

Gelegentlich erwähne ich bei meinen Erklärungen, daß wichtige Dinge der Commodore-Computer nicht in den Handbüchern stehen.

Kürzlich erhielt ich zu diesem Thema den Brief eines Lesers, der mich darauf hinwies, daß entgegen meiner Behauptung die meisten der von mir vermittelten Erklärungen in einem Handbuch, nämlich im »64 Intern« von Data Becker zu finden sind. Natürlich hat dieser Leser recht: In diesem Buch steht in der Tat sehr viel Informatives. Ich benutze es selbst oft und kann es den fortgeschrittenen Computer-Amateuren sehr empfehlen.

Nur, lieber Leser, ich halte es nicht für ein Handbuch, sondern für Fachliteratur. Wenn ich Handbuch sage, dann meine ich die von Commodore offiziell herausgegebenen Schriften — für Anfänger und Amateure. Und diese Gattung könnte wirklich etwas ergiebiger sein.

Adresse 183 (\$B7)

Länge des derzeitigen Filenames

Die LOAD-, SAVE- und VERIFY-Befehle für Disketten verlangen die Angabe eines Programm- oder Dateinamens, auf Computerdeutsch »File-Name«. Nähere Angaben dazu finden Sie im Texteingabebereich # 1 »Files-Geräte-Namen-Nummern«.

Auch der OPEN-Befehl kann einen File-Name haben. Bei Kasettenoperationen kann der File-Name weggelassen werden.

In der Speicherzelle 183 steht während und nach der Verwendung eines der oben genannten Befehle eine Zahl, die angibt, aus wievielen Zeichen der File-Name besteht.

Bei Disketten sind File-Namen möglich, die aus maximal 16 Zeichen bestehen.

Bei Kassetten dagegen sind Namenslängen von maximal 187 Zeichen erlaubt. Allerdings werden vom Computer auf dem Bildschirm nur 16 Zeichen ausgedruckt (siehe dazu den Texteingabebereich »Tape-Headers«).

Für die Längenangabe in Zelle 183 gilt dabei nur die Anzahl derjenigen Zeichen, die zwischen den Gänsefüßchen stehen.

Diese Zahl kann nach einer Ein-/Ausgabeoperation, auch nach einer ungültigen oder abgebrochenen, durch PEEK (183) ausgelesen werden.

Ein File-Name wird übrigens auch bei einem OPEN-Befehl der RS232-Schnittstelle angegeben. Dieser Name, der bis zu vier Zeichen lang sein kann, wird in die Speicherzellen 659 bis 662 übertragen und gibt dort die Übertragungsrates, Wortlänge und Parity-Prüfung an.

Memory Map mit Wandervorschlägen (11)

Bei unserer Wanderung durch die Speicherlandschaft treffen wir heute auf die Speicherstellen 183 bis 199. Sie sind unter anderem auch für die Filenummer, Geräteadressen und Sekundär-Adresse verantwortlich.

Adresse 184 (\$B8)

Nummer der derzeitigen Datei (File)

Hinter jedem OPEN-Befehl steht eine Zahl, die der durch diesen Befehl angeforderten Datei zugeordnet wird. Diese Datei- oder File-Nummer gilt als Referenz für alle anderen Ein- und Ausgabebefehle derselben Datei. Nähere Angaben dazu können Sie dem nebenstehenden Texteingabebereich Nr. 1 »Files-Geräte-Namen-Nummern« entnehmen.

Ein OPEN-Befehl ruft die entsprechende Routine des Betriebssystems auf, welche die File-Nummer in die Speicherzelle 184 schreibt. Vor dort kann sie mit PEEK(184) ausgelesen werden. Geben Sie die folgende Zeile direkt ein:

A=30:OPEN A,3:PRINT PEEK(184):CLOSE A

Um verschiedene File-Nummern auszuprobieren, definieren wir sie als Variable A. Nach dem »A« des OPEN-Befehls steht die Zahl 3. Damit wird der Bildschirm angewählt (siehe »Sekundär-Adresse« im schon erwähnten Texteingabebereich). Das Anwählen des Bildschirms vermeidet eine störende Meldung des Betriebssystems.

Mit RETURN nach der obenstehenden Zeile wird der jeweilige Wert von A als Inhalt der Zelle 184 ausgedruckt.

Adresse 185 (\$B9)

Derzeitige Sekundär-Adresse

Die Sekundär-Adresse steht als dritte Angabe hinter den Ein- und Ausgabe-Befehlen LOAD, SAVE, VERIFY und OPEN. Sie hat bei den verschiedenen Peripheriegeräten spezielle Funktionen. Diese Funktionen sind im nebenstehenden Texteingabebereich näher erläutert.

