

Genie / TRS-80

User Club

'Bremerhaven'

Club INFO \* \* Club INFO \* \* Club INFO \* \* Club INFO

Ausgabe: 05 / 1987  
Mai

Jahrgang: 5

Druck: Peter Spieß

Redaktion: Ralf Folkerts

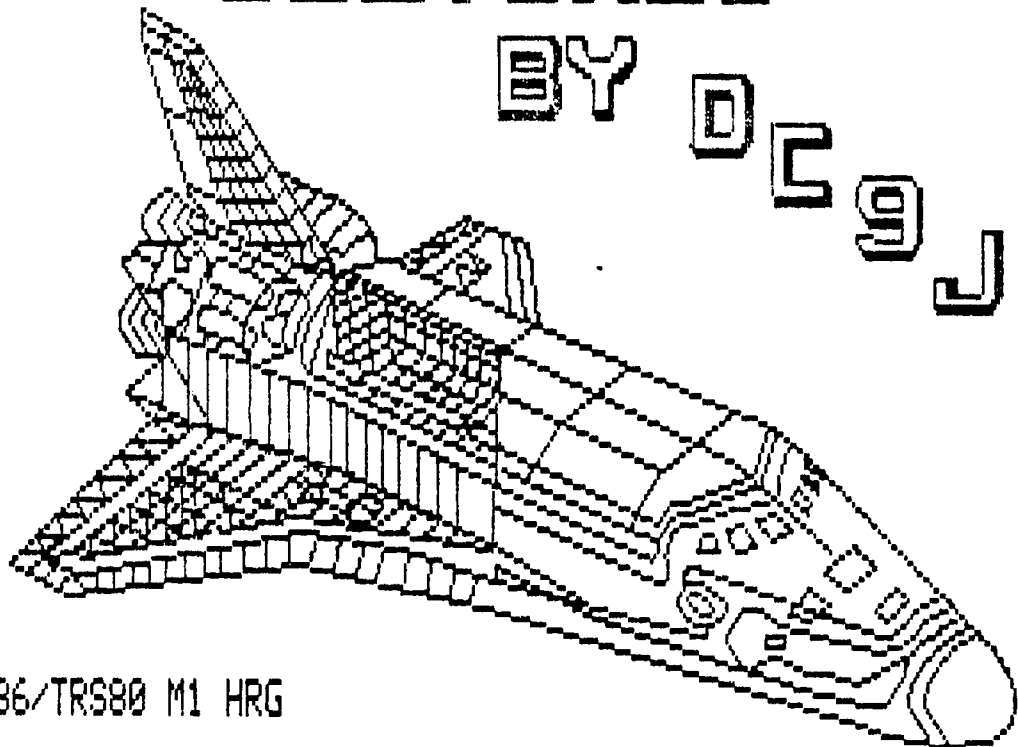
Freeware: Gerhard Loose

Auflage:

095 Ex

**SPACE SHUTTLE  
DESIGNED**

**BY O C 9 J S**



1986/TRS80 M1 HRG

Inhalt der Ausgabe 05/1987:

- 01 - 01 Titelblatt (Grafik von Gerhard Loose)
- 02 Inhaltsverzeichnis, neues Mitglied
- 03 Internes vom Betreuer
- 04 Der Fall 'Druckermotor'
- 05 Kurzinfo von Gerhard Loose
- 06 - 16 Selbstbau Plotter von Gerhard Loose
- 17 Ergänzung zur GDP-64K von Jörg Seelmann-Eggebert
- 17 DFÜ - News von Gerhard Loose
- 18 - 19 Das Clubleben von Jörg Seelmann-Eggebert
- 19 Ein Praxistip für Bastler von Jörg Seelmann-Eggebert
- 20 - 30 Adreßliste von Paul Kröher  
Mit freundlicher Genehmigung von:  
Röckrath Mikrocomputer  
Kaiserstraße 54  
4050 Mönchengladbach 1

\*\*\*\*\*

**Neues Mitglied:**

Peter Klenner aus dem Nelkenpfad / Burlo. Er 'fährt' ein Genie I mit 2\*Disk 80 Tr., DS/DD. Seinen Printoutput erledigt ein Panasonic P 1092. Hardwaremäßig möchte er sich eine HRG 1-B einbauen (GDP-64K ?), um mit seinem Gerät Platinenvorlagen erstellen zu können. Des weiteren plant er, seinen Compy in Verbindung mit einem Funkgerät einzusetzen. Auf der Software-Seite interessiert er sich für Textverarbeitung und für Programmierung in Pascal. Außerdem sucht er gute Assembler - Lehrprogramme.

Liebe Clubkollegen,

nun ist sie da: unsere MAI Ausgabe. Ich hoffe, Ihr habt den 'Sprung' in diesen Monat alle gut verkraftet.

Mein erstes Thema auf dieser Seite ist diesmal der Erscheinungstermin der April - Ausgabe. Diese liegt heute nämlich noch hier (sortiert, geleimt und zum Teil sogar schon in den Umschlägen). Da mich das späte Erscheinen der INFOs auch nervt, habe ich die April Ausgabe frühzeitig fertig gemacht und an Peter geschickt. Dieser hat sich auch Mühe gegeben; am 25.04. waren die INFOs wieder hier. Die einzige Sache an der es hapert sind die: A d r e ß a u f k l e b e r. Auf der nächsten Seite werde ich die Sache mal etwas genauer erörtern. Hier nur soviel: Nachdem mir der Motor des Druckers im November '86 durchbrannte, konnte ich die INFOs noch mit dem Commodore 8028 erstellen und mit diesem auch die Adreßaufkleber drucken. Dieser ist mir jedoch nun auch 'sauer' geworden, so daß ich nur noch mit dem NEC 8023 arbeiten kann. Bei diesem kann ich jedoch die Adreßaufkleber nicht in den Traktor bekommen - und im Friktions - Modus läuft die Bahn schnell schief. Ich habe es dreimal probiert und dabei einen ganzen Haufen Aufkleber 'verjubelt'). Beim ersten Mal lief es fast ganz durch aber 'hakete' bei den (ca.) letzten 15. Ich habe den Druck darauf abgebrochen und es noch einmal probiert hier 'schaffte' ich so um die 35 Aufkleber. Beim dritten und letzten Versuch kamen gar nur noch 10 Stück durch.

Da ich jedoch im Programmteil 'Drucken Adreßaufkleber' alle Funktionen, die den Versand betreffen, integriert habe, kann ich die letzten Aufkleber nicht 'von Hand' drucken, da hierdurch die Personen aus der 'Rückstandsauflösung' nicht erfaßt würden (und der gesamte Mitgliederstamm 'durcheinander' wäre).

Ich probiere z.Zt. 'krampfhaft' das LTERM Programm aus unserer Freeware - Bibliothek an die RB RS-232 anzupassen. Der Gerhard Loose benutzt diese - und wir suchen nach einem Weg, die Daten aus der Bibliothek per Telefon auszutauschen. Das Programm macht jedoch aus irgendeinem Grund nicht mit: Nachdem ich alle Parameter an die RB Schnittstelle angepaßt habe, hängt der Rechner sich jedesmal sofort nach dem Aufruf des 'M' (Reset RS-232) auf, ohne auch nur eine Eingabe abzuwarten. Alle anderen Funktionen arbeiten ohne Fehler. Weiß jemand hier weiter (ich werde inzwischen mal DEBUG bemühen).

Meinen Dank möchte ich A. Röckrath für seine Erlaubnis, die Adressliste von Paul Kröher veröffentlichen zu dürfen, aussprechen.

Seit einiger Zeit arbeite ich auf dem Modell III mit der 5.3.0 Release von LDOS. Dies ist wirklich eine hervorragende Erweiterung der 'alten' 5.1.4a Release dieses OSs. Für die nächste Ausgabe werde ich einen kleinen Testbericht über dieses spitzenmäßige DOS bringen. Für alle Modell Ier jedoch ein Wermutstropfen: Die 5.3.0 gibt es nur für das Modell III (und als LS-DOS 6.3.0 für das Modell IV (als Nachfolger des TRSDOS 6.2.x)).

Uff, damit ist mal wieder eine Ausgabe des INFOs fertig. Ich hoffe jetzt, daß ich den richtigen Druckermotor bald bekomme, um endlich die INFOs versenden zu können und bitte um Euer Verständnis für die Verzögerung des Versandes der April Ausgabe. Meine Befürchtung ist nun die, diese Mai Ausgabe auch nicht 'rauszubekommen, falls mir der Motor nicht bald vorliegt.

Bis zur nächsten Ausgabe wünsche ich Euch alles Gute.

*Poly*

3

Nachfolgend nun die genaue Schilderung des Falles 'Druckermotor':

Das Ganze ging los, als mir im November ('86) der Papierantriebsmotor meines Star Gemini-10x - ich benutzte diesen für das Drucken von Adreßaufklebern für die INFOs und von Listings meiner COBOL Mitgliederverwaltung - durchbrannte. Keine große Sache, sollte man meinen. Es handelt sich um einen unipolaren Stepper mit 7,5 Grad/Step und 50 Ohm. Nun gut; ich fragte sowohl bei Star direkt als auch bei Firma Conrad Electronic (in Hirschau) an. Von Star erhielt ich einen Brief, in dem mir mitgeteilt wurde, daß ich den Motor nicht direkt bei Star, sondern nur über einen Star-Fachhändler beziehen könnte. Beigelegt war eine Übersicht aller Star Händler. Ich fragte daher bei einem dieser Händler an. Das Angebot von Conrad erhielt ich dann am 06.12.1986; Lieferzeit ca. 2 Wochen nach Auftragseingang. Das Angebot des Star - Händlers erhielt ich am 23.12.1986. Lieferzeit dort ca. 1 Woche. Da dieses Angebot wesentlich höher als das von Conrad war, bestellte ich den Motor noch am 23.12 bei Conrad. Ich gab dabei sowohl in meiner Anfrage als auch in der Bestellung die Typenbezeichnung des Motors an. Ferner wies ich darauf hin, daß es sich um den Motor für den Papiertransport handele. In meiner Bestellung gab ich als Zahlungswunsch Bankeinzug an und teilte auch meine Bankverbindung mit.

Da ich daraufhin nichts mehr von Conrad hörte, schickte ich am 31.01. eine Anfrage ab, wo der Motor bliebe. Auch in dieser Anfrage gab ich die Typenbezeichnung des Motors, die Druckertyp und den Hinweis, daß der Motor für den Papiertransport bestimmt sei, an.

Auf diese Anfrage erhielt ich am 07.02. eine Karte mit dem Hinweis, daß meine Bestellung nicht vorliege und daß mein Schreiben vom 31.01. als Bestellung bearbeitet würde.

Die INFOs habe ich seit der Dezember - Ausgabe mit einem CBM8028 Typenraddrucker erstellt, auch die Adreßaufkleber druckte ich mit diesem Drucker, da sie in den NEC nicht passen.

Da ich den Motor trotz der Bestätigung wieder nicht erhielt, schickte ich am 11.03. eine weitere Anfrage ab, wo der Motor bliebe (wie erwähnt, wurde in dem Angebot eine Lieferzeit von ca. 2 Wochen genannt). Am 26.03. erhielt ich dann, oh Wunder, ein Päckchen von Conrad Electronic. Die Freude war groß, es wurde gejubelt.... jedoch nur solange, bis ich dieses öffnete. Inhalt war ein Motor für den Star Gemini-10x - jedoch der für den Druckkopf zuständige. Das Päckchen kam übrigens nicht per Nachnahme; als Zahlungsart stand in der Rechnung vielmehr 'Bankeinzug'. Diesen Zahlungswunsch hatte ich jedoch nur in meiner 'eigentlichen' Bestellung vom 23.12. - die ja nach der Karte vom 05.02. gar nicht vorlag - erwähnt. Und auch nur dort habe ich die Einzugsermächtigung erteilt.

Na, was soll's. Sofort nachdem ich den Fehler festgestellt hatte, habe ich einen Brief an Conrad geschrieben, in dem ich mitteilte, daß man mir den falschen Motor zugeschickt hatte. In diesem Brief bat ich auch um Zusendung des richtigen Motors. Da ich befürchten muß, daß mir der Druckkopfmotor u.U. auch 'durchgeht', erklärte ich, daß ich den falschen Motor behalten würde. Daraufhin rief Firma Conrad am 30.03. an, um mitzuteilen, daß der richtige Motor zugeschickt würde. Am 06.04. rief Conrad Electronic noch einmal an und fragte nach dem Druckertyp, für den der Motor bestimmt sei (den Druckertyp habe ich sowohl in meiner Anfrage als auch in der Bestellung und der Anfrage vom 31.01. angegeben). Bis heute (02.05.) ist der Motor nicht bei mir eingetroffen. Wie lange es noch dauert bis er kommt kann ich auch nicht sagen. Solange ich den Motor nicht habe bin ich jedoch nicht in der Lage, die Adreßaufkleber zu drucken. Aufgrund dieser Tatsache kann ich auch keinerlei Reports über Löschung/Neuaufnahmen drucken, da diese erst nach dem Druck der Aufkleber erstellt werden können.

TRS-80/GENIE USER CLUB  
Public Service  
Gerhard Loose

=====

Ich GrüÙe Euch.

