

FORMEL-128

Dieses Modul befindet sich noch in der Entwicklungsphase. Wir können daher hier den geplanten Leistungsumfang dieses Moduls nur unverbindlich angeben:

- Erweiterung des C-128 in allen 3 Betriebsarten.
- im 64-er-Modus alle Funktionen von Formel-64.
- im CP/M-Modus alle Funktionen von Formel-C.
- umfangreiche Befehlserweiterung für den 128-er-Modus.
- Beschleunigung der Load- und Save-Operationen auf das über 30-fache (im Vergleich zur 1541).
- Betrieb mit 1541, 1570 und 1571, auch im Mischbetrieb.
- Maus- bzw. Joystick-gesteuerte Benutzeroberfläche.
- Pull-Down-Menues.
- Terminalemulator für den Betrieb mit Mailboxen.
- komfortable Graphiksoftware.
- einfacher Einbau ohne zu Löten.

Formel-64plus

Dieses Modul entspricht dem Modul Formel-128, es entfallen jedoch die Funktionen, welche ausschließlich den CP/M-Modus und den 128-er-Modus des C-128 betreffen.

Und das Wichtigste: sowohl Formel-64 als auch Formel-C können ohne Probleme zu FORMEL-128 bzw. Formel-64plus aufgerüstet werden.

Die Preise für FORMEL-128 und FORMEL-64plus stehen noch nicht fest, aber daß sie wieder sensationell niedrig sind, können wir jetzt schon garantieren.

Liefertermin für FORMEL-128 und FORMEL-64plus: voraussichtlich 3. Quartal 1986. Beachten sie diesbezüglich bitte unsere Anzeigen in der 64-er.

FORMEL

64

BEDIENUNGS — HANDBUCH

GREWE COMPUTERTECHNIK

WIESENSTRASSE 82

D - 4350 RECKLINGHAUSEN

TELEFON (02361) 181354

Inhaltsverzeichnis

<u>Kapitel 1: EINBAU DES MODULS</u>	Seite 2
1.1 Einbau von Formel-64	Seite 2
1.2 Fehlersuche	Seite 4
1.3 Inbetriebnahme	Seite 5
<u>Kapitel 2: BEFEHLSERWEITERUNG</u>	Seite 6
2.1 Toolkit	Seite 6
2.2 DOS-Befehle	Seite 8
2.3 Graphik	Seite 10
<u>Kapitel 3: ASSEMBLER-PROGRAMMIERUNG</u>	Seite 14
3.1 Maschinensprachemonitor	Seite 14
3.2 Floppymonitor	Seite 15
3.3 Assembler	Seite 16
<u>Kapitel 4: FLOPPYFUNKTIONEN</u>	Seite 21
<u>Kapitel 5: CENTRONICS-SCHNITTSTELLE</u>	Seite 22
<u>ANHANG:</u>	
Illegale Opcodes	
Demoprogramm Graphik	
Demoprogramm Assembler	
Bild 1 und 2	

Garantiebestimmungen:

Wir gewähren auf unsere Produkte eine Garantie von 6 Monaten. Mängel, die auf Fehler im Material oder in der Herstellung zurückzuführen sind, beseitigen wir in dieser Zeit kostenlos. Weitergehende Ansprüche aus Mängelhaftung oder Schadensersatzansprüchen sind ausgeschlossen.

1.1) Einbau des Moduls

Bitte lesen Sie sich dieses Kapitel gut durch, bevor Sie mit dem Einbau des Moduls beginnen. Da keine Lötarbeiten anfallen, ist der Einbau relativ einfach.

Wir weisen jedoch darauf hin, daß wir für Defekte in Rechner und Floppy, die durch den Einbau entstehen, keine Haftung übernehmen. Falls Sie im Umgang mit elektronischen Bauteilen also keine Erfahrung haben, ist es ratsam, den Einbau von einem Fachmann vornehmen zu lassen!

Zunächst müssen Rechner und Floppylaufwerk ausgeschaltet werden.

Formel-64 besteht aus einem Steckmodul für den Expansionsport des Rechners und einem Kabel mit einer Adapterfassung für das Floppy-Laufwerk.

1.1a) Installierung im Floppy-Laufwerk:

Nach Lösen von vier Schrauben auf der Unterseite der Floppy wird deren Oberteil abgenommen. Dann wird vorsichtig ein Bauteil aus seiner Fassung gehoben. Ein breiter Schraubenzieher ist hierfür empfehlenswert.

Folgendes IC muß aus seiner Fassung gehoben werden:

- beim alten Floppylaufwerk mit der über die ganze Länge reichenden Platine das IC in der Fassung UAB1.
- beim neueren Floppylaufwerk mit der kürzeren Platine das IC in der Fassung UC3.
- beim tragbaren Rechner SX-64 das IC in der Fassung UBC3.

Das IC selbst trägt die Bezeichnung 6522 (evtl. auch 6522A o.ä.).

In die freigewordene Fassung wird nun die am Ende des Flachbandkabels befindliche Zwischenfassung eingesetzt. Die Einkerbung am Ende der Zwischenfassung muß mit der Einkerbung der Fassung auf der Platine zur Deckung kommen.

In diese Zwischenfassung wird nun das zuvor herausgenommene IC eingesetzt, wobei wieder auf Deckung der Einkerbungen zu achten ist.

Das IC muß jetzt wieder in der gleichen Richtung wie vor dem Herausnehmen in der Fassung stecken.

ACHTUNG: Falsches Einstecken von IC oder Fassung kann zu erheblichen Zerstörungen führen!

Das Flachbandkabel kann nun am Durchbruch für den Sicherungshalter aus dem Gehäuse herausgeführt werden. Als letztes ist der Gehäusedeckel wieder anzubringen.

Optional: Anschluß des Kabels für ein zweites Laufwerk:

Für 49.- DM ist ein Anschlußkabel für ein zweites Floppy-Laufwerk erhältlich. Dieses Kabel hat auf beiden Seiten die oben erwähnte Adapterfassung. Das eine Ende des Kabels wird im zweiten Laufwerk installiert wie oben beschrieben, die Zwischenfassung am anderen Ende wird auf die am Ende des am Modul befestigten Kabels befindliche Zwischenfassung aufgesteckt.

Im zweiten Laufwerk muß die Geräteadresse auf 9 geändert werden.

1.1b) Installierung in C-64 und SX-64:

Das Modul wird in den Modulsteckplatz des Rechners eingesteckt. Auf einer Breitseite des Modulgehäuses ist eine glatte Fläche (für Klebeetiketten). Diese Seite muß nach oben zeigen.

Aus dem Modul kommt außer dem Flachkabel zur Floppy noch ein einzelnes Kabel mit einer Klemme am Ende heraus. Dieses Kabel muß nicht unbedingt angebracht werden, es ist nur dann notwendig, wenn Programme den RAM-Speicher im Bereich \$E000-FFFF benutzen (z. B. Simon's Basic und einige Spiele). Solche Programme laufen ohne dieses Kabel nicht einwandfrei.

Um dieses Kabel anzuschließen, muß der Rechner geöffnet werden. Dazu sind drei Schrauben an seiner Unterseite zu lösen. Das Kabel wird an das Adreßmanager-IC (in der Fassung U17) angeschlossen, und zwar an Pin 7.

Eine zweite Anschlußmöglichkeit für dieses Kabel ist die CPU selbst. Dieser Baustein steckt in Fassung U7 und hat die Bezeichnung 6510. Hier muß die Klemme an Pin 28 befestigt werden. Bei Druck auf das Ende der Klemme schiebt sich an der Spitze ein Haken heraus, dieser wird an eins der oben erwähnten IC-Beine angeschlossen.

