 08	php	zu übergebende Register in eigenen Stack
feb3 78	sei	
feb4 48	pha	
feb5 8a	txa	
feb6 48	pha	
feb7 98	tya	
feb8 48	pha	
.....		
		Einsprung aus Ersatzkernal- routine
feb9 20 19 ff	jsr \$fff19	Stack in Bank x initialisieren LDA \$0100,y (YR=\$ff) aus Bank X Stackstart in Bank X
febc a8	tay	YR = 'Stackpointer' für Bank X
febd a5 00	lda \$00	alte Bankadresse
febfb 20 2a ff	jsr \$fff2a	STA (\$ac),y : DEY
fcc2 a9 04	lda #S04	
fcc4 a2 ff	ldx #Sff	STX (\$ac),y : DEY :
fcc6 20 24 ff	jsr \$fff24	STA (\$ac),y : DEY
fcc9 ba	tsx	
fccab bd 05 01	lda \$0105,x	UP Startadresse -3
fecd 38	sec	
fece c9 03	sbc #S03	
fed0 48	pha	
fed1 bd 06 01	lda \$0106,x	
fed4 e9 00	sbc #S00	
fed6 aa	tax	
fed7 68	pla	STX (\$ac),y : DEY :
fed8 20 24 ff	jsr \$fff24	STA (\$ac),y : DEY
fedb 98	tya	
fedc 38	sec	
fedd e9 04	sbc #S04	
fedf 8d ff 01	sta \$01ff	'Stackpointer' merken
fee2 a8	tay	
fee3 a2 04	ldx #S04	Stack von Programmbank in Bank X schaufeln
fee5 68	pla	
fee6 c8	iny	
fee7 91 ac	sta (\$ac),y	
fee9 ca	dex	
feeca d0 f9	bne \$fee5	
feec ac ff 01	ldy \$01ff	Bank X 'Stackpointer'
feef a9 2d	lda #S2d	RTS-Adresse für RTS nach Bankwechsel in
fef1 a2 ff	ldx #Sff	Bank X - Stack schreiben
fef3 20 24 ff	jsr \$fff24	UP-Startadresse löschen
fef6 68	pla	
fef7 68	pla	

fef8	ba		tsx		Stackpointer für RIS nach
fef9	8e	ff 01	stx	\$01ff	"Ausflug" in Bank X merken
fefc	98		lya		Stackpointer =
fefd	aa		tax		Bank X - 'Stackpointer'
fefe	9a		txs		
feff	a5	01	lda	\$01	Bank X
ff01	4c	f6 ff	jmp	\$ffff	STA programmregister : RIS
					RTS nach UP in Bank X springt
					hierhin
ff04	ca		nop		RTI Einsprung
ff05	08		php		RIS Einsprung
ff06	08		php		Register retten
ff07	78		sei		
ff08	48		pha		
ff09	8a		txa		
ff0a	48		pha		
ff0b	98		tya		
ff0c	48		pha		
ff0d	ba		tsx		
ff0e	bd	06 01	lda	\$0106,x	alte Bankadresse Hauptprogramm
ff11	85	01	sta	\$01	Datenrichtungsregister
ff13	20	19 ff	jsr	\$ff19	\$ac+ = \$0100 : YR = \$ff
ff16	4c	dc fe	jmp	\$fedc	Stack zurückschaufeln : RTS-HP
ff19	a0	01	ldy	#\$01	\$ac = \$0100 : YR = \$ff
ff1b	84	ad	sty	\$ad	
ff1d	88		dey		
ff1e	84	ac	sty	\$ac	
ff20	88		dey		
ff21	b1	ac	lda	(\$ac),y	
ff23	60		rts		
ff24	48		pha		STX (\$ac),y : DEY :
ff25	8a		txa		STA (\$ac),y : DEY
ff26	91	ac	sta	(\$ac),y	
ff28	88		dey		
ff29	68		pla		
ff2a	91	ac	sta	(\$ac),y	
ff2c	88		dey		
ff2d	60		rts		
ff2e	68		pla		Register holen nach Bankum-
ff2f	a8		tay		schaltung, RTS zum UP hin
ff30	68		pla		ausführen
ff31	aa		tax		
ff32	68		pla		
ff33	28		plp		
ff34	60		rts		
ff35	20	49 fe	jsr	\$fe49	Vektor für NMI-ausführung
ff38	ea		nop		
ff39	20	49 fe	jsr	\$fe49	Vektor für BRK-ausführung
ff3c	20	49 fe	jsr	\$fe49	Vektor für IRQ-ausführung
ff3f	48		pha		NMI Einsprung
ff40	a5	ac	lda	\$ac	alle RAM-Adressen, die bei
ff42	48		pha		NMI-Abarbeitung verändert
ff43	a5	ad	lda	\$ad	werden retten
ff45	48		pha		
ff46	ad	ff 01	lda	\$01ff	
ff49	48		pha		