\*\*\*\*\*

K U R Z I N F O

\*\*\*\*\*

Wie die PC-Woche in der Ausgabe vom 6. April 1987 Mitteilte, hat Tandy vor einigen Wochen dir 17 eigenen Computershops in der Bundesrepublik Deutschland geschlossen. Die in Ratingen verbliebene Tandy-Rumpforganisation, die aus vier Mitarbeitern besteht, will sich um einen Ausbau des Händlernetzes bemühen. Die Anzahl der Händler beleuft sich zur Zeit auf 27. Der im belgischen Namur ansässige Europa-Direktor von Tandy, Peter Stein, erklärte das sich Tandy nicht vom deutschen Markt zurückziehen werde. Es ist aber notwendig die Strategie auf dem härtesten Markt Europas zu überdenken.

Im Geschäftsjahr 85/86 sei bei einem Umsatz von 6.2 Millionen Dollar vier Millionen Dollar Verlust gemacht worden. Bis Ende Februar 1987 belief sich der Verlust auf schon 2.5 Millionen Dollar bei nur 5.5 Millionen Dollar Umsatz. Alle Tandy Kunden werden in den nächsten Wochen angeschrieben, um sie über Händleradressen und den zukünftigen Service zu informieren, da auch der Firmeneigene Reparaturdienst in Ratingen abgebaut wird. Den Kundendienst übernimmt die Firma Hoffmann in Andernach. Tandy hat 54 Mitarbeiter in BRD entlassen.

In Frankreich, Belgien und den Niederlanden ist die zur Kanadischen Interan Inc. gehörende Tandy-Europa-Organisation mit eigenen Läden scheinbar erfolgreicher. In Frankreich wurde die Zahl der Shops im vergangenen Jahr auf 150 im gegensatz zu 100 im vergangenen Jahr erhöht. In Belgien gibt 116 und den Niederlanden 52 Läden.

(Aus PC-Woche vom 6.04.1987)

(GELOS)

\*\*\*\*\*

5

TRS-80/GENIE USER CLUB

Gerhard Loose

-----  
Ich GrüÙe euch.

Ich bin nun mal ein hoffnungsloser Graphik Freek, und so ist mir beim durchstüßern meiner Unterlagen die Bauanleitung für einen Scanner in die Hände gefallen, die vor einiger Zeit im WDR Computerclub veröffentlicht wurde. Diese möchte ich euch nicht vorenthalten. Mit ein paar kleinen Änderungen läßt sich die ganze Geschichte, sie ist in der Anleitung für einen Appel beschrieben, auch für unsere Maschinen nutzen. Und wenn wir weiter spinnen, kann man sich das Ding auch als Plotter vorstellen. Man ersetze nur den Reflexkoppler durch einen Schreibstift, gesteuert durch einen Hubmagneten. Na wie wär's ?

Wer hat mal wieder richtig Lust zum Stricken? Auch die Erstellung der Software kann eine reizvolle Aufgabe sein. Ich wünsche euch auf jeden Fall viel Spaß dabei. Und schreibt doch über eure Arbeit einmal einen Beitrag für die Info. Nur keine Angst davor. Wie ihr lest, auch ich bin keiner grosser Schreiber, aber was soll's.

Und noch eins: Wenn ihr die Software erstellt habt. Ab damit in die Clubeigene FreeWare-Bank.

Bei aller Hackerei : Ihr solltet nicht vergessen

**COMPUTER LACHELN NICHT WENN SIE EUCH WECKEN**

Deshalb, denkt auch einmal an eure besseren Hälften.  
So, nun viel Spaß beim Lesen und Nachbauen euer

GERHARD

6

## SELBSTBAU - SCANNER

Der nachstehend beschriebene Selbstbau-Scanner ermöglicht uns auf einfachstem Wege die Speicherung einer Bildvorlage mit Hilfe eines Rechners. Dieser Bauvorschlag soll allerdings keine endgültige Lösung für Hobbyisten darstellen. Er soll zu eigenen Weiterentwicklungen und Aktivitäten im Hardwarebereich anregen. Um ein Bild mit einem Rechner darstellen oder speichern zu können, muß man jeden Bildpunkt in eine für den Rechner verständliche Information umwandeln. In unserem Falle (APPLE) 320 horizontale und 200 vertikale Punkte, also 64000 Bildpunkte insgesamt.

Weil mit einem 8-Bit-Prozessor keine 64000 Informationen auf einmal (parallel) übernommen werden können, bleibt nur die Möglichkeit der etwas zeitraubenderen, dafür aber wesentlich kostengünstigeren, seriellen Bildumwandlung. Hierzu wird das zu digitalisierende Bild auf einer Trommel befestigt, diese in Drehung versetzt und mit einem Reflexkoppler Punkt für Punkt und Zeile für Zeile abgetastet. Dieser Reflexkoppler besteht aus einer Lichtquelle (Leuchtdiode) die genau einen Bildpunkt beleuchtet und einem Fototransistor, der durch die reflektierte Lichtmenge des Bildpunktes eine Widerstandsänderung erfährt. Diese Widerstandsänderung wird durch eine externe Beschaltung in eine Spannungsänderung umgesetzt und mittels eines Schmitt-Triggers, dessen Umschaltswelle (Kontrast) einstellbar ist, mit den TTL-Pegeln High und Low dem Rechner zugeführt.

Trotz Verzicht auf jegliche Synchronisation (außer Zeilenvorschub) kann man bei sorgfältiger Arbeitsweise während der Herstellung des Scanners erstaunliche Resultate erzielen. Die Bildpunktauflösung des beschriebenen Scanners ist  $< 0,25 \text{ mm}^2$  bei einem Kostenaufwand von insgesamt weniger als 50.-DM. Aber gerade in Bezug auf Synchronisation (man bedenke, daß 320 Bildpunkte pro Zeile ohne fest definierte Anfangs- und Endabfragezeiten übernommen werden) und Kontrastumfang (Graustufung des Bildes) bieten sich jedem Nachbau-Interessenten unzählige Möglichkeiten der hard- und softwaremäßigen Verbesserung.

### Übersicht

Das Hauptproblem, die möglichst konstante Drehzahl der Bildtrommel zu erreichen, läßt sich am einfachsten bewältigen. Man nehme einen beliebigen Plattenspieler, eine ausgediente, Langspielplatte, montiere ein Kunststoffrohr auf die LP, setze das entstandene Gebilde auf den Plattenteller, und schalte den Plattenspieler ein. Die Bildtrommel dreht sich dann mit konstant 33,3 Umdrehungen pro Minute. Bild 1 zeigt die gesamte Anordnung des Scanners.

Als nächstes muß der Sensor zeilenweise (vertikal) an der Bildtrommel entlang geführt werden.

Hierzu befestigen wir einen Reflexkoppler an dem Arm eines Schlittens, der sich auf einer parallel zur Trommel montierten Führungsschiene befindet und verbindet diesen mit einer an einem Schrittmotor befestigten Gewindestange. Wird der Schrittmotor in Drehung versetzt, erreichen wir jetzt die gewünschte, vertikale Positionsänderung der Abtasteinheit.

Um Taumelbewegungen der Bildtrommel zu verhindern, wird am oberen Ende des Führungsgestänges ein justierbarer Arm mit einem Kugellager als Laufrad befestigt. Dieses Laufrad garantiert später einen gleichmässigen Abstand zwischen Bild und Sensor.

Das Problem der Zeilensynchronisation wird mit einem zweiten, ebenfalls justierbaren, kürzeren Arm mit aufgeklebtem Reedrelais am unteren Ende des Gestänges und einem kleinen, auf die Schallplatte geklebtem Magneten gelöst.

Zum Schluß muß man den Reflexkoppler nur noch mit einer Schaltung verbinden, welche die analogen Widerstandsänderungen des Fototransistors in digitale TTL-Pegel umwandelt und somit den Anschluß an einen Rechner ermöglicht.

### Werkzeuge

Für die Herstellung des Scanners werden lediglich Bleistift, Lineal, Winkelmesser, Zirkel, Holzsäge, kleine Eisensäge, halbrunde Raspel, halbrunde Schlüsselfeile, feines Schmirgelpapier und eine Bohrmaschine mit Holzbohrern 0,8/2/4,5 mm Stärke sowie Lötkolben und Werkzeuge zur Fertigung der Elektronik-Platine benötigt.

### Bildtrommel

Als Bildtrommel eignen sich Kunststoff-Abflußrohre. Will man DIN A 4-Vorlagen abtasten, so sollte der Außendurchmesser mindestens 10 cm und die Länge des Rohres mindestens 30 cm betragen. Eine Langspielplatte als Grundplatte zu verwenden, bringt den Vorteil einer einfachen Zentrierung des Rohres. Mit Zwei-Komponenten-Schnell-Kleber werden Rohr und Platte folgendermaßen miteinander verbunden:

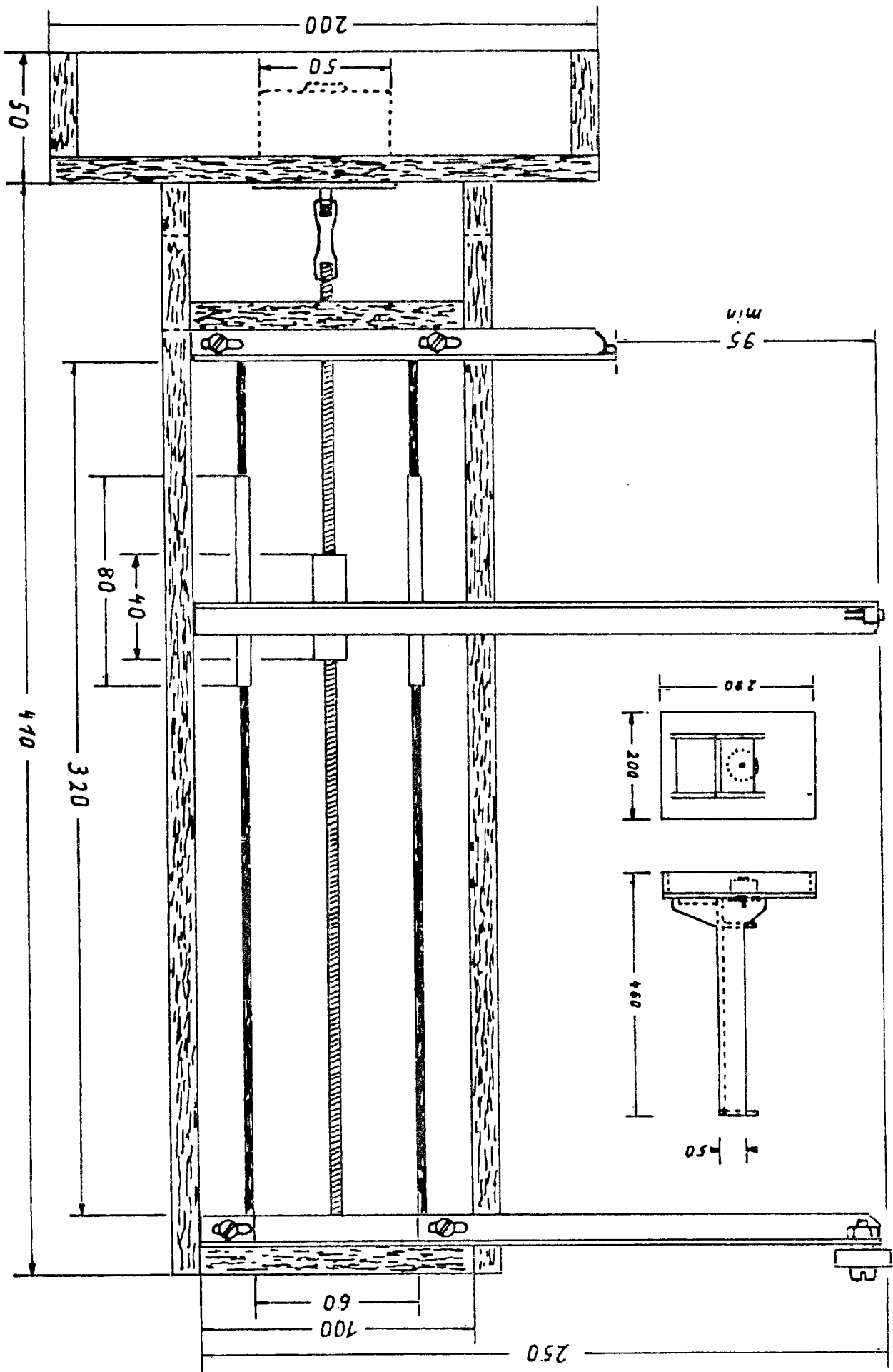
Schallplatte auf eine glatte Unterlage legen, das sauber und winklig abgeschnittene und verschliffene Rohr auf der Platte zentrieren und mit drei bis vier Klebepunkten an der äußeren Stoßkante fixieren. Die exakte Einhaltung des 90 Grad Winkels zwischen Rohr und Platte sind Voraussetzung fuer die spätere, saubere Bildabtastung. Also lieber einmal mehr als einmal zu wenig diesen Winkel überprüfen, denn eine spätere Korrektur ist so gut wie unmöglich. Sind die Fixierpunkte angehärtet so wird die Stoßkante rund herum mit Kleber beschichtet.

### Abtasteinheit

Das Führungsgestänge besteht aus zwei 3 mm Stahldrähten, die in einem Abstand von 60 mm auf zwei Aluprofilen befestigt sind. Genau mittig zwischen diesen Stahldrähten verläuft die für die Schlittenbewegung benötigte M4-Gewindestange. Diese wird durch zwei Kugellager gehalten.

