

Ein Wolf im Schafspelz - der 264



Bild 3. Der Maschinesprache-Monitor ist schon eingebaut



Bild 1. Commodore 264

Die 64'er Redaktion hatte als erste die Möglichkeit, den Commodore 264 zu testen. Was uns überzeugt, ist das eingebaute Basic. Der zusätzliche Maschinensprache-Monitor rundet das positive Bild von diesem neuen Computer ab.

So beliebt und verbreitet der Commodore VC 20 und C 64 auch sind, einige Dinge hat man doch bemängelt, und das zu Recht. Vor allem ist es ihr spartanisches Basic. Befehle, die andere Computer bereits »in sich« haben, mußten beim VC 20 und C 64 »soft« umgangen werden. Und das war natürlich auch die Geburtsstunde für Basicerweiterungen, Toolkits, Assemblern, Grafikmodule und was da sonst noch auf den Markt gekommen ist.

Das alles weiß Commodore natürlich auch. Viele andere Computerhersteller geben Ihren Computern ein wesentlich komfortableres Basic mit. Mit dem neuen C264 (Bild 1) setzt Commodore zwar keinen neuen Standard, aber es wird gleichgezogen. Und berücksichtigt man zum Beispiel den komfortablen Editor, meiner Meinung nach der Beste in dieser Leistungsklasse, übrigens der gleiche wie der des C64, und den eingebauten Maschinensprache-Monitor, Assembler und Dis-

assembler, so findet der 264 seinen Platz in Puncto Bedienungs-freundlichkeit ganz vorne, an der Spitze. Dabei ist die schon in der Ausgabe 4, Seite 9 ff erwähnte Built-In-Software noch gar nicht berücksichtigt.

Bild 2. Die Built-In-Software des C 264



Built-In-Software

Laut Commodore wird es folgende »eingebaute« Programme geben (Bild 2):
 * Magic Desk, eine Simulation eines kompletten Büroschreibtisches, mit den Funktionen Briefe schreiben, Rechnen, Schriftverkehrsablage, Zeitanzeige und so weiter, ein Programm mehr für den Home und privaten Einsatz (Wir werden noch einen ausführlichen Test dazu bringen).
 * Superscript, eine semiprofessionelle Textverarbeitung

- * Commodore 3-Plus-1
- * Logo, eine Programmiersprache vor allem für den Anfänger
- * Pilot, eine weitere Programmiersprache
- * Easycalc 264, ein Tabellenkalkulationsprogramm
- * COM 264
- * Financial Advisor

Der Commodore 264 wird mit einem dieser Programme, das fest installiert wird, ausgeliefert. Der Käufer gibt bei seiner Bestellung an, welche Software er haben will. Diese Programme lassen sich dann kinderleicht über die Funktionstasten ein- und auch wieder ausschalten.

Alle restlichen oben angegebenen Programme können dann über den Expansionport als Modul eingesteckt und benutzt werden.

Da ein Wechseln der Built-In-Software nicht so einfach möglich sein dürfte (der Computer muß geöffnet, das ROM ersetzt werden, Garantiebedingungen müssen beachtet werden), ist schon beim Kauf des C264 eine wichtige Entscheidung zu treffen. Hoffentlich werden diese Überlegungen durch eine umfassende Beratung durch den Verkäufer und Händler unterstützt. Sobald der C264 mit dieser Software auf dem deutschen Markt erhältlich sein wird, werden wir Ihnen wichtige Informationen und Hilfestellungen geben. Diese Programme werden oder sind nämlich zum Teil auch schon für den C 64 erhältlich.

Doch kommen wir zum Basic des C264: Wie schon kurz angedeutet, besitzt der Commodore 264 ein recht komfortables Basic. Es unter-

weilige Belegung dieser Tasten kann mit der Anweisung DISPLAY abgerufen werden. Eine Besonderheit ist die Help-Funktion. Wurde ein Programm mit einer Fehlermeldung abgebrochen, ist es möglich, sich die fehlerhafte Programmzeile durch Drücken der Help-Taste (zum Beispiel Funktionstaste f8) revers anzeigen zu lassen. Leider wird nicht der Fehler in der Zeile selbst angezeigt, etwa wie bei ExBasic-Level II, wo der Cursor direkt auf den Fehler zeigt. Der Wert dieser Help-Funktion ist eigentlich nicht ganz einsehbar, da auch die normale Basic-Fehlermeldung auf die fehlerhafte Zeile hinweist.