Der jeweilige Wert der Sekundär-Adresse steht in der Speicherzelle 185, allerdings um 96 erhöht. Für Sekundär-Adressen stehen, über die Standardwerte der einzelnen Peripheriegeräte hinaus, die Zahlen von 0 bis 31 zur Verfügung. Ab 32 fängt in Zelle 185 wieder der Zyklus ab 0 an. Das können wir uns anschauen. Ich wähle zur Eröffnung einer Datei wieder den Bildschirm als »nicht-störendes« Empfangsgerät.

A=15:OPEN 1,3,A:PRINT PEEK(185)-96:CLOSE 1

Durch Verändern des Wertes von A können Sie alle Möglichkeiten durchspielen.

Adresse 186 (\$BA)

Derzeitige Geräte-Nummer

Jedes an den Computer anschließbare Gerät hat eine eigene Nummer, die zusammen mit den Ein-/Ausgabe-Befehlen LOAD, SAVE, VERIFY und OPEN angegeben werden muß. Wird keine Nummer angegeben, nimmt der Computer automatisch an, daß die Datensette gemeint ist.

Alle von Commodore vorgegebenen Geräte-Nummern sind in der folgenden Tabelle aufgelistet.

Geräte-Nummer	angesprochenes Gerät
0	Tastatur
1	Datensette
2	RS232- (User-Port) Schnittstelle
3	Bildschirm
4	Drucker (normal)
5	Drucker (zusätzlich)
8	Disketten-Laufwerk Nr. 0
9	Disketten-Laufwerk Nr. 1
10, 11	weitere Disketten-Laufwerke

Die normale Geräte-Nummer eines Druckers ist 4, die eines Disketten-Laufwerks 8. Die zusätzlichen Nummern müssen gesondert am betreffenden Gerät eingestellt werden.

Nach der Ausführung eines der oben genannten Befehle steht die entsprechende Geräte-Nummer in der Speicherzelle 186, aus der sie mit PEEK(186) ausgelesen werden kann.

Adresse 187 bis 188 (\$BB bis \$BC)

Zeiger auf Adresse des derzeitigen Filenames

Die Bedeutung eines Programm- oder Dateinamens — normalerweise kurz »File-Name« genannt, sind im nebenstehenden Texteingabebereich »File-Geräte-Namen-Nummern« näher beschrieben. In den Speicherzellen 187/188 steht in der Low/High-Byte-Darstellung ein Zeiger auf diejenige Adresse im Programm-Speicher, wo dieser Name gespeichert ist.

Eine Ausnahme ist hier der OPEN-Befehl der RS232-Schnittstelle. Ihr File-Name wird in die Speicherzellen 659 bis 662 gebracht, wo er verschiedene Parameter dieser Schnittstelle steuert.

Adresse 189 (\$BD)

Zwischenspeicher für RS232-Parity-Prüfung und für Kassettenoperationen

Die RS232-Routinen benutzen diese Speicherzellen als Zwischenspeicher für ein Prüf-Byte (Parity-Prüfung) bei der Ausgabe. Die Parity-Prüfung habe ich kurz im letzten Heft im neunten Teil des Kurses erklärt.

Auch die Kassetten-Routinen bedienen sich dieser Speicherzelle. Sie verwenden sie als Zwischenspeicher für das gerade gesendete oder empfangene Zeichen.

Adresse 190 (\$BE)

Blockzähler für Kassetten-Ein-/Ausgabe

Das Betriebssystem des Computers schreibt bei SAVE ein Programm zweimal auf das Band der Datensette. Beim LOAD-Befehl wird der erste Block in den Arbeitsspeicher des Computers geladen; der zweite — identische — Block wird dann mit dem ersten Block Byte für Byte verglichen, um Datenfehler auf dem nicht immer ganz zuverlässigen Bandmaterial zu erkennen.

In der Speicherzelle 190 wird dem Betriebssystem angezeigt, wieviele Blockteile bei diesem Prozeß noch gelesen oder gespeichert werden müssen. Vom Basic-Programm aus ist diese Speicherzelle nicht zugänglich.

Adresse 191 (\$BF)

Zwischenspeicher für LOAD-Operationen vom Band

Diese Speicherzelle wird beim Laden eines Programms vom Band dazu benutzt, um Zeichen aus einzelnen Bits zusammenzusetzen.