Die Montage des Gestänges erfolgt am einfachsten mit Hilfe eines DIN A 3 großen Millimeterpapiers auf dem die gesamte Mechanik im Maßstab 1:1 aufgetragen wurde. Zunächst werden die Halteprofile auf das erforderliche Maß zurecht gesägt und mit den Bohrungen für Schrauben und Kugellagern versehen. Die Bohrungen für die Kugellager müssen groß genug sein, um den M4-Muttern, die die Gewindestange mit den Lagern verbinden, genügend Spielraum für eine leichtgängige Drehung zu ermöglichen (Bild 2). Da eine





Bohrung in Aluminium mit einem Spiralbohrer höchstens ein gleichschenkliges Dreieck hinterläßt, verwenden wir nur Holzbohrer. Für das Kugellager wird mit einem 4,5 mm Bohrer vorgebohrt und anschließend die Bohrung mit der Schlüsselfeile auf ca. 10 mm vergrößert. Die Federstahldrähte werden etwa 1 mm kürzer als erforderlich abgelängt (um evtl. nötig werdenden Feilarbeiten vorzubeugen) und mit den sauber entgrateten Messingröhrchen (Innendurchmesser 3,2 mm) bestückt.

Bis der Kleber des Führungsgestänges ausgehärtet ist, werden aus Alu-Profil die Arme für das Reedrelais, für das Kugellager, für den Sensor und die Gewindebuchse des Schlittens abgelängt und entgratet. In den Armen des Abstandhalters und der Relais-Halterung werden, passend zu den Bohrungen in den Profilen des Führungsgestänges, Langlöcher von ungefähr 10 mm Länge eingelassen (hierdurch ist nachher die Justierung des Sensorabstandes und der Zeilensynchronisation möglich). Anschließend wird die Gewindestange auf Maß gesägt und an den Enden mit der Schlüsselfeile abgerundet. Jeweils zwei Muttern werden zur Justierung der Kugellager verwendet und zwei dienen zur Herstellung der Gewindebuchse (Bild 3). Justiert werden die Kugellager direkt am Führungsgestänge. Der Abstand Gewindestange-Stahldraht darf an den Profilen maximal um 0,1 mm variieren. Ansonsten ist keine Leichtläufigkeit zu erreichen.

Der nächste Schritt ist die Herstellung des Schlittens. Hierzu wird das Führungsgestänge mit der Vorderseite nach oben auf die Zeichnung gelegt. Die Messingröhrchen werden bis zum Anschlag nach einer Seite geschoben und der Sensorhalter im rechten Winkel zu den Stahldrähten mit den Rohren verklebt.

Sind auch diese Verbindungen ausgehärtet, wird das Gestänge wiederum gewendet und das kurze Profilstück mit den beiden Muttern und dem Sensorarm zu einer Gewindebuchse verklebt. Es ist unbedingt darauf zu achten, daß kein Klebstoff mit der Gewindestange in Kontakt kommt und die bisherige Arbeit zu nichte macht. In Bild 4 sind Details der drei Arme zu sehen.

## Gehäuse

Der Bau des tragenden Gehäuses ist -je nach eigenen Vorstellungen über das spätere Aussehen- wenig problematisch. Einzige Bedingung ist die Möglichkeit der vertikalen Montage unserer Abtasteinheit und eine gewisse Dreh- und Standfestigkeit. Das hier beschriebene Gehäuse wurde aus einer 65x41 cm großen und 10 mm starken Spanplatte gefertigt. Bild 5 zeigt, wie man die Teile aus der Platte schneidet und in einer Explosionszeichnung die Anordnung der Teile zum Zusammenbau des Turmes.

Deckel (2) und Zwischenstück (3) erhalten die Bohrungen für die Schrauben und für die Gewindestange der Abtasteinheit und werden mit Weißleim auf die Rückwand (1) geklebt. Die Anleimung der Seitenteile (4,5) und des Stabilisierungsstückes (6) beenden den Turmbau. Um einen möglichst strammen Sitz des Schrittmotors im Gehäuseboden zu ermöglichen, muß das Loch etwas kleiner ausgesägt werden und dann mit der Raspel auf den erforderlichen Durchmesser aufgefellt werden. Nach dem Ankleben der Füße steht einer Lackierung nichts mehr im Wege.

## Montage

Die Abtasteinheit wird ohne Verwindung mittels 4 Holzschrauben in dem Turm befestigt. Als Kupplung zum Motor wird ein Stück Kunststoffschlauch auf die Gewindestange geschraubt und anschließend auf die Achse des Schrittmotors geschoben. Der Turm wird nun so ausgerichtet, daß ein minimales, axiales Spiel zwischen Gewindestange und Motorachse entsteht. Mit Hilfe des restlichen Alu-Profiles (zwei 10 mm lange Stücke mit je 4 Bohrungen versehen) wird der Turm mit dem Gehäuseboden verschraubt.

## Sensorblende

Der von uns verwendete Reflexkoppler CNY-70 besitzt nur ein sehr geringes Auflösungsvermögen. Mit etwas Fingerspitzengefühl und Geduld ist es jedoch möglich, eine mechanische Optik herzustellen, mit der eine für uns notwendige Auflösung 1x1 mm zu erreichen ist.

Die Blende muß für diesen Zweck die Maße 5x5x2 mm haben und zwei Bohrungen von 0,8 mm besitzen. Sie kann aus dem restlichen Aluminium-Profil hergestellt werden.

Zuerst werden auf der Innenseite der Blende zwei Körnungen in einem Abstand von 2,5 mm angebracht, und der Umriss der Blende aufgetragen. Das Aluprofil wird jetzt möglichst in einem Schraubstock so befestigt, daß man die Bohrmaschine mit aufgestützten Armen halten kann. Der Bohrer wird so geführt, daß seine Achse in einem Winkel von 15 Grad in Richtung der zweiten Bohrung zeigt. Die zweite Bohrung wird auf gleiche Weise hergestellt. Nach dem Entgraten der Blendeninnenseite mittels eines 2mm-Bohrers, dem Aussägen und dem sauber 'auf Winkel' Feilen der Seiten, besitzt die Blende den in Bild 6 gezeigten Querschnitt. Der Reflexkoppler und das Reed-Relais werden nach Vorlage auf die entsprechenden Arme geklebt. Das Reed-Relais muß so angebracht werden, daß bei Vorbeiführen eines Magneten der Kontakt geschlossen wird. Der Sensor muß mit horizontal angebrachten Löchern montiert werden. Der mechanische Teil der Herstellung unseres Scanners ist hiermit beendet. Wir können uns nun dem elektronischen Teil widmen.

## Scanner-Elektronik

Um den Scanner mit dem Rechner zu verbinden, brauchen wir drei elektronische Schaltungen.

- a. Motorsteuerung für den Schrittmotor
- b. Signalaufbereitung für die Motorsteuerung
- c. Signalaufbereitung zur Übergabe der Bildinformation und Zeilensynchronisation an den Rechner

Die gesamte Schaltung und ein Platinenlayout sind in Bild 7 abgebildet.

### a. Motorsteuerung für den Schrittmotor

Der Schrittmotor wird mit dem Steuer-IC SAA1027 in unipolarer Betriebsweise angesteuert. An weiteren Bauteilen sind nur noch zwei Widerstände und ein Kondensator erforderlich. An Pin 15 des IC's werden die Impulse zur Drehung des Motors angelegt und durch

High- oder Low-Pegel an Pin 3 die Drehrichtung des Motors bestimmt. Die Ausgänge Q1 - Q4 (Pin 6,8,9,11) werden direkt mit den Wicklungen des Motors verbunden.

#### b. Signalaufbereitung für die Motorsteuerung

Das IC SAA1027 hat den Nachteil, Spannungen unter 8V nicht als Steuerspannungen anzunehmen. Die 5V des Rechners reichen also nicht aus. Aus diesem Grunde werden die TTL-Pegel durch NAND-Gatter mit Open-Kollektor-Ausgängen auf Betriebsspannung der Gesamtschaltung erhöht. Die beiden jeweiligen Eingänge der Gatter werden zusammengeschaltet und die Ausgänge durch jeweils einen Widerstand mit der Versorgungsspannung verbunden und. Somit liegen sie im Ruhezustand auf High-Pegel.

#### c. Signalaufbereitung Bild und Zeilensynchronisation

Der Strom, der durch die LED des Reflexkopplers fließen soll, wird durch einen Widerstand auf ca. 16 mA begrenzt. Da die Intensität des reflektierten Lichtes im Fototransistor eine Widerstandsänderung hervorruft, wird dieser in einen Spannungsteiler integriert, dessen Ausgangsspannungsänderung durch einen Spindeltrimmer bestimmt wird (Arbeitspunkteinstellung). Diese Spannungsänderungen werden durch einen als Schalttransistor betriebenen NPN-Transistor ausgewertet und auf ein Gatter des 74LS26 gegeben. Das auf gleiche Weise beschaltete letzte Gatter des IC's, wird für den Synchronimpuls verwendet. Durch einen kleinen Magneten, der auf den Rand der Schallplatte geklebt wird, wird bei jeder Umdrehung der Schallplatte das Reed-Relais betätigt. Somit wird der Ausgang des Gatters für die Dauer der Betätigung auf High-Pegel gesetzt. Die RC-Kombination vor dem Gatter unterdrückt einen durch Kontaktprellen möglichen Fehlimpuls.

#### Aufbau der Elektronik

Um den Nachbau der Elektronik möglichst einfach zu halten, sind dem Schaltbild ein Platinenlayout, ein Bestückungsplan (Bild 8) und eine Bauteileliste beigelegt. Bestückt wird die Platine in der Reihenfolge: Widerstände, Fassungen, Trimmer, Kondensatoren, Transistoren und Steckkontakte. Zuletzt werden die IC's in ihre Fassungen gesteckt. Die Verbindung mit dem Scanner beendet diesen Bauabschnitt.

#### Bauteile:

IC1	: SN74 LS 26	IC2	: SAA 1027
C1	: 0,1 uF	C2	: 10 nF
R2/6	: 100 Ohm	R10	: 680 Ohm
R1/3/4/5	: 1 kOhm	R7/8	: 4,7 kOhm
R9	: 47 kOhm	R11	: 25 kOhm Wendel
TI	: BC 238		

## Steckstifte:

1 Anode Sendediode	13 (-)Statorwicklung Q1
2 Kathode Sendediode	14 (+)Statorwicklung Q1
3 Emitter Fototransistor	17 Eingang Schritt
4 Kollektor Fototransistor	18 Eingang Auf/Ab
5 Reedrelais	19 Ausgang Sync
6 Reedrelais	20 Ausgang Bild
7 (-)Statorwicklung Q4	
8 (+)Statorwicklung Q4	22 Masse Rechner
9 (-)Statorwicklung Q3	23 + 12 V
10 (+)Statorwicklung Q3	24 0 V (Masse)
11 (-)Statorwicklung Q2	
12 (+)Statorwicklung Q2	

## Teileliste

Anz Einh. Bezeichnung	Anz Einh. Bezeichnung
1 Stück Schrittmotor	21 Stück Steckstifte 15x1 mm
1 Stück Reflexkoppler CNY70	21 Stück Steckschuhe 15x2 mm
1 Stück SAA 1027	1 Stück Platine 80x55 mm
1 Stück SN 74LS26	130 cm Alu-Winkel-Profil 12x12x2 mm
1 Stück BC 238	68 cm Stahldraht 3 mm
1 Stück Kondensator 0,1µF	16 cm Messingrohr 3,2 mm innen
1 Stück Kondensator 10 nF	38 cm Gewindestange M4
2 Stück Widerstand 100 Ohm	2 Stück Kugellager 4 mm innen
1 Stück Widerstand 680 Ohm	1 Stück Kugellager 20 mm außen
4 Stück Widerstand 1 kOhm	1 Stück Spanplatte 65x41cm 10mm dick
2 Stück Widerstand 4,7 kOhm	
1 Stück Widerstand 47 kOhm	
1 Stück Wendel-Trimmer 25 kOhm	
1 Stück Reed-Relais	
1 Stück IC-Fassung 14 polig	
1 Stück IC-Fassung 16 polig	

## Sonstiges:

Weißleim  
2-Komponenten-Kleber  
M4 Schrauben/Muttern/Unterlegscheiben

## Anschluß an Apple-Rechner und Kompatible

Die von uns entwickelte Software zum Betrieb des Scanners wurde auf einem Apple II Rechner entwickelt. Um den Scanner mit dieser Software direkt betreiben zu können, wird die Scanner-Elektronik an den Spiele Ein-/Ausgabeanschluß (GAME I/O, Buchse rechts neben den Einsteckschlitten) des Rechners angeschlossen. Als Stecker benutzen wir einen 16 poligen DIL-Steckverbinder. Ein 5 poliges Flachbandkabel dient als Verbindungskabel.

## Anschluss des DIL-Steckers

		o1	16o		
Bild	SW0	-->--o2	15o-->--	AN0	Schritte
Sync	SW1	-->--o3	14o-->--	AN1	Richtung
		o4	13o		
		o5	12o		
		o6	11o		
		o7	10o		
Masse	GND	-----o8	9o		

### Justierung der Mechanik

Der optimale Abstand Bildvorlage zu Sensor ist von der Blende des Sensors abhängig und muß deshalb experimentell ermittelt werden. Um diesen, einmal festgestellten Abstand dauerhaft einhalten zu können, wurde die Abtasteinheit justierbar ausgelegt. Die Justage ist recht einfach und erfordert lediglich ein Universal-Meßinstrument und einen Schraubendreher. Wir befestigen mit Klebeband ein weißes, mit feinen Linien (<1mm, >0,3mm Stärke) bemaltes Blatt Papier auf unserer Trommel, bauen unseren Scanner auf und richten die Abtasteinheit so aus, daß deren Arme genau ins Zentrum der Trommel zeigen. Die Anschlüsse des Fototransistors werden von der Steuerplatine abgezogen, mit dem Meßinstrument (Widerstandsmessung) verbunden und die Elektronik mit der externen Versorgungsspannung in Betrieb gesetzt. Wenn jetzt der Sensor auf eine weiße Fläche gerichtet ist, so wird das Meßinstrument einen Widerstand anzeigen, der sich durch Ändern des Koppler-Bild-Abstandes (gesamten Turm verschieben) auf einen minimalen Wert einstellen läßt. Ist diese Position gefunden, wird der Abstandhalter so festgestellt, daß dessen Kugellager die Trommel berührt.