Grafik-Befehle

Es ist möglich, drei Bildschirmmodi darzustellen: Den hochauflösenden Grafik-Modus mit einer Auflösung von 200 x 320 Punkten, den Multicolor-Modus mit einer Auflösung von 160 x 200 Punkten und den normalen Textmodus mit 40 x 25 Zeichen. Diese verschiedenen Modi können, und das ist ein echtes Plus, gleichzeitig dargestellt werden. Sie

werden durch den GRAPHIC-Befehl definiert. Und das wird folgendermaßen gelöst:

Modus 0 = Textdarstellung, das ist der Standard-Modus, in dem sich der C264 nach dem Einschalten befindet.

Modus 1 = Hochauflösende Grafik
Modus 2 = Hochauflösende Grafik + Textdarstellung in den unteren fünf Zeilen

Modus 3 = Multicolor Grafik
Modus 4 = Multicolor Grafik + Text in den unteren fünf Zeilen

Es ist auch möglich, im hochauflösenden Grafikbereich Text hineinzusetzen (siehe CHAR-Befehl). Im Modus 3 und 4 kann vollkommen unabhängig vom Hires-Bildschirm, also ohne ihn zu beeinflussen, irgendein Text in den letzten fünf Zeilen des Bildschirms dargestellt werden. Es ist somit möglich, im oberen Teil eine Grafik zu setzen und gleichzeitig unten durch den LIST-Befehl ein Listing ablaufen, oder zum Beispiel eine Tabelle sich anzeigen zu lassen. Durch Wahl der Grafik-Modi 1 bis 4 wird ein Speicherplatz von 10 KByte reserviert. Und mit rund 60 KByte frei programmierbarem RAM-Speicher ist das keine nennenswerte Einschränkung.

Der C264 ist in der Lage, 128 Farben darzustellen. Genauer gesagt, 16 Farben in je acht verschiedenen Abstufungen (Bild 8). Diese Farben können direkt angesprochen werden, entweder mit der Control-Taste im Direkt-Modus oder durch einfachen Basic-Befehl im Programm. Die Grafik-Befehle sehen Sie in Bild 5, 6 und 9.



Bild 4. Ein Beispiel für geschachtelte Schleifen

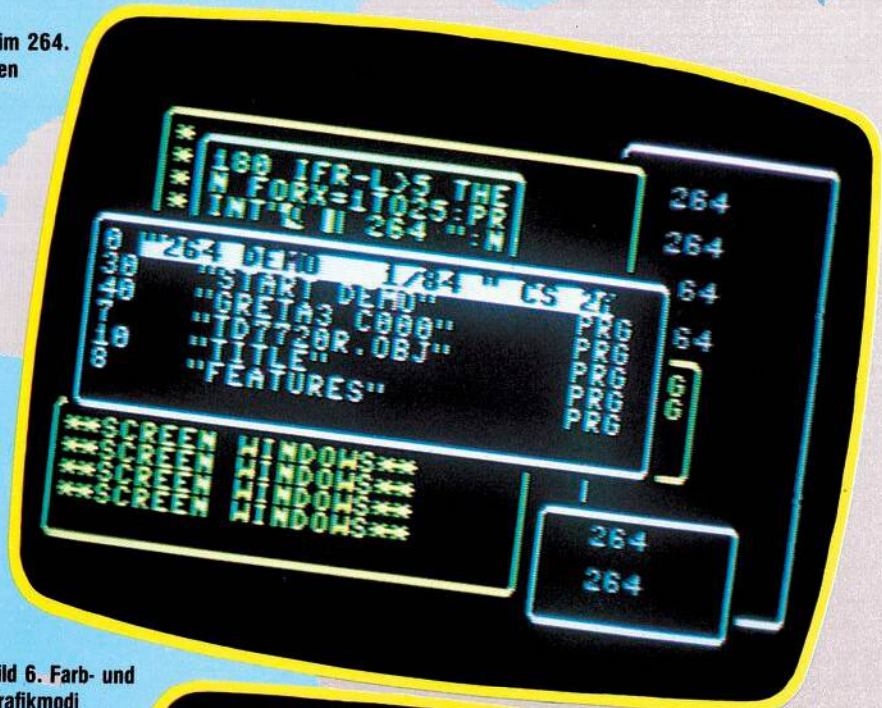
stützt jetzt nicht nur die Tonausgabe, auch umfangreiche und mächtige Grafik-Befehle sowie Hilfsfunktionen, Sound- und Diskettenbefehle wurden zusätzlich zum bisherigen Commodore-Basic implementiert.