Es ist darauf zu achten, daß die Klemme nicht 2 IC-Beinchen kurzschließt.

Sehen Sie sich bitte hierzu auch Bild 2 im Anhang an.

Das Kabel sollte direkt am Modul vorbei neben dem Modulschacht in den Rechner geführt werden.

Installierung in C-128 und C-128d:

Das Modul wird wie oben beschrieben eingesteckt.

Der Anschluß des einzelnen Kabels ist jedoch aufwendiger.

Da die Platine in diesem Rechner mit einem Abschirmblech überdeckt ist, kann die Klemme evtl. nicht darunter untergebracht werden. In diesem Fall müßte die Klemme abgetrennt und das Kabel angelötet werden.

Falls Sie keine Löt erfahrung haben, sollten Sie dies besser von einem Fachmann vornehmen lassen.

In den Rechnern C-128 und C-128d wird das Kabel an den Baustein 8502 angeschlossen, und zwar an Pin 29.

Das serielle Anschlußkabel zum Floppy-Laufwerk muß weiterhin angeschlossen bleiben.

Nun werden erst das Floppy-Laufwerk und dann der Rechner eingeschaltet. Ausgeschaltet werden die Geräte in umgekehrter Reihenfolge.

ACHTUNG: Bei Nichteinhalten der Einschalt- bzw. Ausschalt-Reihenfolge sind Schäden am Modul bzw. an der Floppy nicht auszuschließen.

Betrieb mit C-128 und C-128d: der Befehl "go 64" funktioniert mit Formel-64 nicht. Um in den 64-er Modus zu gelangen, muß während des Einschaltens oder beim RESET die Commodore-Taste gedrückt werden.

1.2 Fehlersuche

Sollte Ihr Formel-64-Modul nicht einwandfrei funktionieren, so muß nicht unbedingt ein Defekt am Modul vorliegen. Anhand der folgenden Liste können Sie versuchen, eventuelle Schwierigkeiten selbst zu beheben.

I) Rechner meldet sich mit seiner normalen Einschaltmeldung
Vermutlich sitzt das Modul nicht richtig im Modulschacht. Falls es etwas locker im Schacht sitzt, sollte es mit etwas mehr Kraft hineingedrückt werden.

II) Rechner meldet sich bei eingestecktem Modul überhaupt nicht
Es könnte ein Defekt im Expansionsport vorliegen. Falls Sie die Möglichkeit haben, das Modul an einen anderen Rechner anzuschließen, können Sie auf diese Art feststellen, ob der Fehler am Modul oder an Ihrem Rechner liegt. Für diesen Test muß das Kabel nicht in das Floppy-Laufwerk eingebaut werden.

III) Modul meldet sich, aber es werden keine Programme geladen

Hier gibt es mehrere Möglichkeiten:

- a) es wurde vergessen, den herausgezogenen Baustein 6522 wieder auf die Adapterfassung aufzustecken (Polung beachten!).
- b) Der Baustein 6522 in Ihrem Laufwerk könnte defekt sein. Dieser Defekt ist bei normalem Betrieb nicht festzustellen, da Formel-64 einen 8-Bit-Port dieses Bausteins benutzt, der im normalen Betrieb von der Floppy nicht verwendet wird. Falls Sie nicht die Möglichkeit haben, Formel-64 in einem anderen Floppylaufwerk auszutesten, kann die Funktionsfähigkeit des Bausteins auch folgendermaßen geprüft werden: Auf der Laufwerksplatine befinden sich zwei Bausteine mit der Bezeichnung 6522 bzw. 6522A. Wenn diese (bei ausgebautem Modul) gegeneinander ausgetauscht werden (Polung beachten!) und die Floppy dann nicht mehr einwandfrei arbeitet, liegt der Defekt an diesem Baustein. Er muß dann erneuert werden.

Sollten Sie trotz der oben beschriebenen Möglichkeiten keine einwandfreie Funktion mit Formel-64 erreichen, ist es notwendig, uns Ihr Modul zur Überprüfung (bzw. Umtausch) zurückzusenden. Wir werden uns bemühen, es Ihnen so schnell wie möglich wieder zur Verfügung zu stellen.

1.3) Inbetriebnahme des Moduls

Nach dem Einschalten des Rechners oder nach Betätigen des RESET-Tasters ist das Modul automatisch initialisiert.

Wird beim Einschalten oder RESET gleichzeitig die CTRL-Taste gedrückt, übergeht der Rechner ein evtl. eingeladenes Autostart-Programm im Speicher bei \$8000.

Wenn beim Einschalten oder RESET die INST/DEL-Taste gedrückt wird, meldet sich der Rechner mit einem Menu:

- F1-Taste: springt in das oben erwähnte Autostartprogramm.
- F3-Taste: schaltet das Modul aus.
- F5-Taste: initialisiert das Modul, aber ohne Toolkit-, DOS- und Graphikbefehle, dadurch wird die Kompatibilität erhöht.
- F7-Taste: verläßt das Menu und schaltet das Modul ein.

Neben der Basicbefehlserweiterung bietet Formel-64 noch eine Reihe von Funktionen, die durch Tastendruck bewirkt werden:

SHIFT/RUN: wenn mit dem Cursor nach Einladen der Directory auf ein Programm darin gefahren und diese Tastenkombination gedrückt wird, wird das gewählte Programm geladen und gestartet.

Weitere Funktionen stehen zur Verfügung, wenn die RESTORE-Taste gedrückt wird. Dann geht der Rechner in eine Warteschleife und wartet auf eine weitere Taste. Im folgenden ist erklärt, welche Tasten welche Funktionen auslösen:

- RESTORE-RETURN: zeigt die Diskettendirectory an.
- RESTORE-INST/DEL: lädt und startet das erste Programm auf Disk.
- RESTORE- F1: Drucker-Hardcopy (siehe nächste Seite).
- RESTORE- F3: Drucker-Hardcopy invers (siehe nächste Seite).
- RESTORE- F5: Einsprung in den Maschinensprachemonitor.
- RESTORE- F7: Statusabfrage der Floppylaufwerke.

Beim Druck auf die RESTORE-Taste wird die Rahmenfarbe geändert, nach Ausführen der Funktion wird sie wieder zurückgesetzt.

Wird die RESTORE-Taste versehentlich gedrückt, kann die Warteschleife durch Drücken einer Cursor-Taste wieder verlassen werden.

Die RESTORE-Taste sollte allerdings nicht gedrückt werden in Programmen, die den NMI-Vektor verbiegen oder den RAM hinter dem Betriebssystem einschalten. Das Drücken der Taste könnte in diesem Fall zum Absturz des Programms führen.

Druckerhardcopy:

Die Hardcopyfunktion unterstützt die Commodore-Drucker MPS 801/802/803. Außerdem werden EPSON-FX/RX-80-kompatible Drucker unterstützt, allerdings nur, wenn diese mit unserem Centronics-Kabel (Best.-Nr. 406-cdk) angeschlossen werden. Bei Verwendung von anderen Interfaces für diese Drucker kann die Hardcopy nicht gedruckt werden.

Die verschiedenen Druckertypen werden folgendermaßen selektiert:

- für Epson-kompatible Drucker wird die oben angegebene Tastenkombination eingegeben.
- für die Drucker MPS 801 u. 803 wird zusätzlich zur F1-Taste die rechte Shift-Taste gedrückt.
- für den Drucker MPS 802 wird zusätzlich zur F1-Taste die linke Shift-Taste gedrückt.

Es wird der jeweils angezeigte Bildschirminhalt ausgedruckt. Hardcopies vom Textbildschirm werden in halber Papierbreite ausgedruckt, hochauflösende Graphikbilder über die volle Papierbreite.