Die Empfindlichkeit der Schaltung wird mit dem Wendeltrimmer eingestellt. Hierzu wird der Fototransistor wieder mit der Schaltung verbunden und mit dem Meßinstrument die Spannungsänderung am Bildsignalausgang (Pin 19) überprüft. Bei langsamer Drehung der Trommel sollen die schwarzen Linien einen eindeutigen Spannungssprung hervorrufen.

Zur Zeilensynchronisation muß pro Trommelumdrehung an Pin 20 (Synchron-Ausgang) ein einwandfreier Spannungssprung zu messen sein. Durch Verschieben des Relais-Armes wird der optimale Abstand des Relais zum Magneten gesucht.

Das Basic-Programm wurde so ausgelegt, daß sofort nach dem Öffnen des Kontaktes die Bildabtastung beginnt. Eine Markierung auf der Trommel erleichtert uns später das Auswechseln der Vorlage.

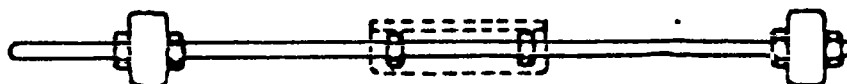


Bild 3: Gewindebuchse auf Gewindestange

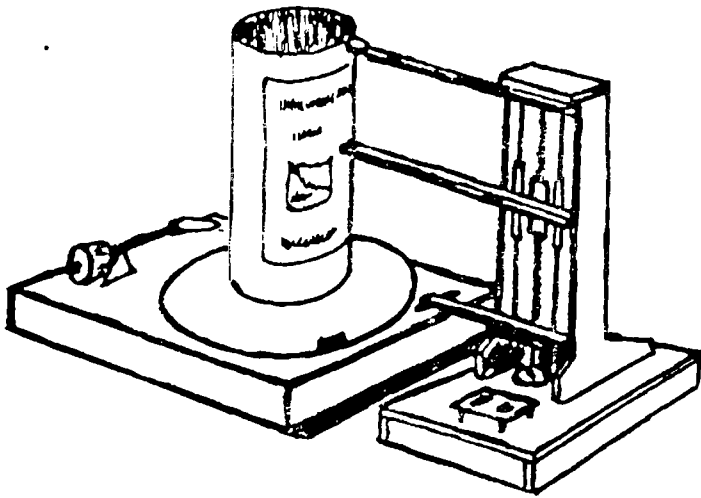


Bild 1: Gesamtanordnung

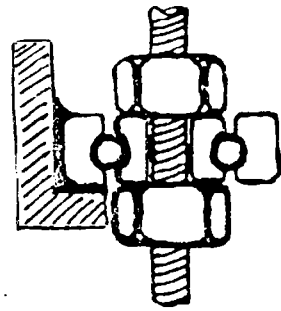


Bild 2: Kugellager mit Gewindestange

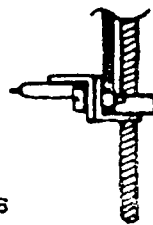
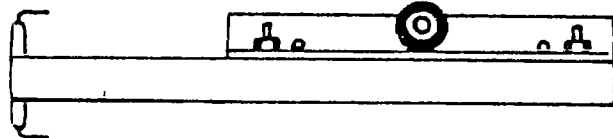
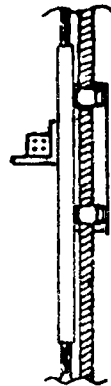
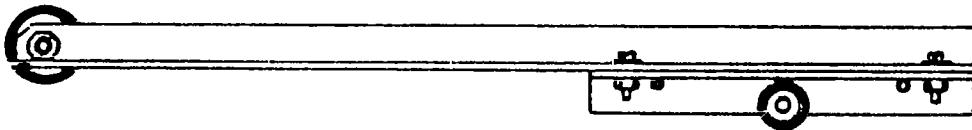


Bild 4: Die Arme des Scanners

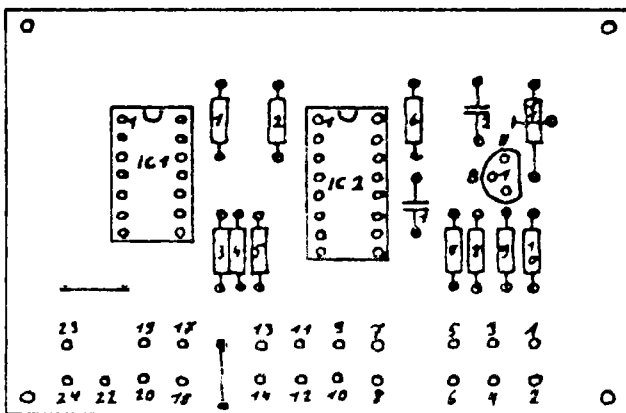
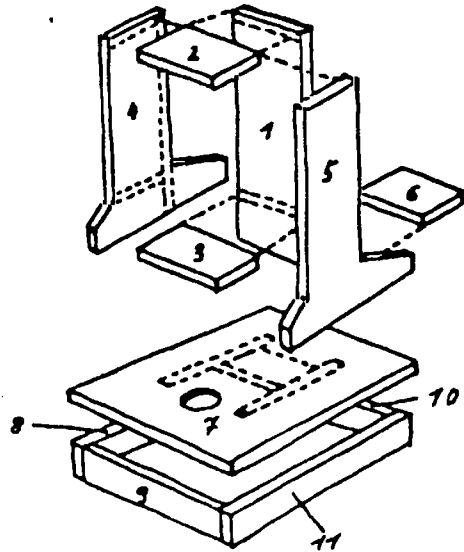
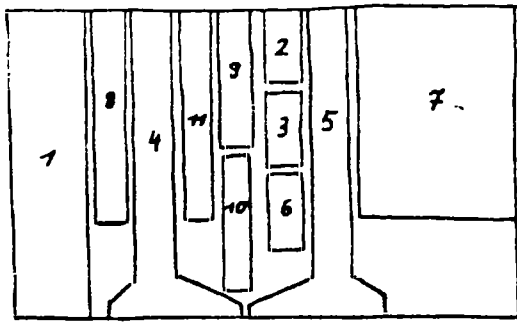


Bild 8: Bestueckungsplan fuer die Platine



1	Rückwand	41 x 10	cm
2	Deckel	10 x 5	cm
3	Zwischenstück	10 x 5	cm
4	Seite rechts	41 x 5/17	cm
5	Seite links	41 x 5/17	cm
6	Stabilisierung	10 x 5	cm
7	Boden	28 x 20	cm
8+11	Füße	28 x 4	cm
9+10	Füße	18 x 10	cm

Bild 5: Aussage-Skizze und Explosionszeichnung des Turmes

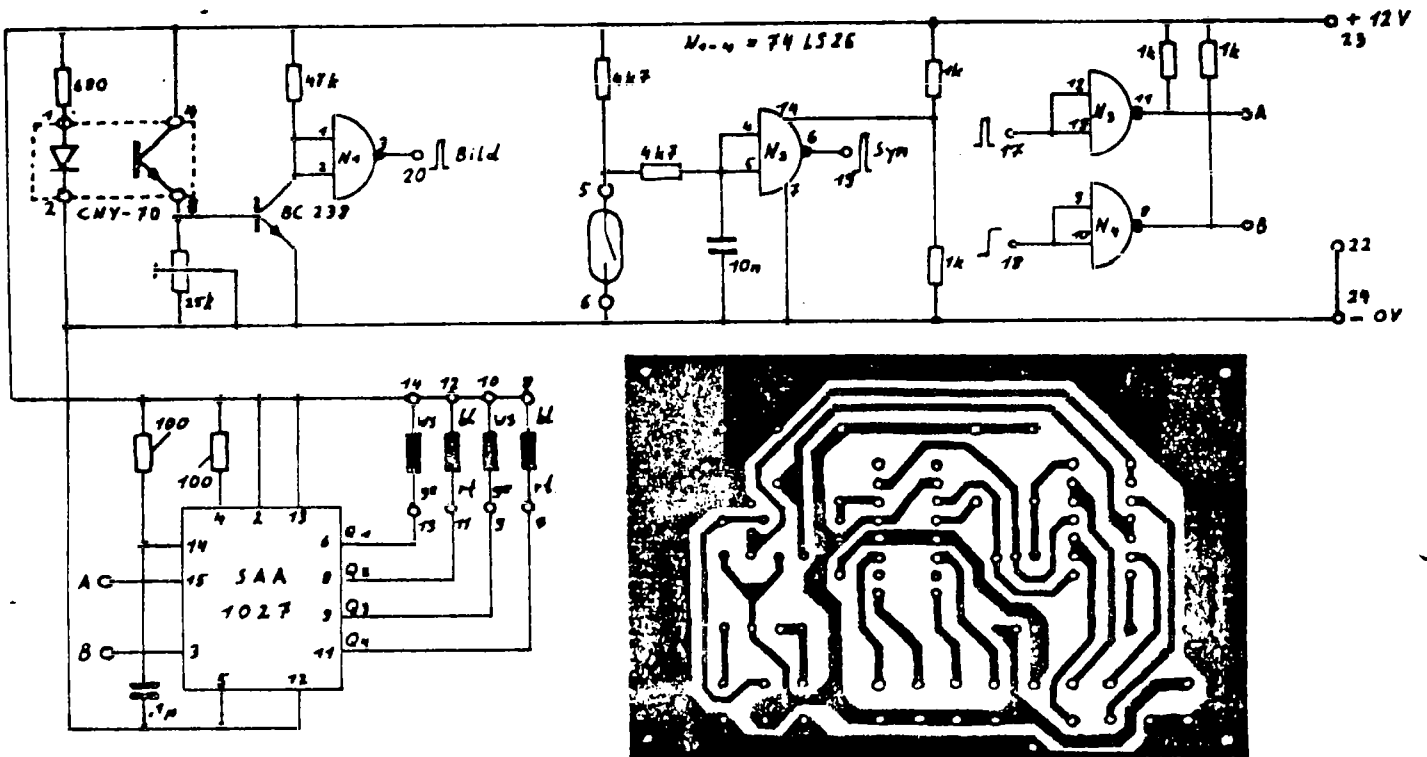


Bild 7: Schaltung und Platinenlayout

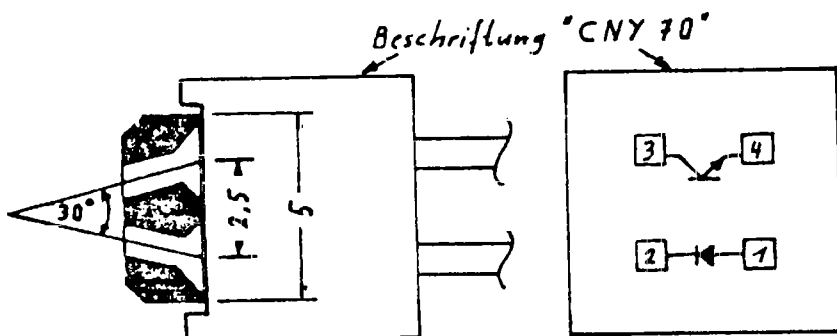


Bild 6: Reflexkoppler mit Blende

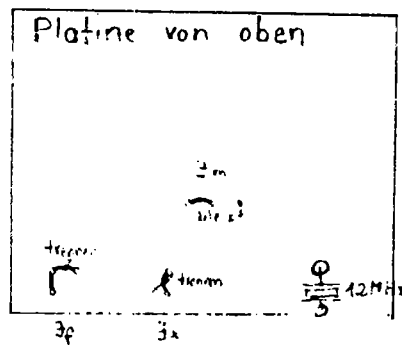
10/116



Ergegnung zur Grafikkarte GDP 64K

Der Umbau zum Auslesen des Grafikspeichers bei der GDP64K ist ja in sofern ein gewisses Risiko, als das er nicht bei jedem EF 9366 funktioniert. Schuld daran ist das Timing des MFREE\* - Signales. Wesentlich besser dafuer geeignet ist der Prozessor EF 9367. Er ist dafuer aber auch einige Mark teurer. Auf seine sonstigen Eigenschaften moechte ich hier aber nicht eingehen.

Rolf Dieter Klein (Entwickler der GDP) hat ein Buch mit dem Titel "Die Prozessoren 68000 und 68008" herausgebracht, indem er auch beschreibt, wie man den EF 9367 auf der GDP einsetzt. Als erstes muss man den 14 MHz Quarz durch ein 12 MHz Quarz ersetzen. Dieses fuehrt dazu, dass man jetzt einen rechteckigen Bildschirm besitzt. Weiterhin muessen die drei Jumper, die sich auf der Platine befinden, nach folgendem Bild geaendert werden :



Ich selber habe den Einsatz des EF 9367 nicht ausprobiert, da bei meiner Grafikkarte das Auslesen des Speichers auch ohne Probleme mit dem EF 9366 funktioniert.