ff4a	a5	01	lda	\$01	Datenrichtungsregister
ff4c	48		pha		
ff4d	20	35 ff	jsr	\$ff35	JSR \$fe49, Bankumschaltung
ff50	68		pla		RAM-Adressen zurückholen
ff51	85	01	sta	\$01	
ff53	68		pla		
ff54	8d	ff 01	sta	\$01ff	
ff57	68		pla		
ff58	85	ad	sta	\$ad	
ff5a	68		pla		
ff5b	85	ac	sta	\$ac	
ff5d	40		rti		

ff5e	29	10	and	#\$10	IRQ oder BRK ?
ff60	d0	05	bne	\$ff67	
ff62	a5	ac	lda	\$ac	Akku
ff64	4c	3c ff	jmp	\$ff3c	JSR \$fe49, Bankumschaltung
ff67	a5	ac	lda	\$ac	Akku
ff69	4c	39 ff	jmp	\$ff39	JSR \$fe49, Bankumschaltung

					Kernal Sprungleiste
ff6c	4c	9d fe	jmp	\$fe9d	Vektor auf Bankumschaltroutine
ff6f	20	49 fe	jsr	\$fe49	
ff72	20	49 fe	jsr	\$fe49	
.	.	.	.	.	
.	.	.	.	.	
fff0	20	49 fe	jsr	\$fe49	
fff3	20	49 fe	jsr	\$fe49	
fff6	85	00	sta	\$00	in Programmregister schreiben
fff8	60		rts		zur Bankumschaltroutine
fff9	00		brk		nicht benutzt
fffa	3f	ff	nmi	\$ff3f	NMI-Bankumschaltroutine
fffc	00	00	reset		nicht benutzt, da reset immer
					in Bank 15 ausgeführt wird
fffe	87	fe	irq/brk	\$fe87	IRQ-Bankumschaltroutine

Krimskram

Der nicht dokumentierte "delete"-Befehl mit der selben Syntax wie "list" hat lediglich die Macke, daß er ohne "ready" abschließt. Für Neulinge: Hiermit kann man Basiczellen löschen.

Basiccompile älterer CBM - Rechner laufen oftmals auch auf dem 600/700 unter 8432.

Bei "input" ist ein Programmabbruch möglich durch die Eingabe von "run/stop return".

Die SFD 1001 ist eine Art "verkrüppelte" 8250. Der auf der Zwischenplatine montierte 6530 enthält das 8250 - Betriebssystem, das ausgeblendet ist. Das EPROM auf der selben Platine enthält eine minimal abgewandelte Version desselben. Viele Programme lassen sich wegen des Fehlens von Laufwerk 1 nicht mit der SFD benutzen. Ein ausgefuchster DOS-Kenner könnte eventuell die SFD so umprogrammieren, daß die beiden Diskettenseiten wie zwei Laufwerke angesprochen werden. Das könnte eine halbwegs brauchbare Notlösung für manche Probleme sein. Die Disketten würden nicht herumgedreht werden können wegen der dann falschen Drehrichtung.

Profitext ist nicht ganz sauber programmiert. Es springt Routinen im Betriebssystem direkt an statt über die dafür vorgesehenen Vektoren. Daher kann es nicht mit dem DIN - Kernal zusammenarbeiten. Ansonsten bezog sich meine "Kritik" in der Nullnummer vornehmlich auf die arg übersteigerten Erwartungen, die die Firma Völkner durch ihre Werbung erzeugt hat. Für diejenigen, die nur mal eben einen Text runterpinnen wollen und die 1541 benutzen, ist es wohl nicht verkehrt, da problemlos ohne langwierige Einarbeitung handhabbar. Eine etwas ausführlichere Besprechung ist angepeilt.

An der RAM - Floppy ist noch kein Byte geschrieben. Es geht nun mal nicht alles auf einmal. Mit Macrobasic ist aber schon jetzt völlig unproblematisch, den voll ausgebauten Speicher auszunutzen.

