

POKEs, die Sie kennen sollten

Welcher C64-Benutzer hat sich noch nicht mit dem Basic seines Computers herumgeärgert? Viele Programmierlösungen sind nur über PEEK und POKE möglich. Die folgende Liste der wichtigsten PEEKs und POKEs soll Ihnen behilflich sein.

Hex	Dezimal	Beschreibung
1	1	Inhalt 55 = normal Inhalt 54 = Basic ausgeschaltet (auf RAM umgestellt) Inhalt 53 = Basic und Kernal auf RAM umgestellt. (Es empfiehlt sich dabei, das Basic und das Kernel vorher ins RAM zu POKEn, damit der Computer bei der Umschaltung nicht aussteigt.)
A	10	Ist PEEK (10) = 0, so war der letzte Befehl LOAD. Ergibt PEEK (10) eine 1, ist VERIFY eingegeben worden.
11	17	Mit diesem PEEK läßt sich abfragen, wie die letzte Variable zugewiesen wurde. Ist PEEK (17) = 00, dann war die letzte Variablenzuweisung ein INPUT, oder es hat noch keine Zuweisung stattgefunden. Ist PEEK (17) = 64, dann wurde die letzte Variable durch GET geholt. Bei PEEK (17) = 152 erfolgte die letzte Variablenübergabe durch einen READ-Befehl.
13	19	Durch POKE 19,64 wird beim nächsten INPUT-Befehl kein Fragezeichen mehr ausgegeben. Allerdings kann man nachher durch Drücken der RETURN-Taste nicht mehr in die nächste Zeile gelangen. Es empfiehlt sich daher, nach dem INPUT-Befehl diesen Befehl wieder mit POKE 19,0 rückgängig zu machen.
18	24	Mit POKE 24,0 wird FORMULAR TOO COMPLEX ERROR aufgehoben.
2B/2C	43/44	Der Anfang des zur Zeit im Speicher befindlichen Basic-Programms errechnet sich durch PEEK (43)+PEEK (44)*256.
2D/2E	45/46	Das Ende des Basic-Programms erhält man durch PEEK (45)+PEEK (46)*256.
37/38	55/56	Mit PEEK (55)+256*PEEK (56) kann das Ende des Basic-RAMs abgerufen werden.
39/3A	57/58	Die Zeilennummer, bei der nach einer Programmunterbrechung gestoppt wurde, errechnet sich durch PEEK (57)+256*PEEK (58).

Hex	Dezimal	Beschreibung
3D/3E	61/62	Zeiger auf Basic-Statement für CONT: Durch PEEK (61)+PEEK (62)*256 erhält man die Speicherstelle, die nach dem zuletzt ausgeführten Basic-Befehl liegt, das heißt die Speicherstelle, von der sich der Basic-Interpreter bei CONT den nächsten Befehl holt. Tip: Bei CONT kommt öfter CAN'T CONTINUE ERROR vor, wenn man nach dem Stoppen ein CLR eingegeben oder in irgendeiner Programmzeile etwas geändert hat. Liest man die Werte mit PEEK (61) und PEEK (62) nach der Unterbrechung aus, dann macht ein CLR oder ähnliches nichts aus, wenn man vor CONT die zuvor ausgelesenen Werte wieder in die Speicherstellen POKEt.
3F/40	63/64	Nummer der aktuellen DATA-Zeile: Mit PEEK (63)+PEEK (64)*256 erhält man die Nummer der DATA-Zeile, aus der gerade das letzte Datum geholt wurde. (Gut zum Finden von Fehlern in DATA-Zeilen geeignet.)
45/46	69/70	Zuletzt zugewiesene Variable: Bei normalen Fließkommavariablen liest man den Wert mit PRINT CHR\$(PEEK (69))+CHR\$(PEEK (70)) aus. Bei Intervariablen (zum Beispiel XY%) erhält man den Namen durch PRINT CHR\$(PEEK (69)-128)+CHR\$(PEEK (70)-128). Strings (zum Beispiel VX\$) erhält man durch PRINT CHR\$(PEEK(69))+CHR\$(PEEK (70)-128).
78	120	Nach Ausführung dieses POKEs nimmt der C64 keinerlei Befehle mehr an: POKE 120,2
90	144	Statusbyte: Mit PRINT ST läßt sich das Statusbyte abfragen.
91	145	Nach WAIT 145,16,16 wartet der Computer auf den Feuerknopf von Joystick 1.
93	147	Wenn man die LOAD-Routine im Betriebssystem anspringt, holt es sich aus der Speicherstelle 147 die Information, ob LOAD oder VERIFY durchgeführt wird. Inhalt 0 = LOAD Inhalt 4 = VERIFY
9D	157	Ausgabe-Kontrolle: Inhalt 000 = Programm-Modus Inhalt 128 = Direktmodus Damit bei LOAD-Befehlen vom Programm aus die Mitteilungen SEARCHING, LOADING oder VERIFYING auf dem Bildschirm erscheinen, setzt man vor dem LOAD-VERIFY- oder SAVE-Befehl ein POKE157,128.
B6	182	PEEK (182) ruft die Anzahl der Zeichenlesefehler ab.
B7	183	PEEK (183) gibt die Länge des File-namens an.
B8	184	Die laufende Filenummer kann mit PEEK (184) abgerufen werden.
B9	185	Die aktuelle Sekundäradresse kann mit PEEK (185) angegeben werden.
BA	186	Die derzeitige Gerätenummer wird