Ich wuensche Euch gutes Gelingen.

Joerg Seelmann-Eggebert

\*\*\*\*\*

DFU - News:  
6.04.87

Post ändert Gebühren Ordnung im Datex - P Netz.

Ab 1.04.87 wurden die Gebühren im Datex-P-Netz neu geordnet. War es bisher möglich zum normal Tarif die nächsten Datex P Knotenpunkt zu erreichen, sprich für 23 Pfennig im Acht Minuten Takt, wurden die Preise im neuen Quartal um den faktor 10 auf 23 Pfennig für 50 Sekunden erhöht. Im Gegenzug dafür bietet die Post die Möglichkeit, Bundeseinheitlich die gleiche Nummer für den Datex P Zugang an. Ab 1.4.87 kann also aus dem gesamten Bundesgebiet unter der Nummer 01 34 11, der Zugang zu Datex-P geöffnet werden. Es bleibt jedem selbst überlassen, sich Gedanken darüber zu machen, ob diese möglichkeit die drastische Erhöhung der Gebühren rechtfertigt.

(GELOS)

\*\*\*\*\*

## Das "Clubleben"

---

Auch ich habe mir in der letzten Zeit immer mehr Gedanken ueber unseren Club gemacht.

Wir stehen in absehbarer Zeit sicherlich vor dem Problem, dass unsere Rechner langsam aber sicher aussterben.

Aber darauf moechte ich jetzt nicht naeher eingehen, denn dieses ist schon in der vorherigen Info durch Helmut Bernhardt erfolgt (wobei ich mich ihm voellig uebereinstimme, dass eine Fusion das Sinnvollste ist, um den immer kleiner werdenden Kreis der Genie und TRS 80 Anwender zusammenzuhalten und mit moeglichst vielen Informationen zu versorgen).

Ich aber sehe noch ein ganz anderes Problem :

Wenn man mal das Inhaltsverzeichnis 1986 mit einer aktuellen Mitgliederliste vergleicht, so sollte doch jeder mal etwas nachdenklich werden.

Es handelt sich hier naemlich in keiner Weise um einen Club, sondern vielmehr um einen "Zeitschriftenverlag", in dem bestimmte Leute (professionelle) Artikel schreiben und der Rest der Mitglieder lediglich seine 3.50 DM fuer eine billige, auf seinen Computer bezogene Zeitschrift zahlt (ueber 50% der Mitglieder sind lediglich in der Mitgliederliste vertreten).

Dieses spiegelt sich dann auch in den Infos wieder. Denn fruehere Infos wurden von wesentlich mehr Leuten erstellt, und sie bestand auch aus viel mehr kleinen und dennoch interessanten Beiträgen (Probleme, Tips und Tricks, Fragen ...)

In Laufe der Zeit hat es sich immer mehr zu einer Fachzeitschrift entwickelt, aber wohl kaum deshalb, weil alle Clubmitglieder entsprechend dazugelernt haben.

Viele Artikel, die momentan im Info erscheinen, sind fuer einen Teil der Mitglieder ziemlich uninteressant, da oft noch die noetigen Kenntnisse und Grundlagen fehlen.

Nur versteht mich jetzt nicht falsch : Dieses soll KEIN Vorwurf gegen die jetztigen Artikelschreiber sein, denn diese Artikel sind hochinteressant und ich hoffe, dass sie auch weiterhin im Info erscheinen (die 256K Erweiterung z.B. ist wohl die beste Schaltung seit der Entwicklung des TRS 80/Genie).

Ich moechte hiermit vielmehr an die Leute appellieren, die noch keinen Artikel fuer das Info geschrieben haben.

Es gibt vermutlich zwei Gruende, warum von einigen keine Artikel geschrieben werden.

Ein Teil derer (naemlich diejenigen, die schon ein bestimmtes Wissen ueber ihren Computer besitzen), schieben als Grund vor, dass ihnen die Ideen fehlen oder dass sie keine Zeit haben. Wuerde jeder von denen, auf die das zutrifft, ab und zu ein bisschen in seinen Disketten, Literatur und grauen Zellen herumwuehlen, so wuerde er spaetestens in 10 Minuten soviel brauchbares Material gefunden haben, um die naechsten drei Infos alleine zu gestalten. Denn bekanntlich macht auch Kleinvieh Mist.

Weiterhin gibt es im Club aber auch einen Teil, der sich auf den Standpunkt stellt : "ich bin Einsteiger und mir fehlen noch die noetigen Grundkenntnisse - wie soll ich da etwas Produktives und fuer die Allgemeinheit Interessantes in das Info setzen".

Gerade aber diese Gruppe koennte das Info beleben, indem sie

- Fragen stellt ueber Dinge, die man wissen moechte oder aber nicht verstanden hat
- Probleme vortraegt, die bei bestimmten Programmen aufge-

treten sind

- und sagt (bzw schreibt), ueber welche Bereiche sie gerne mehr erfahren und lernen moechte

In diesem Fall kann das Info auch wieder fuer jeden interessant werden.

Angst, dass man sich blamieren oder blossstellen koennte, ist hier mit Sicherheit fehl am Platz. Denn es ist ja der Sinn und die Aufgabe eines solchen Clubs, sich gegenseitig zu unterstuetzen und zu helfen. Ausserdem hat jeder mal klein angefangen und die erfahrenen Leute mit Fragen geloechert.

Wenn Ihr noch Interesse am Club habt, dann nehmt Euch mal den von mir verfassten Artikel zu Herzen.

Denn Mitglied in Club zu sein, heisst nicht nur jeden Monat die Ueberweisung auszufuellen, sondern sich aktiv am Geschehen im Club zu beteiligen !

Denkt auch mal an die paar Mitglieder im Club, die viele Stunden ihrer Freizeit opfern, dass Ihr ein interessantes Info rechtzeitig erhaltet.

Joerg Seelmann-Eggebert

#### Ein Praxistip fuer Bastler

Oft hat man die Moeglichkeit, billig an irgendwelche Platinen heranzukommen, die mit allen erdenklichen Spezialitaeten bestueckt sind.

Widerstaende, Dioden usw. sind innerhalb weniger Minuten entfernt und auch Transistoren baut man durch geschicktes Biegen und Wackeln relativ schnell unbeschaedigt aus.

Problematisch wird es dann aber bei unseren vielbeinigen Freunden, denn dort ist man mit einem einfachen Loetkolben aufgeschmissen. Und oft sind diese aber gerade die Leckerbissen auf der Platine.

Eine Methode, die ICs zu entloeten ist folgende :

Man saegt mittels einer Laubsaege die ICs aus (ca 1 cm Rand lassen). Gleichzeitig erhitzt man in einem kleinen Blechschaelchen (Durchmesser ca 10 cm) Zinn bzw. ein Zinn/Bleigemisch. Die Schale und das Zinn gibt es in einem Hobby oder Spielwarengeschaeft fuer ca 10 DM zu erwerben.

Dort schmeisst man dann die Platinenschnitzel hinein und zieht das IC nach 1 bis 2 Sekunden aus der Platine heraus. Auf diese Art habe ich schon viele ICs entloetet (und zwar intakt).

Joerg Seelmann-Eggebert

# Paul Kröher

PAUL KRÖHER.

Genie/TRS80 User Club Bremerhaven

Ihr Schreibens	Ihr Zeichen	Mein Schreibens	Mein Zeichen	Datum
-	-	-	Kr.	06.04.1987

Betr.: Für Maschinenprofis und solche die es werden wollen

Liebe Clubkameraden,

seit langer Zeit beschäftige ich mich (wie vielleicht viele von Euch) mit Maschinenroutinen. Dabei wird fast dauernd in den hervorragenden Erläuterungen von Röckrath Mikrocomputer geblättert. So nach und nach werden das ROM und RAM Betriebssystem verständlich. Nur man ist ja kein Lexikon und muß deshalb immer wieder die Nachschlagewerke zu Hand nehmen, um z.B. eine bestimmte Adresse nutzen zu können. Im Laufe der Zeit habe ich mir eine handliche Kurzfassung der einzelnen Routinen zusammengestellt. In den meisten Fällen -insbesondere wenn man die ausführlichen Erläuterungen in den Veröffentlichungen von Röckrath kennt- reicht diese Kurzfassung aus um praktisch arbeiten zu können.

So nach und nach wurde diese Kurzfassung fertiggestellt. Sie muß immer noch nicht komplett sein. Trotzdem will ich sie Euch zur Verfügung stellen. Da ich dabei viele Erklärungen und Funktionsnamen aus den Veröffentlichungen von Röckrath Mikrocomputer verwendet habe (dieser hat dafür das Copyright) mußte ich die Erlaubnis von Röckrath Mikrocomputer einholen. Diese ist erfreulicherweise eingetroffen (siehe Anlage).

Ich hoffe das Ihr die Kurzfassung bei der täglichen Arbeit am Computer genau wie ich gut gebrauchen könnt. Wenn Euch noch neue Routinen bekannt werden informiert mich bitte, damit diese Liste ergänzt werden und zu gegebener Zeit berichtigt wieder im Info stehen kann.

Viel Spaß beim Programmieren  
wünscht Euer Clubkamerad

*Paul*

Unterprogramme und sonstige Adressen im TRS80/Video Genie System ROM  
 Zusammenestellt von Paul Krocher, Karpfenweg 6, 2970 Emden Tel.:(04921) 27707  
 Mit freundlicher Genehmigung von Röckrath Mikrocomputer. Kommerzielle Verwendung verboten!

Diese Liste stellt nur eine handliche Übersicht dar.  
 Näheres über die einzelnen Routinen findet der Interessierte Leser in den Veröffentlichungen von  
 Röckrath Mikrocomputer, Kaiserstr. 54, 4050 Mönchengladbach 1 :

TRS-80 und Video Genie ROM-Listing für Level II

Maschinensprachebuch ISBN 3-925074-07-4

Das DOS Buch ISBN 3-925074-10-4

Diese Bücher sind ein unentbehrliches Arbeitsmittel für denjenigen der sich mit TRS-80 und Video Genie Computer befaßt.

Die Angaben müssen nicht vollständig sein. Der Anwender sollte vor Benutzung der Adressen selber prüfen was im Einzelnen passiert.

Für ergänzende Hinweise bin ich dankbar.

Name	Adr.	Beschreibung	beeinflußt	Basic	Input	Output
				notwendig		
	0008	RST 08 / Vergleiche mit nachf. Symbol Sprung via RAM Vector 4000 nach 1C96				
HERE	000B	ermittelt Stand des PC (POP HL / JP (HL))		HL	-	HL=Zeiger
	0010	RST 10 / hole nächstes Symbol Sprung via RAM Vector 4003 nach 1D78				
READB	0013	liest ein Byte aus einem File	AF		DE=Zeiger auf FCB	wenn Z=1 A=nächstes Byte sonst A= Fehlercode
	0018	RST 18 / vergleiche DE/HL Sprung via RAM Vector 4006 nach 1C90				
WRITEB	001B	schreibt ein Byte in ein File	AF		DE=Zeiger auf FCB A=Byte	ggf. A=Fehlercode
	0020	RST 20 / Sprung via RAM Vector 4009 nach 2D59				
	0028	RST 28 / Sprung via RAM Vector 400C nach 2D??				
INCH	002B	Tastaturabfrage	AF	DE		A=ASCII gedr.Taste oder 0
	0030	RST 30 / Sprung via RAM Vector 400F nach 2D??				
OUTCH	0033	Ausgabe eines Zeichens auf den Bildschirm	F	DE	A=auszugebendes Zeichen	
	0038	RST 38 / Sprung via RAM Vector 4012 nach 2D??				
PRINT	003B	Ausgabe eines Zeichens auf den Drucker	F	DE	A=auszugebendes Zeichen	
INCH	0049	Tastaturabfrage wartet bis Taste gedrückt	AF	DE		A=ASCII gedr. Taste
	0050	Tastaturcodiertabelle Control-Tasten				
DELAY-	0060	Zeitschleife 14,66 usek * BC	AF BC		BC	-
CLS	01C9	Bildschirm löschen	AF		-	(40A6 = Cursorposition)
PULSE	01D9	gibt Impuls auf Kassette aus	AF B	HL	-	-
CASOFF	01F8	Kassettenrecorder ausschalten	AF		-	-
CASND	01FE	decodiert Kassettenrec.Nr. und schaltet ein	AF BC DE HL		???	???
CASON	0215	Kassettenrecorder einschalten	AF		-	-
RESINP	021E	setzt den Eingabeport Bit7 von Port 255 zur.	AF	HL	-	-
PORTCT	0221	steuert Port 255	AF		H = zu löschende Bits L = zu setzende Bits	(403D) = neuer Fortstatus
BLINK	022C	schaltet Stern in Bildecke um	AF		-	-
READC	0235	liest Byte von Kass.	AF		-	A=zugelesenes Byte
READCI	0241	liest Bit von Kass.	AF		A=Byte	A=Byte * 2 + Bit
WRBTW	0261	schreibt Byte zweimal auf Kass.			A=aufzuzeichnendes Byte	-
WRITEC	0264	schreibt Byte auf Kass.			A=aufzuzeichnendes Byte	-
INITW	0284	Kass. an + Synchronbytes aufzeichnen	AF BC DE HL			
WRITESY	0287	schreibt Synchronbytes	AF B		-	-
INITR	0293	Kass. an + Synchronbytes lesen	AF BC DE HL		HL=Pointer auf Kass. Nr.	