Bei einigen Betriebssystemversionen der Commodore-Drucker kann es hier allerdings Schwierigkeiten geben, so daß die Hardcopy in x-Richtung nicht gestreckt wird. Dies ist leider nicht zu umgehen. Bei den Druckern MPS 801/803 wird aus technischen Gründen eine Hardcopy vom Textbildschirm nicht geschlossen gedruckt, d.h., zwischen den Zeilen sind Zwischenräume.

2) Befehlserweiterung

Formel-64 stellt eine Reihe neuer Befehle zur Verfügung. Diese werden wie normale Basic-Befehle eingegeben und können auch entsprechend abgekürzt werden.

2.1) Toolkit

AUTO XX

schaltet die automatische Zeilennummerierung mit Zeilenabstand XX ein. AUTO ohne Parameter schaltet sie wieder aus.

DELETE AA-XX

löscht die Programmzeilen von AA bis XX. Es ist darauf zu achten, daß die angegebenen Zeilennummern auch wirklich vorhanden sind. Die zweite Zeilennummer muß größer sein als die erste, da sonst das Programm zerstört wird.

DEZ "XXXX"

wandelt die hexadezimale Zahl XXXX in ihren dezimalen Wert um. Statt XXXX kann auch eine entsprechende Stringvariable angegeben werden. XXXX bzw. die Stringvariable muß eine Länge von 4 Zeichen haben.

FIND "TEXT"

sucht nach der Zeichenkette TEXT im Basicprogramm. Für unbekannte Zeichen kann als Joker ein "?" eingesetzt werden. Werden keine Gänsefüßchen eingegeben, sucht FIND nach dem Token des angegebenen Basicbefehls.

HCOPY X Y

führt eine Hardcopy des Bildschirms auf den Drucker aus. Für X wird ein "n" für normalen Ausdruck oder ein "i" für inversen Ausdruck eingegeben. Y ist ein "E" für Epson-kompatible Drucker oder eine "1" für die Drucker MPS 801 und 803 oder eine "2" für den Drucker MPS 802.

Beispiel: "HCOPY I E" führt eine inverse Hardcopy auf Epson-kompatiblen Druckern aus.

Für den Hardcopy-Ausdruck mit dem Befehl HCOPY gelten die gleichen Bedingungen und Einschränkungen wie beim Ausdruck mit den Funktions-Tasten (siehe dort).

HELP

zeigt eine Liste der Toolkit-, DOS- und Graphikbefehle auf dem Bildschirm an.

HEX X

wandelt die dezimale Zahl X in ihren entsprechenden hexadezimalen Wert um. Statt X kann auch eine numerische Variable angegeben werden.

JUMP X

mit X=0-9, A-F führt einen Einsprung in Maschinenprogramme aus.

Beispiel: JUMP 6 springt nach \$6000, JUMP C springt nach \$C000.

LLIST

entspricht den Befehlen OPEN 1,4 : CMD 1 : LIST.

LPRINT "TEXT"

gibt Ausgaben statt auf den Bildschirm an den Drucker. LPRINT ohne Text bzw. Stringvariable erzeugt einen Syntax Error.

OLD

stellt zuvor gelöschte Basic-Programme wieder her, sofern der Speicher nicht durch neue Programme überschrieben wurde. Vor OLD darf kein Syntax Error vorkommen, da sonst der Programmanfang zerstört werden kann.

OFF

führt einen RESET aus, ohne das Modul zu initialisieren.

RENUM inc, anf

führt eine Neunummerierung der Zeilennummern eines Basicprogramms aus mit anf = erste neue Zeilennummer und inc = Schrittweite.

Werden keine Parameter eingegeben, so werden als Default die Werte 10, 100 eingesetzt.

RESET

führt einen RESET aus, das Modul wird initialisiert.

TKITOFF

schaltet die gesamte Befehlserweiterung ab. Dadurch wird die Kompatibilität erhöht.

2.2) DOS-Befehle

Die in Klammern angegebenen Werte müssen nicht eingegeben werden.

BACKUP

ermöglicht das Kopieren von kompletten Disketten. Außerdem kann mit diesem Befehl auf das schnelle Format (16-fach) konvertiert werden. Nach Eingeben von BACKUP meldet sich ein Menu. Wenn die voreingestellten Werte bei der Abfrage übernommen werden sollen, genügt RETURN. Falls nicht, sind die gewünschten Parameter einzugeben:

DRIVE: 8 oder 9.

FORMAT: f für schnelle bzw. s für normale Formatierung.

TRACK: 1 bis 35.

VERIFY: y für ja und n für nein.

Nach Ausführung des Backups erscheint die Frage END? [y,n,r].

Dabei steht y für ja, n für nein und r für Wiederholung.

Durch BACKUP wird ein evtl. im Speicher befindliches Programm gelöscht! BACKUP kann während der Ausführung nur durch einen RESET unterbrochen werden.

ACHTUNG: als Quell- und Zieldiskette darf nicht dieselbe Diskette verwendet werden, da sonst die Programme auf dieser teilweise gelöscht werden. Es muß also immer mit 2 Disketten gearbeitet werden (siehe auch im Kapitel "Floppy-Funktionen").

Der BACKUP dauert länger, wenn vom schnellen Format auf das normale oder umgekehrt konvertiert wird.

Während des Backup-Vorgangs können auf der oberen Bildschirmhälfte Zeichenketten erscheinen, dies ist technisch bedingt und ohne Einfluß auf den Kopiervorgang.

CAT oder DIR

listet das Disketteninhaltsverzeichnis auf, ohne Programme zu löschen.

DAPPEND "(L:)NAME" mit L=0 o. 1 für erstes oder zweites Laufwerk. hängt das Programm "NAME" an das im Speicher befindliche Programm an. Eventuell müssen vorher die Zeilennummern der Programme angepaßt werden.

DLOAD "(L:)NAME"

ersetzt den Befehl LOAD "NAME",8,1.

DSAVE "NAME"

ersetzt den Befehl SAVE "NAME",8.

DVERIFY "NAME"

ersetzt den Befehl VERIFY "NAME",8,1.

HEADER

formatiert Disketten. Der Befehl ist selbsterklärend.

Bei der Abfrage ID? sollte auf jeden Fall ein zweistelliger Wert eingegeben werden. Bei der Abfrage NUMBER? kann ein 5-stelliger Name eingegeben werden, dieser überschreibt in der Directory die ID, die ID selbst wird jedoch nicht geändert.

FCOPY

Der F(ile)copy-Befehl ermöglicht es, einzelne Files zu kopieren. Dazu werden die Files aus der Directory angezeigt. Der Rechner wartet bei jedem File auf die Eingabe "y" für "kopieren" oder "n" für "nicht kopieren".

Dateien (seq. und rel.) können nicht kopiert werden.

Wird mit zwei Laufwerken gearbeitet, muß die Zieldiskette schon vor dem Starten des Kopiervorgangs eingelegt sein.

Nach Abschluß des Kopiervorgangs meldet sich der Rechner mit END? [y,n]. y steht für ja und bewirkt einen RESET, n steht für nein und startet FCOPY aufs neue. Wie BACKUP löscht auch FCOPY ein im Speicher stehendes Programm.

LRUN "NAME"

lädt das Programm "NAME" von Diskette und startet es.

REPLACE "NAME"

schreibt das im Speicher befindliche Programm anstelle des Programms "NAME" auf die Diskette. REPLACE entspricht dem Befehl SAVE "@:NAME",B. Da zuerst ein Scratch- und dann ein Save-Befehl ausgeführt wird, wird der bekannte Fehler im Betriebssystem des Floppylaufwerks umgangen.