Adresse 192 (\$C0)

Motorsperre der Datasette

Die Tasten der Datasette werden sechszigmal in der Sekunde von der »Interrupt-Routine« des Betriebssystems überprüft, ob eine von ihnen gedrückt worden ist. Die Speicherzelle 192 spielt dabei eine entscheidende Rolle, beim C 64 allerdings in einer anderen Weise als beim VC 20. Wie sie diese Rolle beim C 64 spielt, ist im Zusammenhang mit der Speicherstelle 1 ganz am Anfang dieses Kurses beschrieben worden (siehe 64'er-Ausgabe, 11/84, Seite 173). Ich habe dabei in zwei Beispielen gezeigt, wie durch Abfrage des vierten Bits von Adresse 1 geprüft werden kann, ob eine Taste der Datasette gedrückt ist und wie der Motor durch Setzen und Löschen des Bit 5 der Zelle 1 ein- und ausgeschaltet werden kann. Vorausgesetzt, der Inhalt der Speicherzelle 192 ist ungleich Null und eine Taste der Datasette ist gedrückt.

Heute will ich, wie damals versprochen, denselben Vorgang für den VC 20 beschreiben.

Wie Sie sich vielleicht noch erinnern, wird die Speicherzelle 1 beim VC 20 nicht für die Steuerung der Ein- und Ausgänge des Mikroprozessors verwendet. Diese Rolle wird beim VC 20 durch zwei Register des »Versatile Interface Adapter« (VIA 6522-A) ausgefüllt.

Für die Abfrage der Datasetten-Tasten ist das sechste Bit des VIA-Registers 37151 zuständig. Bei gedrückter Taste steht es auf 1, sonst auf 0. Ein kleines Programm zeigt es Ihnen:

```
10 X = PEEK(37151)
20 PRINT X
30 IF X = 62 THEN 50
40 GOTO 10
50 PRINT "TASTE
GEDRÜCKT"
```

Wenn keine Taste gedrückt ist, läuft ein Zahlenband mit 126 ab. Die entsprechende Darstellung als Dualzahl lautet 1111 1110. Bei einer gedrückten Taste steht in 37151 die Zahl 62, als Dualzahl 0011 1110. Wichtig, wie gesagt ist nur das zweithöchste Bit.

Mit der Abfrage der Zeile 30 springt beim Drücken einer Taste das Programm auf die Zeile 50 und druckt den Text aus.

Den Motor der Datasette können wir mit Hilfe des Registers 37184 schalten. Wie beim C 64

gilt auch jetzt, daß dazu die hier angesprochene Speicherzelle 192, auch Interlock-Register genannt, eine Zahl größer als 0 enthält und daß außerdem eine Taste der Datasette gedrückt ist. Drücken Sie auf PLAY und geben Sie direkt ein:

POKE 192,255
POKE 37148,251: der Motor bleibt stehen.
POKE 37148,252: der Motor läuft los.

Bestimmend sind hier Bit 2, 3 und 4.

Zum Ausschalten muß lediglich Bit 2 auf 1 stehen, zum Einschalten die drei Bits auf 110. Jede Zahl, die als Dualzahl diese Bedingungen erfüllt, kann dafür hergenommen werden. Um unabhängig von den anderen Bits des Registers 37148 zu bleiben, die ja auch ganz bestimmte andere Funktionen haben, empfiehlt es sich, über Boole'sche Verknüpfungen nur die wichtigen drei Bits zu verändern. Die beiden POKE-Befehle sehen dann so aus:
Ausschalten: POKE 37148,PEEK (37148) OR 2
Einschalten: POKE 37148,PEEK (37148) AND 12

Adresse 193 bis 194 (\$C1 bis \$C2)

Anfangsadresse für Ein-/Ausgabe-Operationen

In diesen Speicherzellen steht in Low/High-Byte-Darstellung die Adresse, ab der ein Programm gerade geladen oder gespeichert wird. Dieses Adresse wird übrigens von hier auch in die Speicherzellen 172/173 gebracht, die wir schon früher besprochen haben.

Bei LOAD und SAVE auf Band steht hier die Anfangsadresse des Bandpuffers (828). Im Bandpuffer steht allerdings nur der sogenannte Bandvorspann (auf englisch »Tape Header«), während der Hauptteil des Programms im Programmspeicher ab einer Adresse steht, auf die der Zeiger in den Speicherzellen 195/196 hinweist.

Adresse 195 bis 196 (\$C3 bis \$C4)

Zeiger auf den Anfang des Programms hinter dem Tape Header

Bei jedem LOAD- und SAVE-Befehl für Kassetten wird der Vorspann (Tape Header), in dem Programmtyp, Anfangs- und Endadresse aufgezeichnet sind, im Kassettenpuffer ab Adresse 828 gespeichert. Der eigentliche Teil des Programms steht dann im Programmspeicher.