READY	0296	liest Synchronisation	AF	-	-
READW	0314	liest Wort (LSB,MSB) von Kass.	AF	-	HL=Wort
OUTCHF	032A	Ausgabe eines Zeichens auf BS LP oder Kass.	F	ja	A=auszugebendes Zeichen (409C)=Ausgabeflag 0=BS 1=LP 30=Kass. A=auszugebendes Zeichen DE gerettet
OUTCHC	033A	Ausgabe eines Zeichens auf den Bildschirm	F		DE gerettet
CURPOS	0348	ermittelt Cursorposition in der Zeile	AF		A = Cursorposition
INCH2	0358	Tastaturabfrage	AF		wie INCH (DE gerettet)
INLINE	0361	Eingabe von max 240 Z. in I/O Buffer	AF	DE HL ja	HL=I/O Bufferanfang
INCHW2	0384	Tastaturabfrage wartet bis Taste gedrückt	AF		wie INCHW (DE gerettet)
RESPR	0368	Ausgabeflag auf Display und CR an Drucker wenn Druckkopf nicht in Position 0	AF		(409C)=Ausgabeflag (409B=PTRPOS)
PRCF	0394	CR auf Drucker ausgeben	A		-
PRINT!	039C	Ausgabe eines Zeichens auf dem Drucker			A=auszugebendes Zeichen (PRTPOS) incrementiert
CALL	0102	Ruft I/O-Routinen über DCB auf	?? ?? ?? ??		BC auf Stack (B=DCB-Typ)
CRZLA	04A1	Cursor zum Zeilenanfang			-
CHOME	04DA	Cursor Home			-
CDOWN	04E7	Cursor eine Zeile runter			-
SCROLL	054F	Bildschirmscroll			-
CLEOL	0573	bis Zeilenende löschen			-
PRRDY	05D1	testet ob Drucker bereit	AF		Z=1 wenn bereit
INBUFF	05D9	Eingabe einer Zeile in den Buffer	AF BC DE		B=max.Eingabeanzahl HL=Zeiger auf Buffer B=Anzahl eingebener Z. C=B bei Eintritt (max.Z.) HL=Zeiger auf Bufferanfang
TAB	0646	Tabulator			A=Anzahl Space
	06CC	BASIC Warmstart			
SADD2	0708	Singleaddition $Z=Z1+0,5$	AF BC DE HL		(FACX) (FACX)
SADDC	070B	Singleaddition	AF BC DE HL		(FACX) + (HL) (FACX)
SSUBC	0710	Singlesubtraktion	AF BC DE HL		HL=Zeiger (HL)-(FACX) (FACX)
SSUBY	0713	Singlesubtraktion	AF BC DE HL ja		BCDE - (FACX) (FACX)
SADDY	0716	Singleaddition	AF BC DE HL ja		BCDE + (FACX) (FACX)
RNUL	0778	reelle Zahl = 0	AF		- (FACX)
RUND1	07A8	Rundung?			
SMADDY	07B7	Singlemantissenaddition	AF C DE HL	???	???
SRYR	07D7	schiebt Singlezahl um A Bits rechts	AF BC L	A=Anzahl ???Zahl	???
LOG	0809	berechnet natürlichen Logarithmus	AF BC DE HL ja		(FACX) (FACX)
SADDY	0816	Singleaddition	AF BC DE HL ja		BCDE + (FACX) (FACX)
SMLTL2	0841	Singlemultiplikation mit LOG(2)	AF BC DE HL		(FACX) (FACX)
SMLTY	0847	Singlemultiplikation	AF BC DE HL ja		BCDE * (FACX) (FACX)
SDIV10	0897	Singledivision durch 10	AF BC DE HL		(FACX) (FACX)
SDIVY	08A2	Singledivision	AF BC DE HL ja		BCDE / (FACX) (FACX)
SMLT10	092E	Singlemultiplikation mit 10	AF BC DE HL		(FACX) (FACX)
RTESTX	0955	testet reelle Zahl	AF		(FACX) wenn $x < 0: A=FF CY=1 S=1$ $x = 0: A=00 Z=1 P=1$ $x > 0: A=01$
_DATA	0964	8-Bit-Zahl mit Sign in Single umwandeln	AF BC DE HL	???	???
ABS	0977	Absolutwert einer Zahl bilden	AF BC DE HL ja		(FACX) (FACX)
NEG	097B	negativen Wert einer Zahl bilden	AF BC DE HL		(FACX) (FACX)
RNEG	0982	negativen Wert einer reellen Zahl bilden	AF HL		(FACX) (FACX)
SGN	098A	bildet die SGN-Funktion einer Zahl	AF BC DE HL ja		(FACX) (FACX)
INTA	098D	wandelt A in 16-Bit Integer um mit Sign	AF HL	A=Zahl mit Vorz. Bit 7	(FACX)
TESTX	0994	testet alle numerischen Typen	AF HL		(FACX) siehe 0955h
ITESTX	099B	testet Intergerzahl	AF HL	Zahl in (FACX)	wenn $(0:A=FF CY=1 S=1$ $=0:A=00 Z=1 P=1$ $>0:A=01$
SPUSHX	09A4	rettet Single auf Stack	DE	(FACX)	SP
SLDXC	09B1	lädt Single Konstante	BC DE HL	HL=Zeiger auf Konstante	???
BLDXY	09B4	transportiert Single Zahl	DE	???	???
SLDXY	09BF	transportiert Single Zahl	BC DE HL		
BLBYC	09C2	lädt Konstante	BC DE HL	HL=Zeiger auf Konstante	BCDE
SLDCX	09CB	speichert Zahl als Konstante	AF B DE HL	(FACX) HL=Zeiger Speicher	(HL)=Zahl

MOVE	09D2	transportiert Wert jeden Typs	AF B DE HL	HL=Quellezeiger DE=Zielzeiger	(DE)=Wert
MOVE1	09D3	wie MOVE bei vertauschten Registern			
LDMY	09F4	speichert Wert um (anschl. MOVE aufrufen)	AF B DE HL	(FACX)	(FACX)
_LDM	09F7	speichert Wert um (HL) => (FACX)	AF BC DE HL	HL=Zeiger	(FACX)
LDMY	09FC	speichert Wert um (anschl. MOVE1 aufrufen)	AF B DE HL	(FACX)	(FACX)
POINTX	0A03	ermittelt X-Achse in Abhängigkeit von TYPX	AF DE		DE=X-Adresse
SCPXY	0A0C	Singlevergleich Z1 und Z2	AF HL	???	Z1 > Z2 : A=1 Z1 = Z2 : A=0 Z=1 Z1 < Z2 : A=FF CY=1 S=1
ICP	0A39	Integervergleich Z1 und Z2	AF	Z1 = HL / Z2 = DE	Z1 > Z2 : A=1 Z1 = Z2 : A=0 Z=1 Z1 < Z2 : A=FF CY=1 S=1
DCPXC	0A49	Doublevergleich Z1 und Konstante	AF BC DE HL	Z1 = (FACX) DE = Zeiger auf Konst.	Z1 : Konst. : A=1 Z1 = Konst. : A=0 Z=1 Z1 < Konst. : A=FF CY=1
DCPXY	0A4F	Doublevergleich Z1 und Z2	AF BC DE HL	Z1=(FACX) / Z2=(FACX)	-wie DCPXC-
DCPXY	0A7B	Doublevergleich	AF BC DE HL	Z1=(FACX) / Z2=(FACX)	(<):Invertierung DCPXC
SINT	0A7F	wandelt Zahl in 16 Bit Integer	AF BC DE HL ja	(FACX)	(FACX)
ILDHL	0A9A	lädt Zahl als Integer nach FACX	A	HL=Zahl	(FACX)=Zahl
TYPXI	0A9D	TYPX = Integer	A	-	(TYPX) = 2
CS'	0AB1	wandelt Zahl in Single	AF BC DE HL ja	(FACX)	(FACX)
ICSNGX	0ACC	Integer in Single umwandeln	AF BC DE HL	(FACX)	(FACX)
ICSNGH	0ACF	Integer in Single umwandeln	AF BC DE HL	HL=Zahl	(FACX)
2DBLX	0ADB	Zahl in Double umwandeln	AF BC DE HL ja	(FACX)	(FACX)
TYPXD	0AEC	TYPX = Double	A BC	-	(TYPX) = 8
TYPXS	0AEF	TYPX = Single	A	-	(TYPX) = 4
	0AF4	testen ob String in X			
FIX	0B26	INT ohne Vorzeichen	AF BC DE HL ja	(FACX)	(FACX)
INT	0B37	ermittle nächst kleinere ganze Zahl	AF BC DE HL ja	(FACX)	(FACX)
INDEX	0BAA	Multiplikation (für Feldverwaltung)	AF DE	BC * DE	DE
ISUB	0BC7	Integersubtraktion	AF BC DE HL ja	DE-HL	(FACX) + HL wenn Integer
IADD	0BD2	Integeraddition	AF BC DE HL ja	DE + HL	(FACX) + HL wenn Integer
IMLT	0BF2	Integermultiplikation	AF BC DE HL ja	DE * HL	(FACX) und HL wenn INT
INES	0C5B	negativer Wert einer Integerzahl	AF BC DE HL	(FACX)	(FACX)
DSUB	0C70	Doublesubtraktion	AF BC DE HL ja	(FACX) - (FACX)	(FACX)
DADD	0C77	Doubleaddition	AF BC DE HL ja	(FACX) + (FACX)	(FACX)
RUND2	0D20	Rundung?		???	???
DMADD	0D33	Doublemantissenaddition	AF C DE HL	???	???
DMSUB	0D45	Doublemantissensubtraktion	AF C DE HL	???	???
RL'	0D6A	8 Speicherstellen rechts schieben	AF C DE	A=Anzahl der Verschieb. HL=Zeiger	
RRX	0D90	Speicherbereich um 1 Bit rechts schieben	AF DE HL	(FACX)	(FACX)
RLM	0D97	Speicherbereich um 1 Bit links schieben	AF C HL	HL=Zeiger auf Speicherb. CY=nachzuschiebendes Bit	
DMLT	0DA1	Doublemultiplikation	AF BC DE HL ja	(FACX) * (FACX)	(FACX)
DDIV10	0DDC	Doubledivision :10	AF BC DE HL	(FACX)	(FACX)
DDIV	0DE5	Doubledivision	AF BC DE HL ja	(FACX) : (FACX)	(FACX)
DMLT10	0E4D	Doublemultiplikation *10	AF BC DE HL	(FACX)	(FACX)
DVAL	0E65	wandelt String in Doublezahl	AF BC DE HL	HL=Zeiger auf String	(FACX)
LVAL	0E6C	wandelt String in Zahl passenden Typs um	AF BC DE HL ja	HL=Zeiger auf String	(FACX) (TYPX)
CREAL	0EFB	wandelt Zahl in Single oder Double	AF	Z = 1 Umwandlung Single	(FACX)
RMLT10	0FOA	multipliziert reelle Zahl mit 10	AF BC DE HL	(FACX) Z=0	(FACX)
RDIV10	0F18	dividiert reelle Zahl durch 10	AF BC	(FACX)	(FACX) A+1
SADDA	0F89	Summe bilden	AF BC DE HL	(FACX) + A	(FACX)
PRZNR1	0FA7	wie PRZNR mit vorangestelltem Wort "IN"	AF BC DE HL ja	HL=auszugebende Zahl	-
PRZNR	0FAF	Integerzahl ohne Vorzeichen ausgeben (ZNR)	AF BC DE HL ja	HL=auszugebende Zahl	-
JSTR	0FBD	wandelt Zahl in unformatierten Sting um	AF BC DE HL ja	(FACX) (TYPX)	HL=Zeiger auf String (FACX) verändert
FSTR	0FBE	wandelt Zahl in formatierten String um	AF BC DE HL ja	(FACX) = Zahl A = Formatflag	HL = Zeiger auf Zeichenfolge