SEND "(L:)Befehle"

ersetzt den Befehl OPEN 1,8,15,"Befehle".

STATUS

zeigt die Statusmeldung des Floppylaufwerks auf dem Bildschirm.

2.3) Graphik

Einführung

Die Graphikbefehle unterstützen sowohl hochauflösende Graphiken mit 320*200 Punkten als auch Multicolorgraphiken mit 160*200 Punkten Auflösung. In einer Multicolorgraphik ist die Auflösung in x-Richtung also nur halb so hoch wie bei einer Hires-Graphik, d.h. der Abstand zwischen zwei Punkten ist doppelt so groß. Es können bis zu vier Bildschirmseiten verwaltet werden.

Im Anhang finden Sie ein Beispielprogramm ausgedruckt, in dem viele der hier aufgeführten Befehle verwendet werden.

Es werden zunächst einige bei den Befehlserklärungen vorkommenden Abkürzungen erklärt, dann folgt eine Liste der Befehle.

Koordinaten x und y:

Der Bildschirm wird im Graphikmodus in eine Matrix von 320*200 Punkten aufgeteilt. x gibt die x-Koordinate an und y die y-Koordinate. Der Ursprung des Koordinatensystems, die Koordinate (0,0) liegt in der linken unteren Ecke des Bildschirms. x kann Werte von 0-319 annehmen, der Werte-Bereich von y geht von 0-199.

Obwohl bei einer Multicolorgraphik in x-Richtung nur eine Auflösung von 160 Punkten möglich ist, gilt auch hier der oben angegebene Wertebereich von 0-319, die Werte werden rechnerintern auf die halbe Auflösung umgerechnet.

Zeichentyp zt:

Dieser Wert gibt an, wie ein Punkt auf dem Bildschirm erscheint.

Er kann Werte von 0-4 annehmen:

zt=0	löscht einen Punkt
1	zeichnet einen Punkt in Farbe 1
2	zeichnet einen Punkt in Farbe 2
3	zeichnet einen Punkt in Farbe 3
4	invertiert einen Punkt.

Die Werte 2 und 3 sind im High-Resolution-Modus ohne Bedeutung, da außer der Hintergrundfarbe nur eine Farbe möglich ist.

Für den Wert 4 (Invertierung) gilt im Multicolormodus eine Besonderheit: ein Punkt der Farbe 1 erhält die Farbe 2,

ein Punkt der Farbe 2 erhält die Farbe 1,

ein Punkt der Farbe 3 wird gelöscht,

ein nicht gesetzter Punkt wird in Farbe 3 dargestellt.

Farbtabelle

Es können Werte von 0-15 angegeben werden, diese sind den folgenden Farben zugeordnet:

hg= 0	schwarz	hg= 8	Orange
1	Weiß	9	Braun
2	Rot	10	Hellgrau
3	Türkis	11	Grau 1
4	Violett	12	Grau 2
5	Grün	13	Hellgrün
6	Blau	14	Hellblau
7	Gelb	15	Grau 3

Graphikbefehlssatz

GRAPHIK n

Dieser Befehl bereitet den Rechner auf die Verwendung einer Bildschirm-Graphik vor. Der Parameter n kann Werte von 1 bis 4 annehmen und gibt die Anzahl der gewünschten Graphikseiten an. Der Befehl GRAPHIK muß am Anfang aller Graphikoperationen (in Programmen am Programmfang) stehen, da er den Speicherplatz des Rechners einschränkt. Die obere Speichergrenze wird entsprechend der Anzahl der Graphikseiten herabgesetzt. ACHTUNG: steht dieser Befehl nicht am Programmfang und werden vorher Strings verwendet, sind diese anschließend verloren!
Den Zusammenhang zwischen der Anzahl der Graphikseiten und dem freien Speicherplatz gibt die folgende Tabelle an:

n	Adresse der Seite	obere Speichergrenze	freier RAM
1	\$A000 = 40960	\$8BFF = 35839	33789
2	\$6000 = 24576	\$5FFF = 24575	22525
3	\$4000 = 16384	\$3FFF = 16383	14333
4	\$2000 = 8192	\$1FFF = 8191	6141

Alle Graphikbefehle außer GSAVE, GLOAD und TEXT erzeugen einen Overflow-Error, wenn vorher nicht GRAPHIK n eingegeben wurde.

HIRES c1, hg

schaltet die hochauflösende Graphik ein (320*200 Punkte). c1 ist die Zeichenfarbe und hg ist die Farbe des Bildschirm-Hintergrundes (siehe Farbtabelle). Es müssen immer beide Parameter angegeben werden.

MULTI c1, c2, c3, hg

schaltet die Multicolorgraphik ein (eingeschränkte Auflösung von 160*200 Punkten). c1, c2 und c3 geben drei Farben aus der Farbtabelle an, die in der Graphik verwendet werden sollen. hg ist die Hintergrundfarbe. Es müssen immer alle Parameter angegeben werden.

FRAME rf

setzt die Rahmenfarbe. Die Werte für rf sind der Farbtabelle zu entnehmen.

CLEAR

löscht den Graphikbildschirm.

TEXT

schaltet den Bildschirm vom Graphikmodus wieder in den Textmodus zurück. Bei eingeschaltetem Graphikmodus kann der Befehl natürlich nur blind eingegeben werden.

Wird der Befehl TEXT benutzt, ohne daß vorher ein anderer Graphik-Befehl verwendet wurde, wird unter Umständen eine willkürliche Farbkombination des Bildschirms eingestellt, das können z. B. schwarze Zeichen auf schwarzem Grund sein, so daß nichts mehr zu sehen ist. Allerdings ist der TEXT-Befehl ohne vorherigen Graphik-Befehl sinnlos.

PAGE n

tauscht das sichtbare Bild (dies entspricht der Page 1) mit dem Inhalt der angegebenen Page aus. Die Lage und Nummer der Pages wurden schon beim Befehl Graphik erklärt. n kann die Werte 2, 3 und 4 annehmen. n=1 ist sinnlos, da die Graphikseite dann mit sich selbst ausgetauscht würde.

Der Parameter n darf nicht größer sein als der beim Graphikbefehl vorher eingegebene, da sonst Programme überschrieben werden könnten. Wird ein größerer Wert eingegeben, gibt es einen Overflow-Error.

COPY n

schreibt den Inhalt der angegebenen Page in die Bildschirmseite. Parameter wie bei PAGE.

MIX n

überlagert den Inhalt der angegebenen Page auf die Bildschirmseite, so können Graphikbilder gemischt werden. Parameter wie bei PAGE.

INVERT

invertiert die Graphik. In Multicolorgraphiken werden die Farben invertiert wie im Absatz "Zeichentyp" beschrieben.

GSAVE "NAME"

speichert die angezeigte Graphikseite einschließlich aller Parameter auf Diskette ab.

GLOAD "NAME"

lädt ein auf der Diskette gespeichertes Graphikbild. Wenn das eingeladene Bild vorher mit diesem Modul abgespeichert wurde, wird nach dem Einladen automatisch der Graphikmodus eingeschaltet und das Bild gezeigt. Mit dem Befehl TEXT kann der Graphikmodus wieder verlassen werden.

PLOT x, y, zt

setzt einen Punkt auf die Koordinate x,y. zt ist der Zeichentyp.

LINE x1, y1, x2, y2, zt

zieht eine Linie von der Koordinate x1,y1 nach x2,y2.

DRAW x,y,zt

entspricht eigentlich dem Befehl LINE, bei DRAW wird jedoch nur die Endkoordinate angegeben. Als Anfangskoordinate wird die Endkoordinate einer vorher mit DRAW gezeichneten Linie angenommen. Vor dem ersten DRAW-Befehl muß mit PLOT x,y,zt eine Anfangskoordinate eingegeben werden.