In den Speicherzellen 195/196 steht in der Low-High-Byte-Darstellung diese Adresse, ab

der das Programm beginnt. Ich habe für alle diejenigen, die mit der Datasette arbeiten, im nebenstehenden Textzeitschub # 2 »Tape-Header« die Zusammenhänge mit einem Beispiel dargestellt.

Adresse 197 (\$C5)

Tasten-Code der zuletzt gedrückten Taste

In Ausgabe 6/85 des 64'er auf Seite 123 habe ich Ihnen mit Wort und Bild beschrieben, wie die Tasten des Computers abgefragt werden. Die dabei für jede Taste entstehende Dualzahl wird in eine Dezimalzahl (0 bis 63) umgewandelt und zuerst in die Speicherzellen 203 beziehungsweise 653 gebracht. Zur Umwandlung und Abfrage der Zellen 203 und 653 bringe ich das nächste Mal mehr Details. Nach der Prüfung, welche Taste gedrückt worden ist, wird die Codezahl von 203 in die Speicherzelle 197 gebracht und dort »aufgehoben«. Diese vermeintliche Verdoppelung wird vom Betriebssystem dafür gebraucht, um zu erkennen, ob die nächste gedrückte Taste mit der vorhergehenden identisch ist. Ist sie identisch, dann entscheidet der Inhalt der Speicherzelle 650, ob das Zeichen dieser Taste mehrfach ausgedruckt wird. In 650 steht die sogenannte Wiederholungsfunktion. Aber ich will nicht vorgreifen. Die Codezahlen der einzelnen Tasten werde ich bei der Besprechung der Zelle 203 auflisten.

Adresse 198 (\$C6)

Anzahl der Zeichen im Tastaturpuffer

Die Funktion des Tastaturpuffers, zu dem wir bei den Speicherzellen 631 und 640 noch kommen werden, habe ich bereits in diesem Kurs und zwar in Ausgabe 7/85 auf Seite 141/142 im Textzeitschub »Dynamische Tastenabfrage« erklärt. Dabei habe ich damals schon sozusagen im Vorgriff die Zelle 198 verwendet.

In dieser Speicherzelle steht die jeweilige Anzahl der Zeichen, die im Tastaturpuffer gespeichert sind und darauf warten, weiterverarbeitet zu werden.

Das folgende kleine Programm zeigt es.

```
10 GET A$
20 PRINT PEEK (198);A$
30 FOR J=1 3000:NEXT J
40 GOTO 10
```

Der GET-Befehl holt ein Zeichen aus dem Tastaturpuffer — sofern eines dort zu finden ist. Die Zeile 20 druckt die Anzahl der Zeichen im Puffer aus, daneben das erste dieser Zeichen. Dann folgt eine Warteschleife, die uns erlaubt, ganz schnell ein paar Tasten zu drücken. Danach

springt das Programm an den Anfang zurück und arbeitet diese eingegebenen Zeichen ab. Es ist dabei deutlich zu sehen, wie durch den GET-Befehl bereits ein Zeichen aus dem Puffer genommen und dadurch der Inhalt der Zelle 198 sofort um 1 reduziert wird.

Der Inhalt der Speicherzelle 198 kann mit POKE auch verändert werden.

Eine sinnvolle Anwendung dieser Beeinflussung erlaubt der nicht gerade sehr populäre WAIT-Befehl.

Ersetzen Sie bitte im obigen Programm die Warteschleife der Zeile 30 durch:

30 POKE 198,0: WAIT 198,1

Zuerst wird dem Computer vorgegaukelt, daß der Tastaturpuffer leer sei. Durch den WAIT-Befehl wartet das Programm danach so lange, bis ein Zeichen im Tastaturpuffer erscheint und springt erst dann auf die nächste Zeile 40.

Wenn Sie nach dem WAIT-Befehl statt der 1 eine 2 eingeben, wartet diese Zeile entsprechend auf zwei Tasteneingaben. Allerdings wird in der Zeile 20 dann nur jedes zweite Zeichen ausgedruckt.

Adresse 199 (\$C7)

Flagge für inverse Darstellung der Zeichen

Mit dieser nützlichen Adresse fahren wir das nächste Mal fort. (Dr. H. Hauck/ah)

Textzeitschub # 1

Files — Geräte — Namen — Nummern

In den Handbüchern von Commodore und auch in anderen Beschreibungen wird von den Ein- und Ausgabe-Befehlen, wie zum Beispiel LOAD, SAVE, OPEN etc. leider ein recht verwirrendes Bild geboten. Ich beziehe mich dabei auf die hinter diesen Befehlen stehenden Ziffern und Namen.