						B = Vorkommastellen C = Nachkommastellen A = Anzahl 0 HL = Bufferzeiger		HL um Anzahl erhöht
FILLO	1269	Nullen in Buffer einschreiben	AF	HL				
SQR	13E7	bildet die Wurzel einer Zahl	AF BC DE HL	ja	(FACX)			(FACX)
POWER	13F2	berechnet Potenz einer Zahl	AF BC DE HL					
EXPO	1439	Exponentialfunktion einer Zahl bestimmen	AF BC DE HL	ja	(FACX)			(FACX) Funktionswert
REIHE1	149A	berechnet Reihe siehe ROM Listing	AF BC DE HL					
REIHE2	14A9	berechnet Reihe siehe ROM Listing	AF BC DE HL					
RND	14C9	Erzeugung einer Zufallszahl	AF BC DE HL	ja	(FACX)=Bereich			(FACX)
RRND	14F0	reelle Zufallszahl erzeugen	AF BC DE HL		-			(FACX)
COS	1541	Cosinusfunktion berechnen	AF BC DE HL	ja	(FACX)			(FACX)
SIN	1547	Sinusfunktion berechnen	AF BC DE HL	ja	(FACX)			(FACX)
TAN	15A8	Tangensfunktion berechnen	AF BC DE HL	ja	(FACX)			(FACX)
ATN	15BD	Arcustangens-Funktion berechnen	AF BC DE HL	ja	(FACX)			(FACX)
	1997	Syntax-ERROR						
LPNT	1AFB	Zeilenzeiger im ganzen Programm erneuern	AF	DE HL	ja			
LPNTP	1AFC	Zeilenzeiger teilweise erneuern	AF	DE HL	ja	DE=Beginnadresse		
LAPG	1B10	Analysiert zwei Argumente für LIST	AF BC DE HL					
SLINE	1B2C	sucht Zeile im Programtext	AF BC DE HL	ja	DE=ZeNR gs. Ze			siehe S. 19 Röckrath
INLINA	1BB3	Gibt ? aus und erwartet Eingabe max. 240 Z.	AF	DE HL				HL=I/O Bufferanfang
ENCTXT	1BC0	erzeugt Zwischencode (Token)	AF BC DE HL	ja	HL=Zeiger auf Textanfang			BC = Länge Zw.code + 5 HL = Zeiger vor Zw.code entspricht IOBUFF -3
CPHLDE	1C90	RST 18 - Unterprgr. vergleicht HL + DE	AF			HL + DE = 16 Bit - Integer ohne Sign		HL > DE : Z=0 CY=0 HL = DE : Z=1 CY=0 A=0 HL < DE : Z=0 CY=1
CHECK	1C96	RST 08 - Unterprgr. Syntaxcheck	AF	HL	ja	HL=Pointer		siehe S. 45 Röckrath
CHRGET	1D78	RST 10 - Unterprgr.	AF	HL		HL=Pointer		siehe S. 45 Röckrath
	1DBE	Break in INPUT						
LETTER	1E3D	testet ob Zeichen ein Buchstabe	AF			HL=Zeiger auf Zeichen		wenn Buchst.CY=0 sonst 1
INTEXP6	1E45	Ausdruck auswerten, Ergebnis Ganzzahl<32768	AF BC DE HL			HL=Zeiger auf Ausdruck		DE=Ergebnis
	1E4A	FC - ERROR						
DECLND	1E4F	String in Ganzzahl wandeln ( < 65530 )	AF	DE HL		HL = Zeiger auf String		DE = Zahl
SEOS	1F05	sucht Ende der Anweisung	AF BC D HL					
SEOL	1F07	sucht Zeilenende	AF BC D HL					
NEXTL	20F9	CR ausgeben, wenn Cursor nicht in Pos.0	AF			(409C)=Ausgabeflag (40Ab)=Curs.pos.		
CR	20FE	CR - ausgeben	AF			(409C) = Ausgabeflag		
EXPB	2335	Ausdruck auswerten der mit Klammer beginnt	AF BC DE HL			HL=Zeiger auf Ausdruck		(FACX) / Typ (TYPX)
EXP	2337	Ausdruck auswerten	AF BC DE HL	ja		HL=Zeiger auf Ausdruck		(FACX) / Typ (TYPX)
IDIV	2490	Integerdivision	AF BC DE HL	ja		DE / HL		(FACX) + HL wenn Integer
OPERND	249F	Operanden für Ausdruckanalyse auswerten	AF BC DE HL			HL=Zeiger auf Ausdruck		(FACX)=Ergebnis
TYPCHK	25D9	RST20 Typstest (FACX)	AF			Typ in (FACX)		siehe Röckrath
SVAR	260D	Variable in Tabelle suchen + ggf. einrichten	AF BC DE HL	ja		HL=Zeiger auf Var.name		siehe S.24 Röckrath
ILDXA	27F8	Zahl als Integer ohne Sign nach FACX	AF	HL		A=Zahl		HL + (FACX)=Ergebnis
CXTYPA	2819	Wert in gewünschten Typ umwandeln	AF BC DE			A = gewünschter Typ		???
STEMPS	2857	ermittelt Stringspaceadresse und legt ab	F BC DE HL			A=Stringlänge		DE=Stringadresse Str.spac HL=vorl. Zwischenspeicher Zw.sp.=Str.länge+Adr.(HL)
STRC	2865	St.konst. in Zw.sp. und (FACX) übernehmen	AF BC DE HL			HL=Zeiger auf String		(FACX)=Zeiger auf Zw.sp.
PRINTS	28A6	Druckt einen String (* oder 0 am Ende d.Str.)	AF BC DE HL			HL=Zeiger auf String		-
MROOM	28BF	Platz für String in Stringspace machen	F BC DE HL			A=Stringlänge		DE=Stringspaceadresse
SARCOL	28E6	Garbage collection	AF BC DE HL	ja		-		-
	298F	Stringaddition (Verknüpfung)						
INTEXP	2B01	Ausdruck auswerten, Ergebnis in Integer	AF BC DE HL	ja		HL=Zeiger auf Ausdruck		DE=Ergebnis
IEXPA	2B1B	Ausdruck auswerten Ergebnis in INT <256	AF BC DE HL	ja		HL=Zeiger auf Ausdruck		(FACX) Typ in (TYPX) A=Ergebnis E=Ergebnis
PRINTT	2B75	Druckt Textstring (0 am Ende d. Str.)	AF	HL	ja	HL=Zeiger auf String		-
GENTXT	2B7E	erzeugt aus Zwischencode lesbaren Text	AF BC DE HL	ja		HL=Zeiger auf Zw.code		max. 255 Z. im IOBUFF siehe S. 18 Röckrath



Unterprogramme und sonstige Adressen im TRS80/Video Genie System RAM  
 Teilweise erst nach Aufruf von BASIC/CMD vorhanden !!!  
 Zusammengestellt von Paul Kröher, Karpfenweg 6, 2970 Emden Tel.:(04921) 27707  
 Mit freundlicher Genehmigung von Röckrath Mikrocomputer. Kommerzielle Verwendung verboten!

Diese Liste stellt nur eine handliche Übersicht dar.  
 Näheres über die einzelnen Routinen findet der Interessierte Leser in den Veröffentlichungen von  
 Röckrath Mikrocomputer, Kaiserstr. 54, 4050 Mönchengladbach ! :  
 TRS-80 und Video Genie ROM-Listing für Level II  
 Maschinensprachebuch ISBN 3-925074-07-4  
 Das DOS Buch ISBN 3-925074-10-4  
 Diese Bücher sind ein unentbehrliches Arbeitsmittel für denjenigen der sich mit TRS-80 und Video Genie Computer befaßt.

Die Angaben müssen nicht vollständig sein. Der Anwender sollte vor Benutzung der Adressen selber prüfen was im Einzelnen passiert.

Für ergänzende Hinweise bin ich dankbar.

Adr.	Beschreibung			
RST08	4000 RST 08 Vektor JP 1C96			
RST10	4003 RST 10 Vektor JP 1D78			
RST18	4006 RST 18 Vektor JP 1D90			
RST20	4009 RST 20 Vektor JP 25D9			
RST28	400C RST 28 Vektor JP 4BC2 /SYS Files laden			
RST30	400F RST 30 Vektor JP 4609 DEBUG			
RST38	4012 RST 38 Vektor JP 45F2 INTERRUPT Routine			
KEYDCB	4015 Tastatur DCB Typ 01			
	4016 Treiberadresse 03E3 oder 4516 bei System AJ=J			
VIDDCB	401D Bildschirm DCB Typ 07			
	401E Treiberadresse 0458 oder 4505 bei System AJ=J			
	4020 Cursoradresse z.B.3C00			
	4022 Zeichen unter Cursor			
PTRDCB	4025 Drucker DCB Typ 06			
	4026 Treiberadresse 058D			
	4028 Zeilen pro Seite			
	4029 Zeilenzähler			
DOSRDY	402D Sprung nach DOS READY			
ERRRDY	4030 nach einem Fehler Sprung nach DOSRDY			
	4036 Tastaturstatusadresse für Tastaturzeile 3801			
	4037 Tastaturstatusadresse für Tastaturzeile 3802			
	4038 Tastaturstatusadresse für Tastaturzeile 3804			
	4039 Tastaturstatusadresse für Tastaturzeile 3808			
	403A Tastaturstatusadresse für Tastaturzeile 3810			
	403B Tastaturstatusadresse für Tastaturzeile 3820			
	403C Tastaturstatusadresse für Tastaturzeile 3840			
	403D Portstatus Port 255			
	4040 Beginn Uhrzeit und Datum			
	4049 HIMEM			
	404B -4050 RAM für DEBUG			
	4051 Zeiger auf Benutzer Routinen #name			
	405D -4062 RAM für DEBUG			
HEXDE	4063 DE hex nach (HL) ausgeben	AF	HL	DE + HL = Zeiger
HEXA	4068 A hex. nach (HL) ausgeben	AF	HL	A + HL (Zeiger)
ab	4080 27h Bytes ex 18F7... f. matemat. Funkt. (ohne Disk !)			
	408E USR-Unterprogramm Startadresse			
	4090 Multiplikator z.B. für RND			
	4093 wird als INF Unterprogramm angesprungen IN A,(..)			
	4094 speichert Portadresse			
	4096 wird als OUT Unterprogramm angesprungen OUT (...),A			
	4097 speichert Portadresse			

4099 INKEY% Zwischenspeicher letztes eingegebenes Zeichen  
 409A letzter Fehlercode  
 PRTPOS 409B Druckkopfposition  
 OUTFLG 409C Ausgabeflag 0=Bildschirm 1=Drucker 80=Kass.  
 409D Zeilenlänge auf Bildschirm  
 409E letzte TAB Position auf Bildschirm  
 SSPANF 40A0 enthält Anfang des Stringspace  
 40A2 aktuelle Zeilennr.  
 TXTANF 40A4 Programmtextanfang Pointer  
 CURSF 40A6 Cursorposition in der Bildschirmzeile / Cursorspalte  
 IOBUFF 40A7 enthält I/O Bufferadresse normal 41EB  
 40A9 Flag für Input (wenn 0 Kasette sonst Tastatur oder Data)  
 40AA Random Number Speicher  
 40AC Cursorposition in Bildschirmzeile  
 40AE DIM Flag (0 kein DIM)  
 TYPX 40AF enthält Typcode: 02 INT 03 String 04 Single 08 Double  
 40B0 Flag für Zwischencodierung (nach DATA 4E)  
 SSPEND 40B1 RAM Ende Pointer / Stringspaceende  
 40B3 Adresse des nächsten freien Zwischenspeichers für String  
 40B5 11 Zwischenspeicherplätze für String (Länge, Adresse)  
 40D3 vorläufiger Zwischenspeicher für Strings  
 40D4 enthält Stringadresse  
 SSPPNT 40D6 Stringspace Pointer  
 40DB Formatflag für Zahl/Stringumwandlung auch Zeiger Arraytab, Zeiger DIM Zähler  
 40DA letzte Datazeilennummer  
 ARRFLG 40DC Arraysperrung (b4 hex)  
 40DD Flag für RESUME und Return bei 60SUB  
 40DE Dataflag für INPUT (>0 wenn READ)  
 40DF Pointer für Progr.fortführung (wird bei NEW aus 40A4 geladen) + Entrypoint bei Kass. laden  
 40E1 AUTO Flag wenn 0 kein AUTO  
 40E2 nächste AUTO Zeilennummer  
 40E4 Zeilenabstand bei AUTO  
 40E6 aktueller Pointer Zeilennummer  
 40E8 Stackpointer  
 40EA Fehler Zeilennummer  
 40EC . -Zeilennummer für .Funktion  
 40EE Fehlerpointer Zeiger auf Anweisung bei deren Ausführung der letzte Fehler auftrat  
 40F0 Trappointer Zeiger auf Anfang der ERRORTRAP Routine ON ERROR GOTO  
 40F2 Trapflag - Errortrap wird bei auftreten eines Traps gesetzt und durch RESUME zurückgesetzt  
 40F3 . Position bei Zahlen (auch Pointer auf Vorzeichen etc. Klammer)  
 40F5 CONT Zeilennummer  
 40F7 CONT Pointer Zeiger auf Anweisung  
 TXTEND 40F9 Programmtextende Pointer / Anfang der Var Tabelle  
 VAREND 40FB Vartab Ende / Anfang der Array Tabelle  
 ARREND 40FD Arraytabellenende / enthält Anfang des freien Speichers bei BASIC  
 40FF Datazeiger zeigt auf Trennzeichen nach den zuletzt gelesenen Daten  
 4101 Var Tabelle Beginn A-Z  
 411A Ende Var Tabelle  
 411B Traceflag (AF hex)=TRON  
 411C Zwischenspeicher für Arithmetik  
 FACX Integer String Single Double  
 411D LSB  
 411E NMSB (next most signif. byte)  
 411F NMSB  
 4120 NMSB  
 4121 LSB LSB LSB NMSB enthält bei CLOAD Filenamen und Verify Flag  
 4122 MSB MSB NMSB NMSB  
 4123 MSB MSB  
 4124 EXP EXP  
 4125 Zwischenspeicher Sign-Flag