BOX x,y,lx,ly,zt

zeichnet ein Rechteck. x,y ist die Koordinate der linken unteren Ecke des Rechtecks, lx gibt die Seitenlänge in x-Richtung und ly die Seitenlänge in y-Richtung an.

BLOCK x, y, lx, ly, zt

Wie BOX, jedoch wird ein ausgefülltes Rechteck gezeichnet.

CIRCLE x, y, rx, ry, zt

zeichnet einen Kreis bzw. eine Ellipse. x,y ist der Mittelpunkt des Kreises, rx der Radius in x-Richtung und ry der Radius in y-Richtung.

FILL x, y, zt

füllt eine geschlossene Fläche, in der der Punkt x,y liegt, mit der in zt angegebenen Farbe aus.

HPRINT x, y, zt, "TEXT"

schreibt einen Text in horizontaler Richtung auf den Bildschirm. x,y ist die linke untere Ecke des ersten Zeichens. Statt "TEXT" kann auch eine Stringvariable eingegeben werden, numerische Variablen müssen vorher mit dem Befehl STR\$(var) in Stringvariable umgewandelt werden.

Im Multicolormodus wird der Text wegen der geringeren Auflösung in doppelter Breite ausgedruckt.

VPRINT x, y, zt, "TEXT"

entspricht dem Befehl HPRINT, jedoch wird der Text vertikal ausgegeben.

Fehlermeldungen

Folgende Fehlermeldungen können auftreten:

OVERFLOW-ERROR: es wurde vergessen, den Befehl GRAPHIK vorher einzugeben
oder der Parameter n beim PAGE-, COPY- oder MIX-Befehl ist größer als die vorher mit GRAPHIK angewählte Seitenzahl.

ILLEGAL-QUANTITY-ERROR: die x-Koordinate liegt nicht im Bereich 0-319 oder die y-Koordinate liegt nicht im Bereich 0-199
oder der Parameter n (bei GRAPHIK, PAGE, COPY, MIX) liegt nicht im Bereich 1-4.

Bei den meisten Fehlermeldungen schaltet der Rechner zurück in den Textmodus, es kann jedoch auch vorkommen, daß der Graphikmodus eingeschaltet bleibt. In diesem Fall kann mit dem blind eingegebenen Befehl TEXT in den Textmodus zurückgeschaltet und die Fehlermeldung abgelesen werden.

3) Assembler-Programmierung

Für die maschinennahe Programmierung sind in diesem Modul einige komfortable Hilfsmittel installiert: der 2-Pass-Assembler, der Maschinensprachemonitor und der Floppymonitor.

Auf den nächsten Seiten finden Sie diese Programme näher erläutert. Diese Anleitung kann jedoch nicht eine Einweisung in die Programmierung in Maschinensprache sein, dafür gibt es eine Vielzahl von Büchern auf dem Markt. Es werden also nur die spezifischen Funktionen dieser Programmpakete erläutert.

3.1) Maschinensprachemonitor

Der Monitor wird durch Drücken der Tasten RESTORE-F5 aufgerufen. Im Maschinensprache-Monitor ist der Floppy-Monitor integriert.

Die in der folgenden Befehlserläuterung auftretenden Abkürzungen werden nachfolgend erklärt:

AA = Anfangsadresse, 2-Byte-Wert, hexadezimal, Beispiel: E20A

EA = Endadresse, wie oben

BY = Byte, 1-Byte-Wert, hexadezimal, Beispiel: F0

Die Befehle und die einzelnen Adressen sind immer durch eine Leerstelle zu trennen.

Die Eingabe der Opcodes erfolgt in der üblichen Syntax, wobei vor die Operanden ein \$ zu setzen ist. Beispiel: lda \$01.

Den Eingabemodus kann man mit SHIFT/RETURN wieder verlassen.

Der Monitor verarbeitet auch die meisten der "illegalen" Opcodes, eine Liste dieser Opcodes finden Sie im Anhang.

Mit den Cursortasten kann beim Memory-Dump und Disassemblieren auf- und abwärts gescrollt werden.

Befehle:

a AA Opcode: assembliert ab der angegebenen Adresse, Beispiel: a 2F00 ldx #\$02

b adresse zähler: setzt einen Breakpoint an die angegebene Adresse (s. auch Quickstep). Adresse und zähler sind vierstellige Hex-Zahlen ohne vorangestelltes "\$". Beispiel: b E000 0002

c AA EA A2: vergleicht den Speicherbereich AA-EA mit dem Speicher ab A2. Beispiel: c 2000 3000 5000

d AA (EA): disassembliert von AA bis EA (die Endadresse muß nicht eingegeben werden). Wird im Listing ein großes D an den Beginn der Zeile gestellt, kann der Hex-Wert geändert werden, bei einem kleinen a kann der Opcode geändert werden.

f AA EA BY: füllt den angegebenen Bereich mit BY.

g (adresse): führt einen JUMP an die angegebene Adresse aus, bei fehlendem Argument wird ein Sprung an die Programmzähler-Adresse ausgeführt.

h AA EA BY BY: sucht im angegebenen Bereich nach der folgenden Bytekombination.

l "name",GN: lädt ein Programmfile vom angegebenen Gerät, GN ist die Gerätenummer (zweistellig!).
Beispiel: l "name",08 lädt File von Floppy.

m AA (EA): zeigt den angegebenen Bereich als Hexdump an. Einzelne Bytes im Speicher können geändert werden, indem sie im Listing einfach überschrieben werden.

po: leitet alle folgenden Ausgaben auf den Drucker.

pf: schaltet die Druckerausgabe wieder ab.

q adresse: QUICKSTEP, ähnlich g, das angesprungene Programm wird bis zu einem Breakpoint so oft durchlaufen, wie der bei b gesetzte Parameter "zähler" angibt.

r: Anzeigen der Register und des Programmzählers. Die Werte können im Listing geändert und mit RETURN übernommen werden.

s "name",GN,AA,EA+1: speichert das Programm im angegebenen Speicherbereich ab. Die Gerätenummer GN muß zweistellig sein.

t AA EA A2: verschiebt den Speicherbereich AA-EA an die Adresse A2.

w AA: SINGLESTEP, arbeitet Assemblerprogramme im Einzelschrittmodus ab, der Druck einer beliebigen Taste bewirkt schrittweises Vorgehen.

x: führt zurück ins Basic.

Die Befehle a, c, d, f, h, m können durch den Buchstaben u ergänzt werden. Sie beziehen sich dann auf die versteckten RAM-Bereiche, das sind die Bereiche, die z. B. hinter dem Kern-ROD und dem BASIC-ROD liegen. Beispiel: du A000 BFFF disassembliert den hinter dem Basic-ROD liegenden RAM-Bereich.

3.2) Floppy-Monitor

Die Befehle a, d, f, h, m beziehen sich, mit dem Zusatz f versehen, auf den Floppy-Speicher. Beispiel: mf E000 FFFF.
Weitere Befehle:

cf AA EA A2: vergleicht den Bereich AA-EA in der Floppy mit dem Bereich ab A2 im Rechner.

Tf AA EA A2: verschiebt den Bereich AA-EA aus dem Rechner in die Floppy an die Adresse A2.

tf AA EA A2: verschiebt den Bereich AA-EA aus der Floppy in den Rechner an die Adresse A2.

fe adresse: führt einen JUMP an "adresse" in der Floppy aus.

fr pu tr se: liest den Sektor se auf Track tr in den Puffer pu in der Floppy ein. pu kann Werte von 0-4 annehmen. Beispiel: fr 1 1A 01. (Puffer 0 liegt bei \$0300, Puffer 1 bei \$0400 usw..) Falls nach fr nicht "status ok" angezeigt wird, muß die Diskette vorher mit fs "i" initialisiert werden.

fw pu tr se: schreibt Pufferinhalt in den angegebenen Sektor.

fs "befehl": sendet Befehle zur Floppy.