In der Bezeichnung »File-Name« und »Geräte-Nummer« sind sich die Autoren noch weitgehend einig. Daß die dritte Angabe hinter den Befehlen aber Sekundär-Adresse, Command, Speicheradressen-Flag oder gar EOT-Flag genannt wird, muß den Computernutzer zwangsläufig verwirren.

Da wir im nebenstehenden Kurs gerade die Speicherzellen 183 bis 188 besprechen, die alle mit diesen Anhängseln der eingangs genannten Befehle zusammenhängen, ist dies eine gute

Fortsetzung auf Seite 139

Fortsetzung von Seite 134

Gelegenheit, etwas Systematik in die Angelegenheit zu bringen.

Als erstes gebe ich die Bezeichnungen der Befehle aus einem Commodore-Buch wieder, bevor die einzelnen Angaben im Detail diskutiert werden.

— LOAD "File-Name", Geräte-Nr., Speicheradressen-Flag
— SAVE "File-Name", Geräte-Nr., EOT-Flag

— VERIFY "File-Name", Geräte-Nr., Speicheradressen-Flag

— OPEN File-Nr., Geräte-Nr., Sekundär-Adresse, "File-Name, Typ, Modus"

— INPUT # File-Nr., Variable

— GET # File-Nr., Variable

— CMD File-Nr.

— CLOSE File-Nr.

File-Name

„File“ wird normalerweise mit „Datei“ übersetzt.

Einen Datei-Namen gibt es aber nur beim OPEN-Befehl. Bei den Befehlen LOAD, SAVE und VERIFY ist der Name des Programms gemeint, der bekanntlich in Gänsefüßchen hinter diesen Befehlen steht. Bei Disketten als Pflicht, bei Kassetten als Option. Beim OPEN-Befehl steht der Name zwar auch in Gänsefüßchen, aber nicht direkt hinter dem Befehl, sondern erst an vierter Stelle. Auch er kann bei Kassetten-Betrieb, aber auch beim Drucker weggelassen werden.

Bei Disketten-Dateien steht hinter dem Namen — als sein Bestandteil — noch Typ und Modus. Typ bezeichnet die Art der Datei (RELative, SEQuential, USeR, PRoGramm), Modus die Operation (Read, Write, Append).

Der File-Name kann bei Disketten auch eine Anweisung an das Betriebssystem des Disketten-Laufwerks sein (NEW, SCRATCH, RENAME etc.) Nähere Erklärungen dazu finden Sie im Handbuch des Disketten-Laufwerks.

Da in 187/188 die Adresse angegeben ist, ab der der Programm- beziehungsweise Datei-Name im Speicher steht, können wir ihn dort ansehen.

Geben Sie direkt ein LOAD "1234"

Nach der RETURN-Taste kommt die übliche Aufforderung der Datasette, die wir aber mit der STOP-Taste abwürgen. In 187/188 steht jetzt die Anfangsadresse »A« des gespeicherten Namens:

A = PEEK(187) + 256 * PEEK(188):PRINT PEEK(A)

Wir erhalten die Zahl 49, das ist der ASCII-Codewert der Ziffer »1«. Durch Erhöhen der Anfangsadresse »A« im PRINT-Befehl um jeweils 1, kommen auch die übrigen Zeichen des Namens zum Vorschein.

Die VC 20-Besitzer kennen sicher den Effekt, der entsteht, wenn beim SAVE-Befehl vor dem File-Namen, aber innerhalb der Gänsefüßchen, eine Farbtaste eingegeben wird. Die FOUND-Meldung druckt nämlich den File-Namen in der gewählten Farbe aus, was besonders bei Kassetten-Operationen sehr eindrucksvoll ist.

Das Disketten-Laufwerk bietet eine andere Überraschung. Die geSHIFtete SPACE-Taste innerhalb des File-Namens, erzeugt das zweite Gänsefüßchen. SAVE "SPIEL 2 (SHIFT-SPACE) SYS 16000", 8 erscheint in der Directory als: "SPIEL 2" SYS 16000 und kann mit LOAD "SPIEL 2", 8 geladen werden.

Geräte-Nummer

Alle an den Computer anschließbaren Peripheriegeräte haben eine eigene Nummer, unter welcher sie »angesprochen« werden können. Es handelt sich eigentlich um eine Adresse, und in der Tat sprechen manche Autoren von der »Primär-Adresse«. In der Beschreibung der Speicherzelle 184 sind sie alle zusammengestellt.