FACY	4127	LSB	LSB	LSB	LSB
	4128	MSB	MSB	NMSB	NMSB
	4129			MSB	NMSB
	412A			EXP	NMSB
	412B				NMSB
	412C				NMSB
	412D				MSB
	412E				EXP
NUMBUF	412F	Ausgabepuffer für Zahlenausgabe zusätzliches Byte für % bei Feldüberlauf			
	4130	normaler Anfang des Puffer			
ab	4152	Disk Basic Erweiterungen bis 41A3			
	4152	CVI			
	4155	FN			
	4158	CVS			
	415B	DEF			
	415E	CVD			
	4161	EOF			
	4164	LOC			
	4167	LOF			
	416A	MKI#			
	416D	MKS#			
	4170	MKD#			
	4173	CMD			
	4176	TIME#			
	4179	OPEN			
	417C	FIELD			
	417F	GET			
	4182	PUT			
	4185	CLOSE			
	4188	LOAD			
	418B	MERGE			
	418E	NAME			
	4191	KILL			
	4194	&-Funktion Oktale oder hex Konstante Absprung 249F			
	4197	LSET			
	419A	RSET			
	419D	INSTR			
	41A0	SAVE			
	41A3	LINE			
ab	41A6	Zeiger zur Erweiterung vorhandener Befehle durch das DISK BASIC bis 41E2			
	41A6	Disk BASIC Error Routine			
	41A9	USR			
	41AC	wird bei BASIC Initialisierung angesprungen bevor READY auf BS			
	41AF	wird von 0368 INLINE angesprungen Tastatureingabe			
	41B2	wird bei Programmeingabe angesprungen nach ersetzen der Befehle mit Token			
	41B5	Input scanner after updating program statement table			
	41B8	Input scanner nach Reinitialisierung BASIC			
	41BB	Neuinitialisierung BASIC nach END			
	41BE	wird von Endabfrage angesprungen (2169) initializing of system output device			
	41C1	during writing to system output device			
	41C4	wird von 0358 INCH2 angesprungen			
	41C7	Startroutine RUN Zeilennummer			
	41CA	wird von PRINT angesprungen Absprung 206F			
	41CD	PRINT # und PRINT item			
	41D0	wird von CR angesprungen (20FE)			
	41D3	wird bei TAB Auswertung angesprungen (2141) und PRINT on Kasette			
	41D6	wird von INPUT angesprungen			
	41D9	wird von MID# auf linker Seite der Anweisung angesprungen			
	41DC	READ Variable			
	41DF	wird von LIST angesprungen und Ende von READ			

	41E2	List processing bei System Befehl			
	41E5	:			
	41E6	0 *			
	41E7	,			
	41E8	I/O Buffer			
	41F8	Stackpointer BASIC			
	42E8	enthält 0 (vor BASIC Programm anfang)			
	42E9	BASIC Programm Level2 (kann bei Disk BASIC anders liegen 40A4 entnehmen!)			
	4300	-4303 Position der Schreib/Leseköpfe			
ab	430A	Pdrive für aktuelles Laufwerk			
	4312	BREAK Routine			
	4317	gibt an welches SYS Modul noch verfügbar 00=keins 03=SYS1 ... 1F=SYS09			
	4318	-4367 Input Buffer für DOS			
	434C	MEM Size			
ab	4371	Pdrive Drive 0			
ab	437B	Pdrive Drive 1			
ab	4385	Pdrive Drive 2			
ab	438F	Pdrive Drive 3			
	43E0	bis 43FF Chaining Buffer			
DOSCMD	4405	DOS-Befehl ausführen + nach DOSRDY		HL=Zeiger Befehl 0D Ende	
DOSERR	4409	Fehlermeldung des DOS ausgeben	F	A=Fehlercode	
DEBUG	440D	DEBUG aufrufen			
INTINS	4410	Benutzer Interrupt Routine einfügen	AF BC DE HL	DE = Zeiger auf Kontroll block der Interruptroutine	
INTDEL	4413	Benutzer Interrupt Routine löschen	AF BC DE HL	DE = Zeiger auf Kontroll block der Interruptroutine	
	4414	Beginn BASIC Programm???			
MOTCONT	4416	Drive Motoren weiterlaufen lassen	AF		
DOSCAL	4419	DOS-Befehl ausführen + zurück		HL=Zeiger Befehl 0D Ende AF=Fehlerstatus	
TFSPEC	441C	Filespec (HL) nach FCB (DE) übertragen	AF BC HL	HL + DE =Zeiger ggf. A=Fehlercode	
INIT	4420	File öffnen ggf. neu anlegen		DE=Zeiger auf FCB HL=Zeiger auf Buffer B=logische Recordlänge ggf. A=Fehlercode	
OPEN	4424	File öffnen		DE=Zeiger auf FCB ggf. A=Fehlercode HL=Zeiger auf Buffer B =logische Recordlänge	
CLOSE	4428	File schließen	AF (A=Fehlercode)	DE=Zeiger auf FCB AF wenn Z=0	
KILL	442C	File löschen		DE=Zeiger auf FCB ggf A=Fehlercode	
LOADDE	4430	Programm laden	BC	DE=Zeiger auf FCB HL=Startadresse d. Progr. ggf A=Fehlercode	
RUN	4433	Program laden + starten (keine Rückkehr)	AF BC	DE=Zeiger auf FCB	
READ	4436	nächsten Sector/Record aus File lesen	AF	DE=Zeiger auf FCB ggf. A=Fehlercode HL=Zeiger auf Buffer wenn Rec.länge <256	
WRITE	4439	nächsten Sector/Record in File schreiben	AF	DE=Zeiger auf FCB ggf. A=Fehlercode HL=Zeiger auf Buffer wenn log. Rec.länge <256	
VERIFY	443C	Sector/Record in File schreiben + testen	AF	DE=Zeiger auf FCB ggf. A=Fehlercode HL=Zeiger auf Buffer wenn log. Rec.länge <256	
POSO	443F	FCB auf Beginn eines Files positionieren		DE=Zeiger auf FCB ggf. A=Fehlercode	
POSBC	4442	FCB auf logische Record(BC) positionieren		BC=log.Rec. ggf. A=Fehlercode	
POSDEC	4445	FCB um 1 Record zurückpositionieren		DE=Zeiger auf FCB ggf. A=Fehlercode	
POSEOF	4448	FCB auf Fileende (EOF) positionieren		DE=Zeiger auf FCB ggf. A=Fehlercode	
EXPAND	444B	File erweitern, wenn File zu kurz ist		DE=Zeiger auf FCB ggf. A=Fehlercode	
POSRBA	444E	FCB auf RBA Format (HLC) positionieren		HL=1.+2. Byte C=3. Byte NEXT Feld DE=Zeiger auf FCB ggf. A=Fehlercode	
PUTEOF	4451	EOF im FPDE auf den neuesten Stand bringen	AF	DE=Zeiger auf FCB ggf. A=Fehlercode	
DRVSEL	445B	Drive auswählen und Motor starten	AF	A=Drive Nr. ggf. A=Fehlercode	
TSTDISK	445E	Drive auswählen, prüfen ob Diskette eingelegt	AF	A = Drive ggf. A=Fehlercode	
USRINS	4461	Benutzer Routine einfügen	AF BC DE HL	HL=Zeiger auf Routine ggf. A=Fehlercode	
USRDEL	4464	Benutzer Routine löschen	AF BC DE HL	HL=Zeiger auf Routine ggf. A=Fehlercode	
TEXTTV	4467	Text auf Bildschirm ausgeben	AF	HL=Textzeiger Textende 03 oder 0D hex	
TEXTLP	446A	Text auf Drucker ausgeben	AF	HL=Textzeiger Textende 03 oder 0D hex	
TIME	446D	Uhrzeit im Format HHMMSS in Buffer ablegen	AF BC DE HL	HL=Zeiger auf Buffer	

DATE	4470	Datum im Format MM/TT/JJ in Buffer ablegen	AF BC DE HL	HL=Zeiger auf Buffer
INSEXT	4473	Filetyp (/EXT) in Filespec einfügen	AF HL	DE=Zeiger auf Filespec HL=Zeiger auf Filetyp
UPCASE	45B5	Kleinbuchstaben und Großbuchstaben umwandeln	AF	A=ASCII-Code des Zeichen A=ASCII Großbuchst.
READS	4530	Sector von Diskette lesen	AF	DE=Sector(4308)=Drive ggf. A=Fehlercode HL=Zeiger auf Buffer
TESTS	4634	Sector auf Lesbarkeit testen	AF	DE=Sector(4308)=Drive ggf. A=Fehlercode
WRITDS	463C	Directory Sector auf Diskette schreiben	AF	DE=Sector(4308)=Drive ggf. A=Fehlercode HL=Zeiger auf Buffer
WRITES	4640	normalen Sector auf Diskette schreiben	AF	DE=Sector(4308)=Drive ggf. A=Fehlercode HL=Zeiger auf Buffer
RESTOR	4745	RESTORE Kommando an FDC senden	AF C	
FDCCMD	4747	Kommando mit richtiger TSR an FDC senden	AF	C=Kommando an FDC
WNBUSY	4750	warten bis der FDC nicht mehr Busy ist	AF	
FBREAK	475E	FORCE-INTERRUPT-Kommando an FDC senden	AF	
MOTON	4767	Drive Motoren starten	A	
DELAY1	47E3	ca. 55 usek warten und Status FDC lesen	AF	A=Status
FILPOS	4810	Disk-Position eines File-Sec. berechnen	BC	IX=Zeiger auf FCB ggf. A=Fehlercode IY=Zeiger auf 4380H DE=Sector Nr. A:00 File darf erweitert HL=Zeiger auf Filebuffer A:B6 nicht erweitert werden vorher ==> Call PUSHR !!!
DIRR	490A	liest einen Sector des Directory	AF HL	A=Sectornr,(4308H)Drivenr AF (A=Fehlercode wenn Z=0 HL=4200H, (4930H)=Sectornr
DIRW	491F	schreibt einen Sector des Directory	AF HL	(4930)=Secn,(4308)=Drivenr wie DIRR
GETFDE	4936	holt einen FDE aus dem Directory		A=DEC,(4308)=Drivenr ggf. A=Fehlercode HL zeigt auf FDE (42xx) (4930)=DIR Sectornr
NXTEOF	4968	NEXT Wert aus FCB holen + mit EOF vergleichen. A		IX=Zeiger auf FCB HL=1.+2. Byte dex NEXT C =3. Byte des NEXT Wertes F: C=1 Z=0 wenn NEXT < EOF C=0 Z=1 wenn NEXT = EOF C=0 Z=0 wenn NEXT > EOF
PUSHR	4980	Push Register siehe Rockrath S.3/39		
WRITXV	4AB8	Sector oder DIR Sector schr. ggf. Verify	AF BC	DE=Sector(4308)=Drive ggf. A=Fehlercode HL=Zeiger auf Buffer A = Verify nur wenn C <>0
WRITEV	4ABD	norm. Sec. auf Disk schr. ggf. Verify	AF B	DE=Sector(4308)=Drive ggf. A=Fehlercode HL=Zeiger auf Buffer C=Verify nur wenn C <>0
WRITDV	4ACA	Directory Sector schreiben ggf. Verify	AF B	DE=Sector(4308)=Drive ggf. A=Fehlercode C:Verify nur wenn C <>0 B :max. Anzahl Verify Versuche
GETSYS	4BC9	SYS-File laden und starten	AF BC DE HL	A=Code f. SYS-Filea ggf. A=Fehlercode
LOADHL	4C28	Programm laden	BC DE	HL=Zeiger auf FCB wie LOADDE
DIRPOS	4C74	Disk-Position eines DIR-Sectors berechnen	AF BC DE HL	A=Sector im DIR DE=Sectornr(Disk relativ) HL=Buffer DIR-Sec.(4200H)
MULTL	4C92	multipliziere A HL = L * A	F	A L A=0 HL=Ergabnis
MULTL	4C94	multipliziere A HL = HL * A	F	A, HL<8080h A * 65536 +HL
MULTC	4C9D	multipliziere HL C = HL * A?	F	A, HL<8080h HL * 256 + C
DIV05	4CB2	Dividiere HL=INT(HL/5), A=Rest	AF BC HL	HL HL=Quotient, ggf. A=Rest
DIVA	4CB4	Dividiere HL=INT(HL/A), A=Rest	AF BC HL	HL + A HL=Quotient, ggf. A=Rest
CP&CHL	4CC5	Vergleiche Text (BC) mit Text (HL)	AF HL	BC Textende 00H + HL wenn Text 1 = 2 A=00 Z=1 HL=Ende Text 2 sonst A=34 Z=0
NEXTC1	4CD5	nächstes Zeichen (HL) holen und Flag setzen	AF HL	HL=Zeiger auf Text Enter,Komma,Biankkennungen
NEXTC2	4CD9	nächstes Zeichen (HL) holen und Flag setzen	AF HL	HL=Zeiger auf Text Komma, blank (Flag)
DELAY2	4CED	ca. B * 3,75 msek warten	AF BC DE	B -

Adressen im BASIC/CMD (ob diese in jeder BASIC Version gleich sind wurde nicht geprüft , Kontrolle bei Sprungvectoren 4... hex !)

572E MERGE  
573B SAVE  
5746 Startroutine RUN Zeilennummer  
574A LOAD  
5756 wird nach ersetzen der Befehle durch Tokens angesprungen  
575D Start Tastatureingabe  
577F FN  
5786 INSTR  
578B MID\$ auf linker Seite der Anweisung  
5790 & ROUTINE  
5795 OPEN  
579A READ Variable  
579F LINE  
57FF CMD  
5852 DEF  
5892USR  
58C4 TIME\$  
58D8 MKD\$  
58DB MKS\$  
58DE MKI\$  
58EF CVD  
58F2 CVS  
58F5 CVI  
5908 LSET  
5909 RSET  
5E63 FIELD  
5EAB wird von INPUT angesprungen  
5EB9 am Ende von READ  
5EC5 EOF  
5EC9 LOF  
5ED1 LOC  
5F3B PRINT on Kasette und PRINT TAB  
5F4B PRINT # und PRINT item  
5F52 wird von PRINT angesprungen  
5FA0 List processing bei SYSTEM Befehle  
5FA0 wird von CR angesprungen  
5FA1 CLOSE  
5FB6 Neuinitialisierung nach END  
6076 during writing to system output device wird von INCH2 angesprungen  
6125 PUT  
6126 GET  
NAME ---> 1E4A = ERROR  
63B3 Input scanner after updating program statement table  
63BC Input scanner nach Reinitialisierung BASIC  
643B during initializing of system output device  
643C KILL