

Der Prozessor des C 64 (MOS 6510) ist als 8-Bit-CPU in der Lage, genau 65536 Speicherzellen zu adressieren. Beim C 64 sind, im Gegensatz zum VC 20, alle diese Speicherzellen mit RAM-Bausteinen bestückt. Trotzdem bleiben nur 3891 Byte zum Programmieren in Basic übrig, denn 24 KByte ROM, 2 KByte für Zeropage und Bildschirm-speicher fordern ihren Tribut. Der Rest ist nur für Programme in Maschinensprache zugänglich. Diesem Manko hilft Business Basic von Kingsoft gründlich ab. Es ist schon beeindruckend, wenn man in der neuen Einschaltmeldung »61183 Bytes free« (Bild) lesen und über dieses Speichervolumen in Basic auch frei verfügen kann.

Insgesamt verfügt der C 64 mit Business Basic über 96 KByte (64 KByte RAM + 28 KByte ROM). Das Geheimnis, das hinter diesem Konzept steckt, ist die konsequente Weiterführung eines schon im Standard-C-64 vorhandenen Prinzips. Es wird, je nachdem welche Funktion benötigt wird, zwischen verschiedenen Speicherbereichen, ROM und RAM umgeschaltet. Für den Computer ist also immer nur der Teil des Speichers »sichtbar«, den er gerade benötigt. Dieses sogenannte »banking« hat aber auch seine Nachteile, denn es treten Geschwindigkeitseinbußen von zirka 18 bis 25 Prozent auf. Dafür wurden aber einige leistungsfähige Befehle und Programmiertricks implementiert, die diesen Nachteil relativieren. So dauert die gefürchtete Garbage Collection jetzt maximal 1,5 Sekunden, statt wie bisher bis zu 20 Minuten. Man kann jetzt also ohne weiteres mit sehr großen Stringfeldern arbeiten.

**Basic-Befehle
wie beim C 16**

Wem die Ablaufgeschwindigkeit der unter Business Basic geschriebenen Programme trotz der Beseitigung der Garbage Collection zu gering ist, kann sie mit einem gleichfalls von Kingsoft angebotenen Spezialcompiler noch um einiges vervielfachen.

Das ist aber nur ein Teil der Vorteile, die das Modul zur Verfügung stellt. Denn neben den genannten Leistungen wird noch eine Basic-Erweiterung geboten, die allerdings speziell die Programmierung von Anwenderprogrammen unterstützt (daher auch der Name des Moduls). Bei der Namenswahl der

**Darf es
etwas
mehr sein?**

**Steckmodule verbrauchen
fast immer Speicherplatz.
Business Basic erweitert
ihn auf 61 KByte RAM
und bietet zudem über
50 neue Befehle.**



Die neuen Befehle

neuen, zusätzlichen Befehle (Bild) wurde darauf geachtet, daß die Syntax und die Tokencodierung der des Basic 3.5 entspricht. Die Programme des C 16 und Plus/4 (beide mit Basic 3.5 ausgestattet) funktionieren aber nur sehr eingeschränkt. Eine Anpassung dürfte aber nicht allzu schwer sein.

Zu den zusätzlichen Befehlen von Business Basic gehört zunächst eine sinnvolle Erweiterung der Fähigkeiten des Basic-Editors. Es gibt Befehle wie AUTO zur automatischen Zeilennummernvorgabe, DUMP zum Anzeigen der Variableninhalte, NUMBER zum Ummumerieren eines Programmes mit Anpassung aller GOTO- und GOSUB-Befehle, KEY zur Funktionstastenbelegung und FIND zum Auffinden bestimmter Programmteile. Interessant ist der CHANGE-Befehl, der ähnlich der FIND-Anweisung bestimmte Stellen des Programmes sucht und gegen neue Ausdrücke ersetzt.