Bei den Befehlen LOAD, SAVE, VERIFY und OPEN steht diese Geräte-Nummer als zweite Angabe, vor der ersten Angabe durch ein Komma getrennt: LOAD "NAME", 1.....bedeutet "Laden von Datasette" LOAD "NAME", 8.....bedeutet "Laden von Diskette" OPEN 12,4.....bedeutet "An den Drucker"

(Die 12 hinter dem OPEN-Befehl ist eine »File-Nummer«. Sie kommt weiter unten an die Reihe.)

Die Gerätenummer kann auch weggelassen werden. Dann allerdings nimmt der Computer automatisch an, daß es sich um die Datasette handelt. Die Geräte-Nummer steht jeweils in der Speicherzelle 186 und kann von dort mit PEEK ausgelesen werden.

Sekundär-Adresse

Die dritte Angabe hinter LOAD, SAVE und OPEN hat widersprüchliche Namen,

wohl deshalb, weil sie abhängig von der Gerätenummer verschiedene Aufgaben hat. Von den ganz am Anfang schon genannten Varianten gefällt mir »Sekundär-Adresse« am besten.

Die folgende Tabelle faßt die Funktionen der Sekundär-Adresse zusammen.

Sie sehen, die Sekundär-Adressen haben es in sich! Die gerade benutzte Sekundär-Adresse steht in der Speicherzelle 185 und kann mit PEEK von dort ausgelesen werden.

File-Nummer

Die File-Nummer, oft auch logische File-Nummer genannt, steht als erste Angabe hinter dem OPEN-Befehl und den damit verbundenen PRINT #-, INPUT #-, GET #- und CLOSE-Befehlen.

Sie gibt einer zu bearbeitenden Datei eine Nummer, die von den nachfolgenden, anderen Befehlen ebenfalls verwendet werden muß, wenn sie sich auf dieselbe Datei beziehen. Auf diese Weise können mehrere Dateien nebeneinander bearbeitet werden, und zwar fünf bei Disketten und auf alle Geräte verteilt insgesamt zehn.

File-Nummern dürfen die Werte von 1 bis 255 haben.

Bei Werten über 127 wird bei einem PRINT #-Befehl nach jedem RETURN-Zeichen — CHR\$(13) — zusätzlich ein ASCII-Code für Zeilenvorschub — CHR\$(10) — an das Gerät gegeben. Diese Eigenschaft kann bei denjenigen Geräten nützlich sein, die normalerweise auf CHR\$(13) ohne Zeilenvorschub reagieren (kein automatisches Line-Feed).

Die File-Nummer steht jeweils in der Speicherzelle 184, von wo sie mit PEEK ausgelesen werden kann. Zum Beispiel, um nachzuprüfen, welches Gerät als letztes angesprochen worden ist.

Abschließend möchte ich nochmals darauf hinweisen, daß in den Speicherzellen 183 bis 188 immer die gerade als letzte verwendete Angabe steht. Wir haben aber gesehen, daß der Computer sich maximal zehn File-Nummern mit dazugehörigen Geräte-Nummern, Sekundär-Adressen etc. merken kann.

Er tut dies in speziellen Tabellen, die in den Speicherzellen 601 bis 630 stehen.

Wir werden uns also noch einmal mit dieser Sache befassen, wenn wir bei dem entsprechenden Adressen angekommen sind.

Befehl	Sekundär Adresse	Funktion
LOAD	0	lädt Programm an den Anfang des Programmspeichers
	1	lädt ein Programm absolut, also an die Adresse, von der ab es gespeichert wurde.
SAVE	0	Normales SAVE, Programm wird bei späterem LOAD an den Anfang des Programmspeichers geladen (Basic-Programme)
	1	erzwingt bei späterem LOAD des Programms die Speicherung ab der Adresse, wo es zur Zeit steht (Maschinen-Programme)
	2	setzt am Programmende auf der Kassette eine »Band-Ende«-Markierung, die beim »Überlesen« das Band mit der Fehlermeldung »DEVICE NOT PRESENT« stoppt.
	3	Kombination von 1 und 2
OPEN bei Kassette	0	Daten lesen
	1	Daten schreiben
	2	Daten schreiben mit »Bandende«-Markierung
OPEN bei Diskette	0	vom Betriebssystem der Floppy für »Laden« reserviert
	1	vom Betriebssystem der Floppy für »Speichern« reserviert
	2-14	reserviert nummerierten Daten-Kanal, bis zu drei gleichzeitig
	15	reserviert Kommando-Kanal (nähere Angaben zu diesen Sekundär-Adressen siehe Floppy-Handbuch)
OPEN bei Drucker	0-10	die Funktionen sind bei den Druckern zum Teil verschieden. Bitte in Drucker-Anleitung nachsehen.