Viele Befehle, nämlich alle, die sich im Direkt-Modus ausführen lassen, können über die Funktionstasten programmiert werden. Sie sind frei programmierbar. Das funktioniert genauso wie beim Simons-Basic mit dem Key-Befehl. Die je-

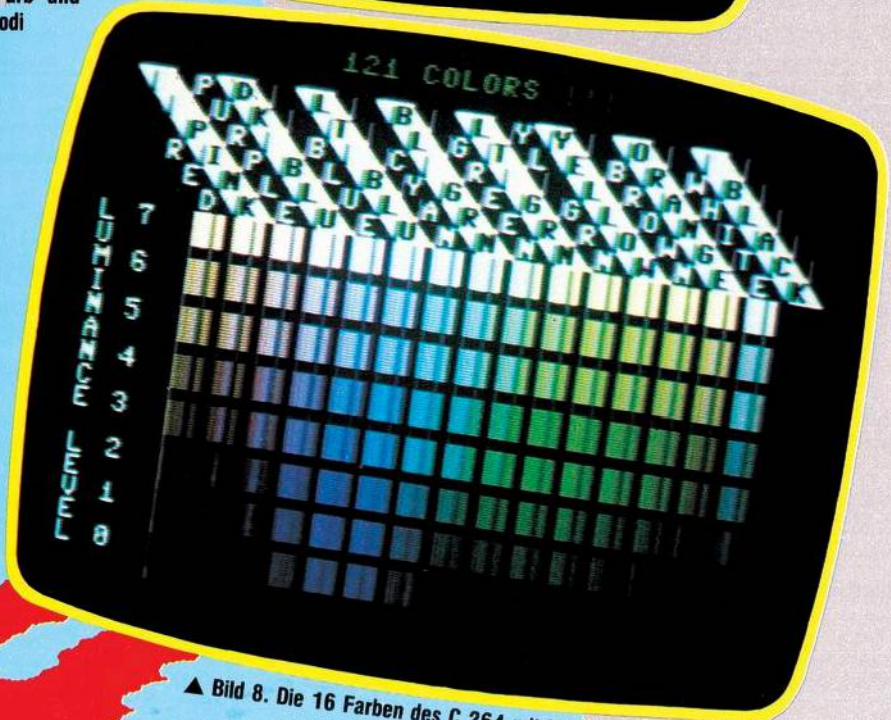


Bild 5. Die wichtigsten Grafik-Befehle

Bild 7. Windowing beim 264.
In jedem Fenster laufen
Aktionen unabhängig
voneinander ab. ▶



◀ Bild 6. Farb- und
Grafikmodi



▲ Bild 8. Die 16 Farben des C 264 mit je 8 Abstufungen

COLOR,	Region, Farbe, Farbintensität
	Region (0 bis 4)
	0 = Hintergrundfarbe
	1 = Zeichenfarbe
	2 = Multicolor 1
	3 = Multicolor 2
	4 = Rahmenfarbe
	Farbe (1 bis 16)
	Farbintensität (0 = dunkel, bis 7 = hell)
GRAPHIC	Modus, löschen
	Modus (0 bis 4)
	0 = Text
	1 = Hires
	2 = Hires + Text (unterteilter Bildschirm)
	3 = Multicolor 1
	4 = Multicolor 2 (unterteilter Bildschirm)
	löschen (0 oder 1)
	0 = nicht löschen
	1 = löschen

Der C264 hat zum Teil sehr mächtige Grafik-Befehle. Sie können sicher schon abschätzen, wie einfach jetzt Grafik erstellt werden kann. Sicher haben Sie bemerkt, daß keine Rede mehr von Sprites ist. Und in der Tat, die beim C 64 so beliebten Sprites gibt es beim C264 nicht mehr. Man kann sich zwar bestimmte Bereiche des Bildschirms reservieren und auch verschieben, dabei ist man auch nicht beschränkt auf die bei den Sprites begrenzten Größe von 24 x 21 Punkten. Ob jedoch mit Merkmalen wie Kollisionsabfrage, etwas Entsprechendes wie ein Hintergrund- beziehungsweise Vordergrund-Sprite definiert werden kann, mit allen sich daraus ergebenden Möglichkeiten für Spiele, ist zumindest zweifelhaft. Aus dem Basic-Wortschatz und dem Handbuch war das nicht ersichtlich.