Für Mbyte-Besitzer bietet sich die Bank 0 für die Macrobasic-Erweiterung an, da diese sowohl vom Basic als auch vom Kernal weder bemerkt, geschweige denn benutzt wird. Dazu aber sind 2 Pokes notwendig: 1357,234 und 1358,234.

High- und Low- Profile sind andere CBM-Bezeichnungen für 600 und 700. Der 500 nennt sich auch PET II.

Beim CBM 700 fehlt der Masseanschluß am Userport.

Der Einzelblatteinzug für den 8028 kostet indiskutabel 1190,-- DM zzgl. MwSt.

Wer hat außer mir noch Interesse am 16 Bit Prozessor 65816 im 600/700, wenn das Ding mit 650 - Emulation mindestens 300,-- unter Freunden kostet? Dieser Prozessor wird außer im allseits hochgelobten, 4000,--DM billigen neuen Apple in einigen Selbstbaucomputern eingesetzt. Bei programmübiger Ausschöpfung seiner Fähigkeiten ist er bei 4 MHz schneller als ein 68000 mit 8 MHz.

Diese Rubrik soll eine feste Einrichtung werden und kostenlos zur Verfügung stehen. Es soll eine Ausnahme bleiben, daß angebotene Sachen wie die folgenden über mich verkauft werden. Wer wird Flohmarktedakteur?

Coprozessorplatine 8088 CBM - Entwicklungsversion mit CP/M 86, benötigt Doppellaufwerk, meistbietend.

Traktorantrieb mit einigem handwerklichem Geschick an CBM 6400 anzubringen. DM 170,--

Basic - Compiler DTL u.a. Macrobasic - kompatibel DM 180,--
kompilare laufen nur mit Dongle.

Calc Result Wohl eines der besten Tabellenkalkulationsprogramme DM 150,--

Profitext zu DM 85,-- ist zu haben von Jörg Bay, Hoben 8, 5232 Flammersfeld.

1541 - IEEE - Anschluß, beschrieben im G4er vom Januar 1986 n. Chr. sollte ursprünglich im Hardwareservice von Markt und Technik vertrieben werden. Karl Baur, Mittelstr. 30, 5100 Aachen ist bereit, Platinen anzufertigen. Eventuell kann durch Gemeinschaftseinkauf der relativ hohe Preis für alle Bauteile (ca. 60,--DM) niedrig gehalten werden.

Wer weiß eine Bezugsquelle für den Schaltplan der CBM High Speed Grafik (512*512 Punkte)?

Ralf Viehl hat eine Centronicsschnittstelle mit dem Userport realisiert.

Wer weiß eine Bezugsquelle für "Datavis", ein HRG-Programm mit allem drum und dran (Besprechung: Computer Persönlich 13/85)?

Wer kann, will und wird ein Servicehandbuch 8280 zur Verfügung stellen?

H.J. Zink, Kamperstr. 27, 5100 Aachen, hat einen 1541 - Anschluß für den seriellen CBM-Port entwickelt. Zwischen ihm und IEEE kann mit sys umgeschaltet werden. Erfahrungen meinerseits liegen noch nicht vor. Das Programm liegt wie die Viehlsche Cenronics-Routine und auch Supertape im Basic High EPROM. Man kann also vorläufig nur eins von all dem benutzen.

Materialbeschaffung

Wir haben uns entschlossen, nachdem sich die Suche nach irgendwelchen Ersatzteilen oder Zusätzen zu unseren Geräten oft als erfolglos, mindestens aber als langwierig herausgestellt hatte, für die wichtigsten notwendigen Teile Sammelbestellungen und einen EPROM - Programmierservice einzurichten.

Weiter vorne sind schon die verschiedenen EPROMs beschrieben worden. Ein kleiner Aufschlag zur Deckung der Gemeinkosten und als kleine Anerkennung für dumme Fleißarbeiten ist von allen Befragten gutgeheißen worden.

Die Nylonfarbbänder lassen in ihrer mechanischen Qualität zwar zu wünschen übrig, dafür sind sie aber sehr preiswert.

Die angebotene Experimentierplatine paßt in den Modulstecker und eignen sich gut zum Aufbau eigener Schaltungen. Nicht die Buspufferung vergessen!

Das angebotene Gehäuse paßt zu der RAM/EPROM - Platine oder zu einer verkürzten Experimentierplatine.