Texteinschub # 2

Tape Header

Wenn ein Programm oder eine Datei auf Band gespeichert wird, setzt der Computer vor das Programm einen Vorspann, der auf englisch »Tape-Header« genannt wird. Da dieser Name weit verbreitet ist, will ich ihn hier beibehalten. Der Tape-Header ist 192 Byte lang. Er enthält alle wichtigen Angaben über das nachfolgende Programm.

Beim Laden eines Programms wird der Tape Header im Kassettenspeicher gespeichert, für den die Speicherstellen 828 bis 1019 reserviert sind. Von dort kann der Inhalt des Tape-Headers gelesen und analysiert werden.

Bevor wir das versuchen, will ich erst seine Zusammensetzung erklären.

Im ersten Byte steht eine Kennzahl für den Typ des Programms. Diese Kennzahl ist abhängig von der Sekundär-Adresse, die beim SAVEN eingegeben worden ist. Die Arten der Sekundär-Adressen und ihre Bedeutung ist im anderen Texteinschub »Files-Geräte-Namen-Nummern« genau beschrieben. Es gibt zwei Kennzahlen: 1 und 3.

In Anlehnung an die Erklärung der Sekundär-Adresse kann man die Kennzahl generell dadurch beschreiben, daß ein Programm mit Kennzahl 1 immer an den Anfang des zur Verfügung stehenden Programmspeichers geladen wird. Hauptsächlich kommt das für Basic-Programme in Frage.

Eine Kennzahl 3 bewirkt, daß das Programm an diejenige Stelle des Programmspeichers geladen wird, wo es vor dem SAVEN gestanden hat. Das ist hauptsächlich der Fall bei Programmen in Maschinensprache.

In Verbindung mit der Bedeutung der Sekundär-Adresse kann man den Zusammenhang wie Tabelle 1 zeigt darstellen:

In Byte 2 und 3 steht in Low/High-Darstellung die Adresse, ab der das Programm im Speicher des Computers stand, als es gespeichert wurde.

In Byte 4 und 5 steht die entsprechende End-Adresse des Programms.

Ab Byte 6 bis Byte 192 steht der Name des Programms. Er darf also maximal 187 Zei-

Sekundär Adresse	Kennzahl	Bedeutung
0 oder leer	1	Basic-Programm
1	3	Maschinen-Programm
2	1	Basic-Programm mit End-Of-Tape-Marke
3	3	Maschinen-Programm mit End-Of-Tape-Marke

Tabelle 1

chen lang sein. Bei LOAD werden allerdings nur 16 Zeichen auf dem Bildschirm dargestellt.

Jetzt wollen wir das alles mit einem kleinen Experiment überprüfen.

Schreiben Sie bitte ein kleines Programm, es braucht nicht sehr sinnvoll zu sein, wie zum Beispiel:

```
10 REM TAPE HEADER
20 REM TEST PROGRAMM
```

Nehmen Sie ein leeres Band und laden das Programm mit einem Namen, der länger sein soll als 16 Zeichen, zum Beispiel:

```
SAVE "TEST PROGRAMM
FUER INHALT TAPE HEAD-
ER"
```

Nach Drücken der RECORD- und PLAY-Tasten der Datensette meldet der Computer:

```
FOUND TEST PROGRAMM
FU
```

Es werden also nur 16 Zeichen inklusive Leerzeichen gedruckt. Sobald das Programm geladen ist, schauen wir im Kassettenspeicher nach, was in den ersten fünf Bytes steht, danach lesen wir die restlichen Bytes des Puffers.

Geben Sie direkt, ohne Zeilennummer, ein:

```
FOR I=0 TO 4: PRINT PEEK
(828+I);: NEXT
```

Sie erhalten die Zahlen 1 1 8 41 8 (beim VC 20 mit 3-K-Speichererweiterung 1 1 4 41 4)

Danach geben wir wiederum direkt ein:

```
FOR I=5 TO 192: PRINT
CHR$(PEEK(828+I));: NEXT
```

Beim VC 20 geben Sie in der FOR...NEXT-Schleife eine kleine Zahl ein, da der Bildschirmspeicher beim VC 20 kleiner ist als beim C 64.

Jetzt erscheint der volle Programmname, gefolgt von nicht sichtbaren Leerstellen. Wenn Sie in der letzten Direkt eingabe den CHR\$-Teil weglassen, dann druckt die Zeile die ASCII-Codes aus, und Sie sehen dann die Leerstellen als Zahl 32.