Apropos Handbuch: Auch dieses

▼ Bild 9. Die Grafik-Befehle des 264

CHAR	= fügt Texte direkt in Hires-Grafik ein
BOX	= zeichnet Rechtecke
CIRCLE	= zeichnet Kreise, Ellipsen
PAINT	= füllt einen bestimmten Bereich mit einer bestimmten Farbe (wie in Simons-Basic)
SCALE	= Die Skalierung der Bit-Muster der Hires- und Multicolor-Grafik kann geändert werden.
DRAW	= Zeichnet Punkte und Linien
LOCATE	= setzt den Anfangspunkt eines Zeichenbefehls fest
SSHAPE	= speichert einen zu bestimmenden Teil der Hires Grafik auf Band oder Diskette
GSHAPE	= Diesen Teil kann man mit GSHAPE wieder an irgendeiner Stelle des Grafik-Bildschirms auszeichnen lassen. GSHAPE erlaubt mehrere Modi: Modus (0 bis 4)
	0 = wie gespeichert
	1 = Reverse Darstellung
	2 = OR — verknüpft mit der Umgebung
	3 = AND — verknüpft mit der Umgebung
	4 = XOR — verknüpft mit der Umgebung
RCLR(N)	= Ordnet einer bestimmten Bildschirmzone N (0 bis 4) einer bestimmten Farbe zu (Zonen siehe COLOR)
RDOT(N)	= gibt die momentane Position des Grafik-Cursors an
RGR(N)	= man erhält den Grafik-Modus in dem man sich befindet
RLUM(N)	= gibt die zugewiesene Farbintensität der Farbzone N an

bisher stiefmütterlich behandelte⁵ Thema reiht sich in der Reihe der positiven Eindrücke nahtlos ein. Es enthält eine umfassende Erklärung aller Basic-Befehle, mit zahlreichen Beispielen, einer kompletten Liste und Beschreibung der Monitor-Befehle und mehreren Anhängen. Dieses Handbuch läßt sicherlich nicht so viele Fragen offen wie man es bisher von Commodore gewohnt war.

Kommen wir zum nächsten Thema.

Sound-Befehle

Was man beim Basic Positives gemacht hat, ist genau entgegengesetzt den Sound-Möglichkeiten des C264. Es gibt nämlich nur zwei Befehle, die die »musikalischen« Eigenschaften des C264 unterstützen (Bild 10). Mehr sind auch gar nicht notwendig, da die Sound-Fähigkeit

VOL (0 bis 7) = reguliert die Lautstärke der erzeugten Töne
SOUND x,y,z = erzeugt einen Ton beziehungsweise ein Geräusch
 x (1 bis 3)
 1 = hohe Stimme von sehr hoch bis mitteltief
 2 = tiefe Stimme von mittelhoch bis sehr tief
 3 = Geräusche (zum Beispiel Donnern)
 y (0 bis 1023) = Notenwert, (0=tiefer, 1023=hoher Ton)
 z (0 bis 65535) = Zeitdauer des erzeugten Tones

Bild 10. Die Sound-Befehle des C 264

ten des C264 gegenüber denen des C 64 stark eingeschränkt wurden.

Ich selbst habe zwischen den beiden Stimmen (x=1 und x=2) allerdings keinen Unterschied feststellen können. Die fantastischen Synthesizer-Eigenschaften des C 64 wurden fast völlig unter den Tisch gekehrt. Ob diese Einschränkung negativ zu bewerten ist, bleibt dahingestellt. Denn Commodore hatte mit diesem Modell nicht vor, seinen Renner C 64 abzulösen.

Dafür können sich die Hilfsfunktionen des C264 wieder sehen lassen. Es gibt vier Hilfs-Funktionen, die ein effektives Editieren von Basic-Programm stark unterstützen (Bild 12)

Eine andere Gruppe sind die Disketten-Befehle. Sie entsprechen

denen der CBM 8000er Reihe (Bild 11.)