Die RAM/EPROM - Platine enthält keine Batteriepufferung. Eine Nachrüstung ist aber möglich. Wenn noch keine Anleitung beigelegt werden kann, erscheint eine im nächsten Heft.

Auf daß keine Mißverständnisse aufkommen mögen: Einen Versandhandel wollen wir nicht aufmachen. Die Idee, eine gemeinsame Materialbeschaffung zu organisieren, ist bei unseren kölnen Treffen entstanden zum Zweck, die Suche nach Bezugsquellen für den einzelnen zu verkürzen und durch Mengenrabatte oder auch -bei den Platinen- durch genau auf den Bedarf abgestimmte Produktion günstig an das Zubehör zu kommen.

Wie künftig die freie Software unter uns verteilt werden kann, muß noch geklärt werden. Hoffentlich braucht das nicht zentral geregelt zu werden. Gegen das schon mal in Erwägung gezogene Schneeballprinzip spricht aber auch einiges. Ein Programm zum Überspielen von Diskette auf Kassette und umgekehrt fehlt noch in der Sammlung. Bei Kassetten bitte auch, das Aufzeichnungsformat angeben, da es deren ja immerhin auch drei gibt.

Kommerzielle Bezugsquellen

Leider ist festzustellen, daß kommerzielle Angebote an Programmen und auch an Hardwarezubehör für unsere Rechner oft ebenso*orbitant teuer sind wie der Rechner selbst billig. - Das ist er ja auch noch nicht lange.- Nun, jeder kann seine Preise festsetzen wie er will, und es scheint, daß mancher es vorzieht, null Stück irgendwas zu 1000,- als 20 Stück zu 100,-DM zu verkaufen. Dennoch muß man manchmal froh sein, überhaupt etwas aufzutreiben. Die Nennung von Bezugsquellen bedeutet nicht unbedingt eine Kaufempfehlung. Es wird, wie schon an anderer Stelle in diesem Heft erwähnt, um Erfahrungsberichte nachgesucht.

Speeddos für 8250, 8050 und SFD: ein verbessertes Betriebssystem zu 450,-DM, das im kommerziellen Einsatz u.U. lohnt. Es ist erhältlich bei: Edotronic, St.- Veit-Str. 70, 8000 München 80.

Austrocomp Basiccompiler für 620/720, DM 785,- ca. 300,-

Business Basic 720 ?? DM 585,72 Beides ist erhältlich bei Digimat, Arbeitergasse 48, A 1050 Wien.

Protext; ausgezeichnetes Textsystem mit integriertem DFD - Programm. Wegen eines ausgeklügelten Kopierschutzes nicht im 8280 - Format und wegen seines Umfangs nicht im 1541 - Format. Bei der Verwendung von Einzellaufwerken müssen Einschränkungen in Kauf genommen werden wie bei fast allen wirklich professionellen Programmen. Meine Beiträge zu diesem Heft dienten auch meiner Einarbeitung in dieses Programm. Es kostet allerdings auch 298,-DM für den CBM 600/700 und nur 89,-DM für den C 128. Es ist erhältlich bei Rolf Velzke Hard & Software, Lindenstr. 12, 5210 Troisdorf. Dort ist neuerdings auch Microbasic zu haben.

Für diese Rubrik werden Hinweise jederzeit gerne angenommen, insbesondere solche auf Restpostenverkäufe.

Nichts geht mehr, wenn es so weitergeht! Wie schon erwähnt, habe ich seit einem halben Jahr zu nichts Anderem Zeit, als mich um Clubangelegenheiten zu kümmern. Daß dieses Heft das erste Ihnen allen zur Verfügung stehende Ergebnis dieser Arbeit seit einem halben Jahr ist, dürfte wohl als unhaltbarer Zustand allgemein anerkannt werden.

Daher noch einmal meine dringende Bitte, mich außer zur unten genannten Zeit nicht anzurufen. Ich habe wochenlang rund um die Uhr am Telefon gesessen. Mein Familienleben, meine Produktivität und mein Nervenkostüm haben darunter gelitten. Im Ergebnis habe ich mich dabei auch noch unbeliebt gemacht. Da ich nicht im Traum mit der Resonanz gerechnet hatte, die das Heft O von HSG?! gefunden hat, war ich in keiner Weise auf die daraus resultierenden Konsequenzen eingerichtet. So brauche ich zunächst viel Zeit, das hier entstandene Chaos zu beseitigen. Eigentlich möchte ich im Anschluß ganz gerne meinen Computer kennenlernen.