Bei Fehlern im Programmablauf wird die fehlerhafte Zeile angezeigt und der Cursor an die betreffende Stelle gesetzt. Diese Eigenschaften lassen sich durch HELP ON/OFF ein- beziehungsweise ausschalten. Es gibt auch den Befehl TRACE, mit

dem man den Programmablauf anhand der ausgedruckten Zeilennummern verfolgen kann. Dieser Befehl läßt sich auch programmieren, so daß man auch Teile des Programmes gezielt beobachten kann. Das ist besonders dann wichtig, wenn eine lange Leseschleife am Anfang des Programms steht. Außerdem kann man jetzt, wie bei Ex-basic Level II, den Ausdruck des Listings auf dem Bildschirm mit den Cursortasten vor- und rückwärts steuern. Ferner ist der LIST-Befehl ebenfalls programmierbar, so daß kein Abbruch mehr nach der Ausführung dieses Befehls unter Programmkontrolle erfolgt.

Formatierte Ausgabe

Sehr nützlich bei der Arbeit mit Bildschirmmasken sind die Möglichkeiten zur formatierten Ausgabe von Zahlen und Zeichenketten mit PRINT USING, das auch auf dem Drucker funktioniert. Erwähnenswert ist hierbei die Möglichkeit mit PUDEF die Verwendung von Kommas und Punkten bei der formatierten Ausgabe von Zahlen an deutsche Verhältnisse anzugleichen (Dezimalkomma und Tausendertpunkt). Weiterhin kann man den Ort der Bildschirmausgabe mit PRINT AT und der Angabe der gewünschten Zeile und Spalte bestimmen. Diese gesteuerte Ausgabe läßt sich auch in Zeichenketten einbauen, indem man den Zeichencode, gefolgt von der entsprechenden Zeichencodierung für Zeile und Spalte, in einen String einbaut (CHR\$(1)+CHR\$(Zeile)+CHR\$(Spalte)).

Auch Kommandos zur gesteuerten Eingabe von Zeichen fehlen nicht. Der Befehl INLINE etwa gestattet die Eingabe aller Zeichen (also auch Kommas und Doppelpunkte), während der INFORM-Befehl die Cursortasten bei der Eingabe blockiert. Ferner gibt es noch den Befehl GETKEY, der auf ein Zeichen von der Tastatur wartet und die lästige Abfrage mit einer IFTHEN-Schleife erspart.

Selbstverständlich wurden einige Befehle zur Diskettensteuerung eingebaut. Da gibt es zum Beispiel CATALOG zum Anzeigen eines Inhaltsverzeichnis oder DISK zum Senden von Floppy-Kommandos. Der Fehlerkanal kann über die Variablen DS und DS\$ abgefragt werden. Weiterhin wird die Verwendung relativer Dateien mit Befehlen wie RECORD und DOPEN unterstützt. Die Befehle LOAD, SAVE und VERIFY beziehen sich ohne Parameteranga-

be immer auf die Floppy, wobei man in einem mit CATALOG gelisteten Inhaltsverzeichnis vor das zu ladende Programm nur noch LOAD schreiben muß und kein Doppelpunkt mehr hinter den Namen notwendig wird. Auch wird jetzt mit der RUN-Taste das erste Programm von Diskette geladen und gestartet. Man kann aber auch jedes beliebige Programm von Diskette laden und starten, indem man RUN und dahinter den Namen des Programmes eintippt. Sehr interessant ist der Befehl MERGE, der nicht nur ein weiteres Programm an das im Speicher vorhandene anhängt, sondern zeilenweise einsortiert.

Auch die Anhänger der Kassettenspeicherung wurden nicht vergessen. Die Aufzeichnungsgeschwindigkeit auf Band wurde um ein Vielfaches erhöht. Trotzdem können Programme im normalen Aufzeichnungsformat geladen werden.

Zur übersichtlichen Programmierung wurden sowohl bestehende Befehle erweitert, als auch neue definiert. Das altbekannte IF THEN wurde um die Möglichkeit der Verwendung eines ELSE-Falles bereichert. Auch kann man den Befehl RESTORE in Zusammenhang mit Zeilennummern benutzen. Ferner liefert die Abfrage ASC(CHR\$(0)) keinen Fehler mehr, sondern den Wert Null.