Ein umständliches Öffnen der Datenkanäle entfällt somit. Allerdings sind die vom C 64 bekannten Disketten-Befehle auch anwendbar. Der Fehlerstatus eines Laufwerkes kann einfach über die Variablen DS und DS\$ angezeigt werden. Sie enthalten die Fehlerart und die entsprechende Fehlermeldung.

Aber auch das Basic unterstützt die Fehlererkennung im Programm. Es gibt die Möglichkeit, dem Programm »Fallen« zu stellen, Commodore nennt es »trapping«.

Mit diesen beiden zusammengehörigen Befehlen (TRAP und RESUME, siehe Bild 13) kann das Pro-

DLOAD/DSAVE DIRECTORY = laden/speichern von Programmen
 = Der Befehl dürfte klar sein. Das Anzeigen des Directorys kann abgebrochen oder verlangsamt werden. Auch eine Auswahl der anzuzeigenden Programme ist möglich (zum Beispiel alle Programme, die mit A beginnen)
COLLECT COPY = kopiert ein Programm von einer Diskette auf eine andere bei einem Doppellaufwerk oder kopiert ein Programm innerhalb einer Diskette unter einem anderen Namen beim Einzellaufwerk.
RENAME SCRATCH = Ändert den Namen eines Files
 = löscht ein File auf einer Diskette

Bild 11. Diskettenbefehle wie bei der 4/8000-Serie

AUTO Abstand = AUTO 20 erzeugt nach jedem RETURN eine neue Zeilennummer, die um 20 höher ist als die Zeile davor
RENUMBER x,y,z = der komfortabelste RENUMBER-Befehl, der mir bekannt ist.
 x = neue Anfangszeile
 y = Abstand
 z = ab welcher bisherigen Zeile
 Der Befehl RENUMBER 1000, 10, 100 nummeriert ein Basic Programm ab der bisherigen Zeile 100 im Abstand von 10 neu durch, wobei die erste neue Zeile mit 1000 beginnt. Alle Sprungadressen (GOTO, GOSUB, ON x GOTO/GOSUB etc.) werden berücksichtigt. Selbst wenn diese Sprungbefehle vor z liegen.
DELETE Bereich = löscht Programmzeilen im angegebenen Bereich
TRON, TROFF = schaltet den Trace Modus ein oder aus. Die jeweils bearbeitete Zeile wird angezeigt.

Bild 12. Diese Befehle helfen bei der Programmerstellung

TRAP = fängt einen Fehler ab. Der Fehler kann mit den Fehlervariablen ER, ERR\$ und EL ausgegeben werden.
 ER = gibt den zuletzt aufgetauchten Fehler aus
 ERR\$ = enthält die entsprechende Basic-Fehlermeldung
 EL = gibt die Zeile an, in der sich der Fehler befindet
RESUME = Mit diesem Befehl wird das Programm fortgesetzt, ohne daß der Fehler zum Absturz des Programms führt. Hier besteht die Möglichkeit, das Programm an irgendeiner Stelle weiterlaufen zu lassen indem die entsprechende Zeilennummer angegeben wird. Durch RESUME NEXT wird das Programm direkt hinter der fehlerhaften Zeile fortgeführt.

Bild 13. Fehlerbehandlung beim neuen Commodore 264

gramm absturzsicher gemacht werden. Auch wenn man zum Beispiel die Stop-Taste betätigt, gibt es keinen BREAK, sondern das Programm wird nach dem Loslassen der STOP-Taste ohne eine Fehlermeldung fortgesetzt. Nur der Fehler FILE NOT FOUND wird nicht berücksichtigt.

Aber es kommt noch besser. Es wurden Befehle implementiert, die das Strukturierte Programmieren unterstützen (siehe Bild 4 und 15).

Im Bild 16 sind die Befehle zusammengefaßt, die sich in keine bestimmte Kategorie eingliedern lassen.