Für Telefongespräche stehe ich Donnerstags von 15³⁰ bis 21³⁰ zur Verfügung.

Des weiteren nimmt auch Andreas Stankewitz Anfragen entgegen und zwar Mittwochs von 17³⁰ bis 21³⁰ und Samstags von 15³⁰ bis 20³⁰. Anfragen, die sich auf "Sag' mir alles" reduzieren lassen, erübrigen sich.

Um Generalabsolution bitte ich dafür, daß ich so manche Zusage bisher nicht habe erfüllen können. Reklamationen in Postkartenform werden ab 16.3.dJ. bearbeitet.

Ab sofort nehmen Günter Brühl und ich Artikel für Heft 2 entgegen. Eine vorherlge Rücksprache erscheint sinnvoll. Mit Ihrer Hilfe könnte es ja vielleicht sogar möglich sein, das nächste Heft im April herauszubringen. Bitte aber keine endlosen Listings! Mehr an Programmen (gut dokumentiert, mit Lerneffekt) als in diesem Heft zu finden sind, wäre für meinen Geschmack zu viel.

Manuskripte für "HSG?!" mögen in Papierform und auf Diskette bzw. Kassette eingereicht werden. Das kann eine Menge Arbeit ersparen und dient einem einheitlichen Erscheinungsbild.

Impressum

Heute schon geust?! ist eine Art Zeitschrift der "CBM Selbsthilfe 600/700", die in loser Folge erscheint. Die Hoffnung auf eine zweimonatige Erscheinungsweise ist noch nicht aufgegeben.

Das Urheberrecht und die Verantwortung für den Inhalt liegen bei den Autoren.

Für die Richtigkeit irgendwelcher Angaben wird keine Gewähr übernommen. Schaltungen, Warenzeichen usf. werden ohne Rücksicht auf Schutzrechte dritter beschrieben.

Die Abgabe erfolgt gegen Erstattung der Kosten, die im voraus zu leisten ist. Für Heft 1 und einige folgende sind DM 20,-- zu entrichten.

Redaktion Software: G. Brühl, Steinerstr. 16, 7958 Laupheim

Redaktion Hardware, Schlußredaktion, Layout und Vertrieb: G. Schumacher, Palanterstr. 12 d, 5000 Köln 41

kostenlose Buchbindearbeiten: Jörg Linnig, Köln

Bankverbindung: Silvia Rothe, Sonderkonto; Postgiroamt Köln, Blz. 370 100 50; Kontonr. 3783 36-504

Vorankündigung
(Bei sofortiger Bestellung Sonderpreis, s.u.)

8028-Drucker aufgemotzt

Betriebssystemerweiterung

- 1) 'proportional spacing' d.h. beliebig große Leerzeichenbreiten. Dadurch ist es z.B. mit einem geeigneten Textverarbeitungsprogramm möglich, Blocksatz mit gleichengroßen Wortabständen zu drucken.
- 2) fehlerloser Rückwärtsdruck bei Proportionalschrift
- 3) genaue Linksbündigkeit bei Proportionalschrift
- 4) programmierbar auf 15 Zeichen pro Zoll
- 5) neue Grundeinstellung: Papierlänge DIN A4
- 6) 1. Zeile nach Papierwechsel (bei manueller Einzelblattzuführung) wird richtig gedruckt
- 7) manuelle Einzelblattzuführung (sheet-in-Taste) bewirkt nicht mehr ein Umschalten auf Sheetfeeder-Betrieb
- 8) Softwaremäßige Änderung der Proportionalschriftlängentabelle
- 9) Wahlweise, aber ohne Aufpreis: Änderung der Proportionalschriftlängentabelle im ROM auf Rustika II Typenrad (statt Polytron)
- 10) IEEE-Geräteadresse softwaremäßig änderbar

neues Druckerhandbuch

- 1) alle Angaben sind überprüft und bei der Arbeit mit dem Betriebssystem erstellt.
- 2) es werden wirklich alle Möglichkeiten des Betriebssystems beschrieben. Zum Beispiel:
 - softwaremäßige Änderung der Zeichensatztablelle (z.B. IBM oder ASCII)
 - Benutzung eines Phrasenspeichers
 - direkter Schreibzugriff auf das Drucker-RAM
 - Nennung aller wichtigen RAM-Adressen
 - direkte Wagenpositionierung
- 3) Beschreibung der hardwaremäßigen IEEE-Geräteadressen-Änderung
- 4) alle Neuerungen sind genau beschrieben
- 5) viele Beispiele

Abgabe der Betriebssystem-Erweiterung (zwei EPROMs) und des Handbuchs nur zusammen! Das Wechseln der EPROMs ist genau beschrieben und leicht ohne Werkzeug durchzuführen. Lieferung wie umseitig erläutert.