Neu hinzugekommen sind Schleifenbefehle wie DO-LOOP-WHILE,

Fortsetzung auf Seite 132

Fortsetzung von Seite 121

mit denen allgemein Schleifen wesentlich flexibler und übersichtlicher aufgebaut werden können. Dabei können 64 solcher Schleifen ineinander verschachtelt werden.

Weiterhin können Fehler im Programm mit dem Befehl TRAP abgefangen und mit den dazu vorhandenen Variablen EL, ER und ERR\$ lokalisiert werden. Anschließend kann mit RESUME zu der Programmstelle zurückgesprungen werden, an der der Fehler auftrat.

Einige Kommandos zur Stringverarbeitung wurden ebenfalls eingebaut, wie etwa INSTR zur Stringsuche. Interessant ist hier die Erweiterung des Befehls MID\$, der jetzt bei Zuweisungen auch links vom Gleichheitszeichen stehen kann und somit ein kontrolliertes Einsetzen von Zeichenketten in andere ermöglicht.

Die gesamte Speicherverwaltung wurde, wie am Anfang erwähnt, umgekrempelt. Deshalb wurden Befehle wie POKE und PEEK mit geändert und greifen jetzt nur noch auf den 64-KByte-RAM-Speicher zu. Um trotzdem alle Ein-/Ausgabebausteine zu erreichen, wurden einige fest installierte Variablenfelder eingerichtet. Man kann zum Beispiel mit den Variablen VIC(x) auf alle Register des Grafikbausteins direkt zugreifen, ohne einen einzigen POKE-Befehl verwenden zu müssen. Ähnliches gilt für den Soundchip und die beiden CIAs. Ferner läßt sich der Bildschirm- und Farbspeicher mit den Variablen VID(x) und COL(x) beeinflussen, wobei x von 0 bis 999

reicht. Daneben gibt es die Variablen BORDER, PAPER und INK, mit denen man die Bildschirmfarben direkt beeinflussen kann. Der Nachteil dieses Konzeptes ist es, daß Programme, die in Basic V2.0 geschrieben wurden und viele POKES verwenden, oft nicht funktionieren.

Abgerundet wird die Palette der neuen Befehle durch Kommandos wie zum Beispiel UPPER und LOWER zum Festlegen des Groß- oder Kleinschriftmodus, CLS zum Löschen des Bildschirms, RESET um den Einschaltzustand zu erreichen und MONITOR zum direkten Aufruf eines vorher eingeladenen Monitors, wenn dieser über einen BREAK-Einsprungspunkt verfügt. Auch der OLD-Befehl ist vorhanden, mit dem man ein NEW oder das oben erwähnte RESET wieder aufheben und damit ein Programm im Speicher wieder restaurieren kann. Ferner läßt sich mit dem Befehl STOP ON/OFF die Stop-Taste blockieren oder freigeben.

Das Handbuch erläutert all diese Befehle in verständlicher Art mit vielen Programmbeispielen.

Bemerkenswert ist in diesem Zusammenhang, daß Erweiterungen wie TurboAccess oder Hypra-Load (ROM-Version) weiterhin funktionieren.

Business Basic ist sowohl wegen der Fähigkeit mehr als 61 KByte direkt in Basic zur Verfügung zu stellen, als auch wegen seiner Qualitäten als Basic-Erweiterung sehr interessant. Nur einige Befehle zur Steuerung der hochauflösenden Grafik, die ja durchaus im Sinne dieser Erweiterung liegen würden, fehlen etwas. Der Name dieser Erweiterung ist trotzdem durchaus berechtigt, denn Business Basic erlaubt professionelles Programmieren zum Heimanwender-Preis (198 Mark).

(K. Hinsch/A. Wängler/rg)

198.-



**Wir wollen nicht die Größten sein!
NUR DIE BESTEN!
KINGSOFT**

Fritz Schäfer

Info anfordern: Schnackebusch 4 · 5106 Roetgen · Telefon 02408/51 19