@: zeigt den Status der Floppy-Laufwerke an.

3.3) Assembler

Einführung:

Der Assembler ist ein nützliches Instrument zur Erstellung von Maschinensprache-Programmen. Durch die Möglichkeit, statt direkter Sprünge und Adressen Labels und sogar logische Ausdrücke zu verwenden, eignet sich dieser 2-Pass-Assembler - im Gegensatz zum Direktassembler - besonders zur Erstellung von längeren Programmen.

Für die Benutzung eines Assemblers sind selbstverständlich Kenntnisse in der Maschinenspracheprogrammierung notwendig. Diese Bedienungsanleitung versteht sich nicht als Einweisung in die Maschinensprache-Programmierung, für diesen Zweck gibt es ausreichend Literatur auf dem Markt.

Der Quelltext des Assemblers wird wie ein Basic-Programm erstellt, das hat den Vorteil, daß alle Toolkit-Funktionen auf dieses Programm angewendet werden können.

Der Assembler eignet sich zur verketteten Assemblierung von Diskette. Näheres dazu im Kapitel "Pseudo-Opcodes".

Im Anhang finden Sie ein Assembler-Listing, in dem viele der auf den folgenden Seiten aufgeführten Befehle und Bedienelemente enthalten sind.

Starten des Assemblers

Der Assembler wird durch den Befehl ASSEMBLER oder mit SYS 57347 aufgerufen. Diese Anweisung muß in der ersten zu assemblierenden Zeile stehen.

Formatierung

Eine Zeile innerhalb des Quellprogramms sieht wie folgt aus:

ZN (LABEL) (Opcode) (Ausdruck) (; Kommentar) (:)

Die in Klammern angegebenen Ausdrücke sind optional.

ZN ist eine Zeilennummer.

LABEL eine Zeichenkombination von bis zu 32 Zeichen Länge. Das erste Zeichen muß ein Buchstabe sein. Nach einem LABEL muß immer ein Opcode folgen.

Opcode ist ein Befehl aus dem MOS-Standard-Befehlssatz, oder ein illegaler Opcode (s. Liste) oder ein Pseudo-Opcode.

Ausdruck muß bei Mehrbyte-Befehlen angegeben werden (s. Kapitel "Ausdrücke").

;
: das Semikolon entspricht dem REM im Basic. Alles darauf folgende wird bis zum Zeilenende oder einem Doppelpunkt ignoriert. Dieses Zeichen darf nicht direkt nach einem Label stehen, nach einem Label muß immer erst ein Opcode folgen.

:
: der Doppelpunkt leitet wie im Basic eine neue Befehlsfolge ein.

Ausdrücke

Ein Ausdruck ist wie folgt aufgebaut:

(Funktion1) Operand1 (Funktion2 Operator2)

Funktion1 kann sein:

- ! der auf ! folgende Ausdruck wird immer absolut behandelt.
Beispiel: LDA !\$01 wird assembliert wie LDA \$0001
- < ergibt das Low-Byte des folgenden Ausdrucks.
Beispiel: LDA <\$0022 wird assembliert wie LDA \$22.
- > ergibt das High-Byte des folgenden Ausdrucks:
Beispiel: LDA >\$0022 wird assembliert wie LDA \$00.
- dreht das Vorzeichen um.
- () die Klammerung erfolgt wie bei normalen arithmetischen Ausdrücken. Nicht geklammerte Ausdrücke werden von links nach rechts abgearbeitet.

Operand1 kann sein:

- eine dezimale Zahl, z. B. 123, 65535.
- eine hexadezimale Zahl, z. B. \$01, \$D000.
- eine Dual-Zahl, z. B. %01, %1111100.
- ein ASCII-Zeichen, z. B. "a", "A", "#".
- ein *, dann wird der Wert des Programmcounters eingesetzt.
- ein LABEL.
- oder ein weiterer Ausdruck.

Funktion2 kann sein:

- a) eine Rechenfunktion : +,-,*,
- b) eine logische Funktion:
 - & : und
 - ! : oder
 - ↑ : exklusive oder
 - < : schiebe linken Wert nach links sooft rechter Wert angibt.
 - > : schiebe linken Wert nach rechts sooft rechter Wert angibt.

Nachfolgend finden Sie einige Beispiele für Ausdrücke:

```
LDA #$00+3
ADC $FFFF-1*3
LDA LABEL1-LABEL2
EOR %1111111<1
LDA <LABEL↑%111111-(3*$2E&("A"! "B")+3)
LDA *-3,x
LDA ("a"+TAB),y
...
```

Pseudo-Opcodes

=
weist einem LABEL einen Wert zu. Beispiel: ZAEHLER = 10.

←
weist einem schon existierenden LABEL einen neuen Wert zu.

*=
weist dem Programmzähler einen Wert zu. *= muß vor der ersten zu assemblierenden Anweisung stehen. Beispiel: *= \$8000 bewirkt, daß Ihr Programm ab \$8000 assembliert wird.

ASC "TEXT"
Der Text wird an der Programmzählerposition eingefügt.

BYT Ausdruck (, Ausdruck) ...
Der Wert des Ausdrucks wird an der Programmzählerposition eingefügt. Der Ausdruck muß einen 1-Byte-Wert ergeben.

END (GA, "FILENAMEN")
Der Befehl END zeigt dem Assembler, daß er hier die Assemblierung beenden soll. Nach dem zweiten Pass geht er zurück ins Basic. Anschließend können Basic-Zeilen folgen. Der in Klammern angegebene Ausdruck muß eingegeben werden, wenn mehrere Files für die Assemblierung verknüpft wurden (siehe FIL). GA ist die Gerätenummer (z. B. B für Floppy-Disk), "FILENAMEN" ist der Name des Files, mit welchem die Assemblierung begonnen wurde.

FIL GA, "FILENAMEN"
Wenn mehrere Files verkettet assembliert werden sollen, muß bei allen Files (außer dem letzten) die Anweisung FIL statt END am Ende des Files stehen. Dann wird das nach FIL angegebene File nachgeladen und dort weiter assembliert. Das letzte File muß mit END GA, "Name des ersten Files" abgeschlossen werden.

GOT Ausdruck
Die Assemblierung wird bei der in Ausdruck angegebenen Zeile fortgesetzt. GOT wird meist in Verbindung mit IF eingesetzt.

IF Ausdruck
Dieser Befehl ermöglicht eine bedingte Assemblierung. Wenn der Ausdruck ungleich Null ist, wird der folgende Code assembliert, ansonsten wird er ignoriert (siehe auch GOT).
Beispiel: 100 IF Ausdruck : GOT 80

WOR Ausdruck (, Ausdruck) ...
Der Wert des Ausdrucks (2 Byte!) wird an die Programmzähler-Position eingefügt.

OPT xxx (,xxx)...

xxx kann sein:

- p : das Assemblerlisting wird auf dem Bildschirm ausgegeben.
- pn : das Assemblerlisting wird über den Kanal mit der logischen
 Filenummer n ausgegeben. Dazu muß vor Aufruf des Assemblers
 der Kanal geöffnet werden.
 Beispiel Druckerausgabe:
 10 OPEN 1,4
 20 ASSEMBLER
 30 OPT P1

- o : Das erzeugte Assemblerprogramm wird in einen Puffer abgelegt.
 \$F7, \$FB enthalten die Anfangsadresse des Puffers, \$F9, \$FA
 die Endadresse. Diese Adressen müssen vorher gesetzt werden.
- oo : Das erzeugte Assemblerprogramm wird ab der im Programmzähler
 enthaltenen Adresse abgelegt.
- n : die Ausgaben werden unterbrochen.