Preis bei Bestellung bis 31. März: DM 67.50
dannach: DM 75.--

Sammelbestellungsbeteiligungsbegehren

Absender:

Name

Straße

Plz. Ort

Hiermit beteilige ich mich an den Sammelbestellungen der
CBM Selbsthilfe 600/700 mit:

Stk.	Bezeichnung	Preis/Stk.	Gesamtpreis
=====	=====	=====	=====
	Basic 256k für 620/720	22.--	
-----	-----	-----	-----
	Kernal DIN/ASCII 600 mit Schalter	16.--	
-----	-----	-----	-----
	Zeichensatz DIN/ASCII 600	11.--	
-----	-----	-----	-----
	Kernal DIN/ASCII 700 mit Schalter	16.--	
-----	-----	-----	-----
	Zeichensatz DIN/ASCII 700	11.--	
-----	-----	-----	-----
	Zeichensatz ASCII 700	11.--	
=====	=====	=====	=====
	Zwischensockel ROM - ERPOM	7.--	
-----	-----	-----	-----
	Platine 24k Bank 15	18.--	
-----	-----	-----	-----
	dito bestückt mit Sockeln & Kond.	25.--	
-----	-----	-----	-----
	dito komplett mit RAMs	45.--	
-----	-----	-----	-----
	Gehäuse dafür	4.50	
-----	-----	-----	-----
	Experimentierplatine Bank 15	11.--	
-----	-----	-----	-----
	Portbaustein CIA 6526 A	25.--	
=====	=====	=====	=====
	8028 Farbband Nylon	7.--	
-----	-----	-----	-----
	8028 Farbband Multistrike	6.50	
=====	=====	=====	=====
-----	-----	-----	-----
	Versandkostenpauschale	5.--	
=====	=====	=====	=====
	Summe		

Ort, Datum, Unterschrift

Die Bestellung geht an Thomas Hingott, Im Hilgersfeld 10,
5060 Bergisch Gladbach-1. Den Bestellungen ist ein Verrech-
nungsscheck beizufügen. Keine Nachnahme!

Sammelbestellungsbeteiligungsbegehren

Absender:

Name

Straße

Plz. Ort

Hiermit beteilige ich mich an den Sammelbestellungen der
CBM Selbsthilfe 600/700 mit:

Stk. Bezeichnung	Preis/Stk.	Gesamtpreis
Basic 256k für 620/720	22.--	
Kernal DIN/ASCII 600 mit Schalter	16.--	
Zeichensatz DIN/ASCII 600	11.--	
Kernal DIN/ASCII 700 mit Schalter	16.--	
Zeichensatz DIN/ASCII 700	11.--	
Zeichensatz ASCII 700	11.--	
Zwischensockel ROM - ERPOM	7.--	
Platine 24k Bank 15	18.--	
dito bestückt mit Sockeln & Kond.	25.--	
dito komplett mit RAMs	45.--	
Gehäuse dafür	4.50	
Experimentierplatine Bank 15	11.--	
Portbaustein CIA 6526 A	25.--	
8028 Farbband Nylon	7.--	
8028 Farbband Multistrike	6.50	
Versandkostenpauschale	5.--	
Summe		

Ort, Datum, Unterschrift

Die Bestellung geht an Thomas Hingott, Im Hilgersfeld 10,
5060 Bergisch Gladbach-1. Den Bestellungen ist ein Verrech-
nungsscheck beizufügen. Keine Nachnahme!