Aus den Bildern erkennen Sie alle Befehle, die das neue Commodore Basic 3.5 bietet. Selbstverständlich kommen noch die vom VC 20/C 64 hinzu. Sie sehen selbst, wie komfortabel es geworden ist. Es läßt kaum noch Wünsche offen. Aber das ist immer noch nicht alles. Wie schon kurz erwähnt, besitzt der Commodore C264 auch noch einen eingebauten Maschinensprache-Moni-

tor. Und das ist eigentlich eine Untertreibung. Denn es gibt nicht nur die sonst üblichen Monitorfunktionen. Auch ein Assemblieren und Disassemblieren ist damit möglich. TEDMON, so wird er genannt, und Basic können nebeneinander laufen und schließen sich nicht gegenseitig aus. Es ist auf einfache Weise möglich, entweder Assemblerprogramme separat oder als Unterprogramm in Basic ablaufen zu lassen. Der TEDMON wird von Basic mit dem Befehl MONITOR aufgerufen. Danach stehen eine ganze Reihe von Funktionen zur Verfügung (siehe Bild 3 und 14).

Betrachtet man das gesamte Gerät mit allen seinen Möglichkeiten wie das vorzügliche Basic, den ein-

gebauten Maschinensprache-Monitor sowie der Built-In-Software, so kann man Commodore zu diesem Computer beglückwünschen. Im Vergleich zum C 64 kann man sagen, daß der C264 zwar in seinen Grafikmöglichkeiten etwas eingeschränkt (keine Sprites) und der Sound-Teil erheblich reduziert wurde, jedoch die Bedienerfreundlichkeit um ein gewaltiges Maß gestiegen ist. Auch das Gehäuse macht einen stabilen Eindruck und ist sehr kompakt. Die Tastatur ist leichtgängig und die Idee mit den Cursorstasten (siehe Bild 1) ist auch nicht übel. Diese Merkmale und der Preis von voraussichtlich um 1200 Mark grenzen dann auch den C264 vom C 64 ab. Der C264 geht eindeutig in Richtung Anwendung. Seine Software (Basic Windowingfähigkeit (siehe Bild 7) und Built-In-Software) unterstützt stark die Programmierung und Nutzung professioneller Software.

Leider setzt auch Commodore, wie so viele europäische und amerikanische Computerhersteller, die unheilvolle Strategie der Inkompatibilitäten fort. Es hätte doch möglich sein müssen, zumindest die Anschlußmöglichkeiten der Peripherie voll kompatibel mit dem C 64 zu machen. (gk)

A = Assemble
C = Compare
D = Disassemble
F = Fill
G = Go
H = Hunt
L = Load
M = Memory
R = Register
S = Save
T = Transfer
X = Exit

— wandelt ein Assemblerprogramm in Maschinencode um
— vergleicht zwei Sektionen des Speichers und teilt die Unterschiede mit
— wandelt Maschinencode um in Assemblerbefehle
— belegt den Speicher mit einem spezifizierten Byte
— startet ein Assemblerprogramm ab einer bestimmten Adresse
— suche den Speicher nach allem Vorkommen eines bestimmten Bytes ab.
— lädt ein Programm von Kassette oder Diskette
— gibt die Inhalte von Speicherstellen an
— gibt die 6802 Register-Werte an
— speichert auf Band oder Diskette
— transferiert einen Teil des Speichers in einen anderen
— verläßt TEDMON

Bild 14. Alle Funktionen des Maschinensprache-Monitors TEDMON

DO UNTIL
DO WHILE
EXIT
LOOP UNTIL
LOOP WHILE
IF THEN ELSE

Mit diesen Befehlen ist ein voll strukturierter Programmablauf möglich. Die Endbedingung kann am Anfang der Schleife (DO UNTIL/WHILE) oder an das Ende (LOOP UNTIL/WHILE) gesetzt werden. Durch EXIT kann die Schleife während eines Schleifendurchlaufs verlassen werden.

Bild 15. Auch strukturierte Programmierung ist möglich

PRINT USING = Es definiert das Format eines Textstrings oder einer Zahlenreihe bei der Ausgabe. Mit ihm wird das Erstellen von Tabellen jeglicher Art zum reinen Vergnügen.
PUDEF = ändert Parameter des PRINT USING-Befehls
INSTR = erlaubt es, zwei Strings ineinander zu verflechten.
DEC = Umwandlung einer Hexadezimalzahl (00 bis FF) in eine Dezimalzahl.
HEX\$ = Umwandlung einer Dezimalzahl (0 bis 65536) in eine Hexadezimalzahl
JOY = Position der Joysticks 1 oder 2
GET KEY = ähnelt dem Get-Befehl, wartet jedoch bis eine Taste gedrückt wird.

Bild 16. Noch einige Befehle