Die Pseudo-Opcodes können mit einem Punkt eingeleitet werden
(Beispiel: .OPT statt OPT), dies sorgt für eine bessere
Übersichtlichkeit des Quellfiles.

Fehlermeldungen des Assemblers

Syntax Fehler

- 0 : Zeile enthält LABEL ohne Opcode.
- 1 : nicht definierter Opcode.
- 2 : unzulässige Adressierungsart für diesen Opcode.
- 3 : unbekannte Funktion im Ausdruck.
- 4 : unpaarige Klammern.
- 5 : falsches Zeichen im Ausdruck.
- 6 : Komma fehlt.
- 7 : nicht definierter Pseudo-Opcode.
- 8 : Symbol beginnt nicht mit einem Buchstaben.
- 9 : falsche Adressierungsart.

Sonstige Fehlermeldungen

- branch out of range: relativer Sprung über mehr als 128 Byte.
- device not present: Gerät nicht vorhanden.
- xxxx Disk: Diskettenfehler.
- illegal quantity: Wert liegt außerhalb des erlaubten
 Bereichs.
- out of memory: der Puffer ist voll (bei OPT o).
- overflow: Zeile ist zu lang.
- redefinition: dem LABEL ist schon ein Wert zugewiesen.
- sym table overflow: Überlauf der Symboltabelle.
- undef'd statement: Ein LABEL im Ausdruck ist nicht definiert
 oder die Zeile beim GOT gibt es nicht.

4) Floppy-Funktionen

Formel-64 rüstet Ihren Rechner mit einem parallelen Datenbus zur Floppy-Station aus. Dies ermöglicht bis zu 16-fach schnelleres Laden und Speichern von Programmen.

Der Faktor 16 läßt sich allerdings nur erreichen, wenn die Disketten ein spezielles Format bekommen. Dies geschieht automatisch bei der Formatierung mit dem Befehl HEADER.

Vorhandene Programmdisketten lassen sich durch einen BACKUP auf das neue schnelle Format bringen. Dazu ist beim BACKUP der Parameter "f" für schnell (fast) bei der Abfrage "Format" einzugeben.

Bei einer Veränderung des Diskettenformats (langsam auf schnell oder umgekehrt) dauert der Backup länger als bei einem Backup auf das gleiche Format.

Es ist wichtig, daß bei BACKUP immer mit zwei Disketten gearbeitet wird. Wenn versucht wird, einen Backup auf dieselbe (Master-) Diskette durchzuführen, werden die Programme auf ihr zerstört.

Disketten mit dem schnellen Format können ohne Probleme auch ohne Formel-64 geladen und beschrieben werden.

Von Disketten, die nicht auf das schnelle Format gebracht wurden, werden Programme mit etwa 6-facher Geschwindigkeit geladen.

Das Laden und Speichern von Dateien (relativ und sequentiell) wird nicht beschleunigt, dies kommt der Kompatibilität zugute.

Bei manchen mehrteiligen Programmen kann es vorkommen, daß Teile mit normaler Geschwindigkeit geladen werden, obwohl sie als PRG gekennzeichnet sind. Das liegt daran, daß diese Files dann als Dateien nachgeladen werden, welche ja nicht beschleunigt werden.

Wie im Kapitel "Einbau" schon erwähnt, erhöht der Anschluß des einzelnen Kabels im Rechner die Kompatibilität. Sollten trotzdem mit einigen (speziell mehrteiligen) Programmen Schwierigkeiten auftreten, so können diese in manchen Fällen beseitigt werden, wenn man vor dem Laden des ersten Programmteils den Befehl OPEN 1,8,2 eingibt.

Wenn bei Diskettenfehlern der Status des Laufwerks abgefragt wird, können zwei neue Fehlermeldungen auftreten, nämlich die Fehler-Nummern 80 und 81. Fehlernummer 80 bedeutet, daß ein Fehler beim Lesen aufgetreten ist, die Nummer 81 besagt, daß beim Schreiben eines Programms ein Fehler aufgetreten ist.

Aus technischen Gründen wird beim Laden eines Programms der Bildschirm für den Bruchteil einer Sekunde ausgeschaltet (ein kurzes Blinken), während des Abspeicherns eines Programms bleibt er ganz ausgeschaltet.

Geschwindigkeitsregulierung des Laufwerks

Sollte beim Formatieren von Disketten mit dem Befehl HEADER häufig ein READ ERROR gemeldet werden oder beim BACKUP ein NO SYNC ERROR, so liegt dies mit großer Wahrscheinlichkeit an einer zu hohen Umdrehungsgeschwindigkeit der Floppy.

Die Geschwindigkeit der Floppy läßt sich jedoch leicht selbst einstellen:

zunächst ist der Deckel des Laufwerks abzunehmen. Nach Lösen von 6 Schrauben kann dann das Laufwerks-Chassis vom Unterteil des Gehäuses abgenommen werden. Wenn man es auf seine Seite stellt, kann man auf die Unterseite des Chassis schauen.

Dort finden Sie eine Stroboskop-Scheibe und ein kleines Trimm-Potentiometer (s. Bild 1 im Anhang). Nun legen Sie eine leere Diskette ein und formatieren sie mit dem Befehl SEND"N:NAME,XX".

Die Stroboskopscheibe beginnt sich dann zu drehen. Wenn diese jetzt mit einer normalen Glühbirne beleuchtet wird, müssen die Markierungen auf der Scheibe scheinbar zum Stillstand kommen. Ist dies nicht der Fall, kann mit einem kleinen Schraubenzieher am Trimpotentiometer ("Regler") die Umdrehungs-Geschwindigkeit nachreguliert werden. Anschließend ist das Chassis wieder ins Gehäuse einzubauen.

ACHTUNG: beim neuen 1541-Laufwerk mit dem Knebelverschluß kann die Geschwindigkeit nicht mehr auf diese Art reguliert werden!

5) Centronics-Schnittstelle

Formel-64 besitzt eine integrierte Centronics-Schnittstelle. Zur Nutzung dieser Schnittstelle ist auf jeden Fall das von uns angebotene Centronics-Kabel (Best.-Nr.: 406-cdk) notwendig.

Eins der üblichen Userport-Centronics-Kabel kann nicht verwendet werden, da unser Centronics-Kabel mit einem speziellen Anschluß direkt auf dem Modul angeschlossen wird (wie das Floppy-Kabel).

Dadurch bleibt der Userport für andere Anwendungen frei. Das Centronics-Kabel kann einzeln nachbestellt und selbst eingebaut werden.

Formel-64 steuert sowohl die parallele Druckerschnittstelle als auch den seriellen Druckerausgang an.

Bei Verwendung des OPEN-Befehls können durch verschiedene Sekundäradressen (sa) unterschiedliche Übertragungsmodi eingestellt werden.

Syntax: OPEN 1,4,sa mit:
keine sa: Ausdruck nach Text- oder Graphikmodus, wie auf dem Bildschirm.
sa = 0 : keine Umkodierung der Zeichen (z. B. für Graphikdruck nötig).
sa = 1-6 : druckt Original-Commodore-Zeichen in unterschiedlicher Breite aus. sa=1-6 entspricht den Druckdichtewerten 1-6, wie sie z. B. mit dem Befehl "ESC *" am Drucker eingestellt werden können.
sa = 8-13 : wie Sekundäradresse 0.
sa = 14 : es wird nur der serielle Druckerausgang angesteuert.
sa = 15 : es wird immer im Textmodus gedruckt.
Die Sekundäradressen 1-6 ergeben einen sinnvollen Ausdruck nur auf EPSON-FX/RX-80 kompatiblen Druckern.