Sammelbestellungsbeteiligungsbegehren

Absender:

Name

Straße

Plz. Ort

Hiermit beteilige ich mich an den Sammelbestellungen der
CBM Selbsthilfe 600/700 mit:

Stk.	Bezeichnung	Preis/Stk.	Gesamtpreis
=====	=====	=====	=====
	Basic 256k für 620/720	22.--	
-----	-----	-----	-----
	Kernal DIN/ASCII 600 mit Schalter	16.--	
-----	-----	-----	-----
	Zeichensatz DIN/ASCII 600	11.--	
-----	-----	-----	-----
	Kernal DIN/ASCII 700 mit Schalter	16.--	
-----	-----	-----	-----
	Zeichensatz DIN/ASCII 700	11.--	
-----	-----	-----	-----
	Zeichensatz ASCII 700	11.--	
=====	=====	=====	=====
	Zwischensockel ROM - ERPOM	7.--	
-----	-----	-----	-----
	Platine 24k Bank 15	18.--	
-----	-----	-----	-----
	dito bestückt mit Sockeln & Kond.	25.--	
-----	-----	-----	-----
	dito komplett mit RAMs	45.--	
-----	-----	-----	-----
	Gehäuse dafür	4.50	
-----	-----	-----	-----
	Experimentierplatine Bank 15	11.--	
-----	-----	-----	-----
	Portbaustein CIA 6526 A	25.--	
=====	=====	=====	=====
	8028 Farbband Nylon	7.--	
-----	-----	-----	-----
	8028 Farbband Multistrike	6.50	
=====	=====	=====	=====
-----	-----	-----	-----
	Versandkostenpauschale	5.--	
=====	=====	=====	=====
	Summe		

Ort, Datum, Unterschrift

Die Bestellung geht an Thomas Hingolt, Im Hilgersfeld 10,
5060 Bergisch Gladbach-1. Den Bestellungen ist ein Verrech-
nungsscheck beizufügen. Keine Nachnahme!

Sammelbestellungsbeteiligungsbegehren

Absender:

Name

Straße

Plz. Ort

Hiermit beteilige ich mich an den Sammelbestellungen der
CBM Selbsthilfe 600/700 mit:

Stk.	Bezeichnung	Preis/Stk.	Gesamtpreis
=====	=====	=====	=====
	Basic 256k für 620/720	22.--	
	Kernal DIN/ASCII 600 mit Schalter	16.--	
	Zeichensatz DIN/ASCII 600	11.--	
	Kernal DIN/ASCII 700 mit Schalter	16.--	
	Zeichensatz DIN/ASCII 700	11.--	
	Zeichensatz ASCII 700	11.--	
	Zwischensockel ROM - ERPOM	7.--	
	Platine 24k Bank 15	18.--	
	dito bestückt mit Sockeln & Kond.	25.--	
	dito komplett mit RAMs	45.--	
	Gehäuse dafür	4.50	
	Experimentierplatine Bank 15	11.--	
	Portbaustein CIA 6526 A	25.--	
	8028 Farbband Nylon	7.--	
	8028 Farbband Multistrike	6.50	
	=====	=====	=====
	Versandkostenpauschale	5.--	
	=====	=====	=====
	Summe		

Ort, Datum, Unterschrift

Die Bestellung geht an Thomas Hingott, Im Hilgersfeld 10,
5060 Bergisch Gladbach-1. Den Bestellungen ist ein Verrech-
nungsscheck beizufügen. Keine Nachnahme!

Riesiger Druckerpuffer zum Einbau

Endlich gibt es einen ausreichend großen Druckerpuffer für den 8028. Er faßt 31 kbyte, immerhin ca. 10 Schreibmaschinenseiten. Jetzt brauchen Sie beim Drucken nicht mehr bis 10 Minuten zu warten, bis Ihr Computer wieder frei ist. Der Druckerpuffer wird in den 8028 eingebaut. Auf der großen Digitalplatine des Druckers müssen dann fünf Leiterbahnen aufgetrennt und neue Verbindungen hergestellt werden.

Der Puffer ist lauffähig nur in Verbindung mit einer Abwandlung der oben beschriebenen Betriebssystemerweiterung und wird daher nur mit dieser zusammen abgegeben. Sollten Sie oder ein Bekannter in Ihrer Nähe den Einbau nicht vornehmen können, stehe ich Ihnen gerne zur Verfügung.

Komplettpreis 31k Druckerpuffer mit Einbauanleitung,
Betriebssystemerweiterung und Handbuch:

Preis bei Bestellung bis 31. März:	DM 216.-
Mit Einbau (nach Rücksprache)	DM 243.-
dannach:	DM 240.- bzw. DM 270.-

Besonderes Angebot:

Sie können mit den beschriebenen Druckererweiterungen etwa Ende März rechnen. Wenn Sie bis zum 31. März bestellt haben, erhalten Sie, wie angegeben, 10% Rabatt.