Diese Resultate habe ich zur besseren Übersicht in Tabelle 2 dargestellt.

Die Kennzahl in Byte »1« können Sie dadurch verändern, daß Sie dem oben verwendeten SAVE-Direktbefehl nach dem langen Namen ein »1,1« anhängen. Im Ausdruck steht dann die Kennzahl »3«.

Übrigens, wenn Sie in den Speicherzellen 195/196 nachschauen, finden Sie dort denselben Wert wie in den Zellen 829/830, so wie die Beschreibung es in der Memory Map erklärt.

Vielleicht fragen Sie jetzt nach dem Nutzen dieser ausführlichen Erklärung. Nun, hauptsächlich kann man damit Programme, die eigentlich wegen LOAD ERROR nicht mehr ladbar sind, retten. Oder aber man kann durch Verändern der Zahlen in den Bytes 2 bis 5 nachträglich die Adressen ändern, in die das Programm geladen wird. Die erste Anwendung werde ich erklären, sobald wir zu den Adressen des Kassettenspeichers selbst kommen.

Das Problem des LOAD oder SAVE mit geänderten Adressen ist aber zu umfangreich für einen Texteinschub innerhalb dieses Kurses. Es wäre eigentlich einen eigenen kleinen Beitrag wert.

Adresse	828	829	830	831	832	833 etc.
Byte Nr.	1	2	3	4	5	6 etc.
Bedeutung	Kennzahl	Low High Byte		Low High Byte		Namen in ASCII-Code
Resultat	1	1	8	41	8	T etc.
bei C 64	1	(2089)				(2049)
Resultat	1	1	4	41	4	T etc.
bei VC 20		(1065)				(1025)

Tabelle 2

Fortsetzung von Seite 45

möchten, finden den Anschluß BASIC an Pin 20 vom Sockel des Basic-ROMs beziehungsweise der Adapterplatine, und Basic-ROM an Pin 20 des ROMs selber, wo die Leitungen dann anzuschließen sind. Am einfachsten ist dies wohl durch Herausbiegen des Pins am Basic-ROM (Achtung: nicht abbrechen) zu bewerkstelligen, so daß dieser nicht im Sockel steckt und ein Kabel angelötet werden kann.

Die Zusatzplatine ist so ausgelegt, daß man sie mit an den Schrauben, die auch zur Befestigung der Tastatur dienen, installieren kann. Ob man sie in der linken oder rechten Seite anbringt, ist völlig belanglos, nur sollte man die Taster auf der Gegenseite befestigen, beziehungsweise vorher prüfen, ob die Einbautiefe der Taster gering genug ist, so daß diese sich nicht mit der Zusatzplatine berühren.

Die Taster werden an dem oberen Gehäuseteil über der Tastatur montiert, wobei man darauf achten sollte, daß genügend Platz für die ausgewählten Taster zur Verfügung steht (Durchmesser; am besten vom Inneren des Gehäuses anpassen).

Ist die Adapterplatine so aufgesteckt, daß die Kerben der ICs in Richtung Kassettensport zeigen und mit EPROM-Trans, dem Kern und mindestens einem EPROM 1 bis 3 bestückt, sind wir fertig und können den Computer wieder zusammenbauen.

Eventuell kann die Abschirmung — dies ist die Pappe mit der Alu-Schicht — so aufgeschnitten werden, daß beim Verlegen der Kabel von Adapter zur Zusatzplatine keine Probleme auftreten.

Die Steckverbindung Tastatur — Computer kann nur in einer Richtung aufgesteckt werden; bei der Leuchtdiode zeigt das rote Kabel zum Ein-Aus-Schalter.

Gehäuse zusammenschrauben, fertig.

Ich betreibe diese Schaltung seit etwa 6 Monaten und möchte sie nicht mehr missen. Was ich damit aber zum Ausdruck bringen will, ist, daß die Stromversorgung den zusätzlichen Belastungen gewachsen ist; auch sind bis dato noch keine Wärme-probleme aufgetreten.

Mir ist bekannt, daß sich viele C64-Besitzer schwer dazu durchringen können, ihren Computer äußerlich oder innerlich (Achtung! Garantieverlust) zu verändern. Diesmal ist es aber, glaube ich, eine Überlegung wert. Alle, die nicht die Möglichkeit zum Herstellen der Platinen haben, können sich an den Verlag wenden. Lesen Sie dazu unseren Hardware-Leser-service. (Peter Rausche/aw)