Anhang: Die illegalen Opcodes

Illegale Opcodes sind Opcodes, welche in der MOS-Applikation nicht beschrieben sind, aber trotzdem von vielen 6510- bzw. 6502-CPU's verarbeitet werden können.

Die Namen für diese Opcodes sind teilweise selbst gewählt, da sie eben nicht genormt sind.

WICHTIG: es können nicht alle CPU's alle illegalen Opcodes verarbeiten. Für die Richtigkeit und Vollständigkeit dieser Opcodes können wir keine Gewähr übernehmen.

SLO entspricht der Befehlsfolge ASL:DRA.
Adressierungsarten: (ind,x); zero; absolut; zero,x; absolut,x

RLA entspricht der Befehlsfolge ROL:AND.
Adressierungsarten: (ind,x); zero; absolut; zero,x; absolut,x

SRE entspricht der Befehlsfolge LSR:EOR.
Adressierungsarten: (ind,x); zero; absolut; zero,x; absolut,x

RRA entspricht der Befehlsfolge ROR:ADC.
Adressierungsarten: (ind,x); zero; absolut; zero,x; absolut,x

STB speichert das Ergebnis von a AND x.
Adressierungsarten: (ind,x); zero; absolut; zero,y; absolut,y

LDB entspricht der Befehlsfolge LDA:LDX.
Adressierungsarten: (ind,x); zero; absolut; zero,y; absolut,y

DCC entspricht der Befehlsfolge DEC:CMP.
Adressierungsarten: (ind,x); zero; absolut; zero,x; absolut,x

INS entspricht der Befehlsfolge INC:SBC.
Adressierungsarten: (ind,x); zero; absolut; zero,x; absolut,x

ASR entspricht der Befehlsfolge AND:LSR a.
Adressierungsart: immediate.

ARR entspricht der Befehlsfolge AND:ROR a, wobei Bit 7 mit Carry ausgetauscht wird. Adressierungsart: immediate.

ATX entspricht der Befehlsfolge ORA \$EE:AND :TAX.
Adressierungsart: immediate.

SFX: entspricht folgendem Ausdruck: x AND HI+1:STX. HI ist das High-Byte der Adresse. Adressierungsart: absolut,y.

SFB entspricht der Befehlsfolge x and HI+1:STB.

FAS absolut,y lädt das Stack-Register mit dem Ergebnis von Akku und x-Register und speichert bei absolut,y eine Null.

LBS entspricht der Befehlsfolge LDA :LDX :LDS .
Adressierungsart: immediate.

```
100 rem *****
110 rem *      grafik-demo      *
120 rem *****
130 :
140 grafik 1
150 multi 1,2,5,0
160 clear
170 :
180 box 20,20,279,159,3
190 :
200 hprint 50, 3,1,"F(x) = sin (x)
210 hprint 72,188,1,"GRAPHIKDEMO"
220 vprint 2, 35,1,"F O R M E L - 6 4"
230 vprint304, 35,1,"F O R M E L - 6 4"
240 :
250 for y= 40 to 170 step 20
260 : line 22,y,296,y,2
270 next
280 :
290 for x= 40 to 290 step 20
300 : line x,21,x,178,2
310 next
320 :
330 line 21,100,298,100,3
340 line 160,21,160,178,3
350 :
360 hprint 144,136,3,"1"
370 hprint 128, 56,3,"-1"
380 vprint 34, 76,3,"-3"
390 vprint 74, 76,3,"-2"
400 vprint 114, 76,3,"-1"
410 vprint 194, 84,3,"1"
420 vprint 234, 84,3,"2"
430 vprint 274, 84,3,"3"
440 :
450 plot 21,100,1
460 :
470 for x= 0 to 276 step 2
480 : y=100+75*sin(pi*4*x/278)
490 : draw x+22,y,1
500 next
510 :
520 for r= 5 to 30 step 5
530 : circle 50, 50,r ,30 ,1
540 : circle 270,150,30, r ,1
550 next
560 :
570 for x=24to 60 step4
580 : box 160, 20,x-20,x,1
590 : box 100,140, x,x-20,1
600 next
610 :
620 fill 0,0,2
630 :
640 hprint 50,3,2,"F(x) = sin (x)
650 for t=0 to 200:next
660 hprint 56,3,4,"press any key"
670 get a$: if a$="" then 650
680 invert
690 for t=0 to 3000 : next
700 text
```

ready.

```

100 assembler
110 .opt oo ; direkt ins ram assemblieren
120 *= $c000 ; ab c000 ablegen
130 ;
140 ;*****
150 ;*      assembler-demo      *
160 ;*****
170 ;
180 videoram = $400
190 zaehler  = $22
200 bsout    = $ffd2
210 ;
220 lda #147 : jsr bsout ; schirm loeschen
230 lda #14  : jsr bsout ; kleinschreibung
240 ;
250 lda #"*" : jsr writescreen
260 lda #" " : jsr writescreen
270 lda #"*" : jsr writescreen
280 ;
290 ldx #$00
300 ausgabe lda text,x      ; text auf bildschirm schreiben
310 beq invert
320 sta videoram+12+(12*40),x
330 inx : jmp ausgabe
340 ;
350 invert ldx #<videoram ; bild invertieren
360 stx zaehler
370 ldx #>videoram
380 stx zaehler+1
390 ldy #0
400 ;
410 inv1 lda (zaehler),y
420 eor #%10000000 ; invertieren
430 sta (zaehler),y
440 dec $55 ; zeitschleife
450 bne *-2
460 inc zaehler
470 ;
480 bne inv1
490 inc zaehler+1
500 ;
510 ldx zaehler+1
520 cpx #$08
530 bne inv1
540 beq invert
550 ;
560 writescreen ldx #8 ; 8 zeilen mit zeichem im
570 ws2 ldy #40      ; akku auf bildschirm schreiben
580 ws1 jsr bsout
590 dey
600 bne ws1
610 dex
620 bne ws2
630 rts
640 ;
650 text asc "assembler-demo"
660 byt 0
670 end
680 sys 4096*12

```

ready.

Bild 1

Untersseite Floppylaufwerk 1541

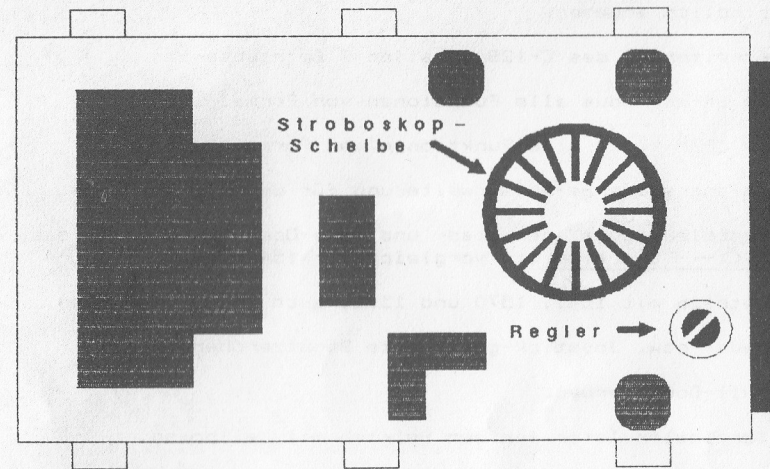


Bild 2

Kiemmenanschluss im C-64