Software von Andreas Stankewitz

Super-Tape

DM 30.-

Kassetten-Treiber-Programm

EPROM, Kassette, Beschreibung.

Super-Tape ist ein sehr schnelles Kassettenaufzeichnungsverfahren, das mit der normalen Datasette eine Aufzeichnungsgeschwindigkeit von 3600 baud erreicht und 7200 baud mit einem guten Rekorder (oder Tonbandgerät), der über ein Interface angeschlossen werden kann.

Super-Tape ist vollständig ins Basic eingebunden. Nach dem Einschalten sind die LOAD-Routinen sofort verfügbar.

Bei einer Bestellung geben Sie bitte an, ob Sie eine 128k- oder eine 256k-Basicversion haben (an der Einschaltmeldung abzulesen).

Monitor 6000

EPROM für Bank 15-Erweiterungskarte.

DM 35.-

Diskette für Bank 15-RAM-Erweiterung (\$6000).

DM 30.-

(SFD 1001=CBM 8050/8250 oder CBM 1541)

Kassette im Super-Tape-Format für Bank 15...

DM 30.-

(siehe auch Bank \$0x KERNAL)

Sehr bald zu haben:

Bank \$0x KERNAL

DM 15.-

Hier ist das Kernal der Bank 15 so umgeschrieben, daß es in jeder anderen Speicherbank läuft. Dadurch ist es möglich, Maschinenprogramme ohne Bankumschaltroutinen, die sehr langsam sind, laufen zu lassen. (Diskette oder Kassette).

Monitor 6000 und Bank \$0x KERNAL zusammen.

DM 40.-

Der Monitor 6000 läuft dann in jeder Bank, ohne Erweiterungskarte! (Diskette oder Kassette).

neues PROTEXT Startprogramm zu den Selbstkosten DM 5.-
(PROTEXT ist ein Programm der Firma HARD+SOFT)

Das PROTEXT in seiner ursprünglichen Version arbeitet nicht mit einem auf ein Megabyte RAM ausgebauten Computer. Das neue Startprogramm beseitigt diesen Fehler. Es ersetzt in keiner Weise das Gesamtprogramm und umgeht auch nicht dessen Kopierschutz! Das neue Startprogramm wird wohl auch auf den Sammeldisketten des Clubs zu finden sein.

Hardware

Ich habe noch ein paar CBM 610 zuviel, die ich gerne möglichst ausgebaut verkaufen möchte. Die Geräte sind selbstverständlich in allen Funktionen geprüft.

Computer CBM 610 ohne Ausbau DM 150.-

Computer CBM 610, wenn dazu etwas Ausgebaut ist DM 130.-

Bank 15 Erweiterung \$0000-\$1fff +DM 30.-

mit Super-Tape +DM 23.-

DRAM Ausbau 256k +DM 150.-

DRAM Ausbau 1 Megabyte +DM 470.-

Basic 256k +DM 10.-

DIN/ASCII Tastaturbelegung und Zeichensatz +DM 40.-

Änderung des Bildschirmmodus entspr. CBM 700

mit DIN/ASCII umschaltbar +DM 40.-

mit ASCII +DM 30.-

Ich baue auch Ihren schon vorhandenen Computer aus. Detailfragen bitte ich Sie telefonisch oder schriftlich mit mir zu klären.

Wie Sie bei mir bestellen können:

Software verschicke ich, sobald bei mir eine Zahlung für das entsprechende Programm eingegangen ist. Dies kann eine Überweisung oder ein Verrechnungsscheck sein. Auf der Überweisung oder dem Scheck bitte den Programmnamen eintragen. Auch Ihre vollständige Adresse sollten Sie nicht vergessen (passiert bei Überweisungen schon manchmal). Nachnahmesendungen gebe ich nur sehr ungerne auf weswegen hierbei die vollen Portogebühren dem Empfänger angerechnet werden.

Computer verschicke ich unter den gleichen Bedingungen wie beim Softwareversand. Zusätzlich berechne ich noch DM 10.- für Porto. Hierbei sind auch Nachnahmesendungen ohne weitere Kosten möglich. Eine ausführlichere Bestellung halte ich bei den Computern allerdings für sinnvoll (Karte oder Brief).

Meine Bankverbindung: Postgiroamt Köln, BLZ 370 100 50, KNr.3842 87-505.

Meine Adresse: Andreas Stankewitz, Lindenstr. 15 Hinterhaus, 5000 Köln-1, Tel.: 0221-230055