Hex	Dezimal	Beschreibung	Hex	Dezimal	Beschreibung
		durch PEEK (186) abgerufen.			Bit 0 = Shift-Taste, Bit 1 = Commodore-Taste und Bit 2 = Control-Taste.
C5	197	Derzeitiger Tastendruck: PEEK (197)	291	657	Nach POKE 657,128 ist eine Umschaltung Groß-/Kleinschrift von der Tastatur her unmöglich. POKE 657,0 erzeugt wieder den Normalzustand.
C6	198	Nach WAIT 198,1 wartet der Computer auf eine gedrückte Taste.	302	770	Durch EinPOKEn eines beliebigen Wertes erfolgt die READY-Ausgabe unendlich oft, nur noch Ausschalten hilft.
C7	199	RVS Flag: eine reverse Zeichenausgabe bei PRINT erfolgt nach POKE 199,1. POKE 199,0 hebt den RVS-Modus auf.	307	775	POKE 775,Wert aktiviert einen Listschutz. Nur über den Wert 167 kann der Listschutz aufgehoben werden.
C8	200	Zeiger auf Zeilenende. PEEK (200) gibt an, wieviel Zeichen die zuletzt eingegebene Zeile hatte.	308	776	Der Befehl POKE 776,1 zerstört das Programm.
C9/CA	201/202	PEEK (201) gibt die Zeile der aktuellen Cursorposition an, während PEEK (202) die Spalte abrufen.	309	777	Nach POKE 777,1 wird kein Befehl mehr ausgeführt. Der Cursor befindet sich in der linken Ecke.
CB	203	Die derzeitig gedrückte Taste im CHR\$-Modus kann mit PEEK (203) abgefragt werden. Ist keine Taste gedrückt, so steht der Wert 64 in diesem Byte.	30D/30E	781/782	Startadresse, ab der ein Programm geladen wird: Durch entsprechende POKE-Werte kann ein Basic-Programm in einen anderen Speicherbereich geladen werden.
CC	204	Nach POKE 204,0 bleibt der Cursor an, auch bei GET-Befehlen. Mit POKE 207,0 : POKE 204,1 kommt man dann wieder auf den Normalzustand zurück.	310/311	784/785	USR-Vektor: Erfolgt der Einsprung in ein Maschinenprogramm über den USR-Befehl, so muß die Einsprungsadresse vorher in diese beiden Bytes gePOKEt werden.
D3	211	POKE 211,Wert setzt die Spalte der Cursorposition. Werte von 0 bis 39 können eingegeben werden. Siehe auch 214.	314/315	788/789	IRQ, Hardware-Interrupt: Das Betriebssystem springt ständig in diese Routine, durch Ändern des Inhalts kann man eigene, »interruptgesteuerte« Maschinenroutinen ständig laufen lassen.
D5	213	Länge der momentanen Bildschirmzeile: über PEEK (212) läßt sie sich abfragen.			POKE 788,49 hebt die Wirkung der Stop-Taste auf. POKE 788,52 schaltet sie wieder ein.
D6	214	POKE 214,Wert setzt die Zeile der Cursorposition. Die Werte 0-24 können eingegeben werden. Nach SYS 58640 erscheint der Cursor auf der entsprechenden Position.	318/319	792/793	Restore-Vektor: PEEK (792)+PEEK (793)*256 ergibt die Speicherstelle, an die bei Restore-Tastendruck gesprungen wird.
D7	215	Das zuletzt eingegebene Zeichen im ASCII-Code liest man mit PEEK (215) aus.			Beispiel: Bei POKE 792,226 : POKE 793,252 wird beim Drücken der Restore-Taste ein Reset ausgelöst.
277-280	631-640	Tastaturpuffer (ASCII-Code): Über den POKE-Befehl lassen sich beispielsweise Zeichen in den Tastaturpuffer eingeben.	321/322	801/802	Durch die Kombination POKE 801,0 : POKE 802,0 : POKE 818,165 wird ein SAVE-Schutz aktiviert.
281-284	641-644	Start- und Endadresse des Basic-RAMs: Durch Ändern dieser Werte kann man die Größe des Basic-RAMs verändern, zum Beispiel: POKE 643,0 : POKE 644,128 : SYS 64764 setzt das Ende des Basic-RAMs um 8 KByte nach unten. Anderes Beispiel: POKE 641,0 : POKE 642,16 : SYS 64764 setzt das Basic-RAM um 2 KByte nach oben.	328	808	POKE 808,225 hebt die Stop-Taste auf, POKE 808,237 reaktiviert sie.
286	646	POKE 646,Farbwert setzt die Cursorfarbe.	32D	813	Nach POKE 813,2 läßt sich ein Programm nicht mehr ändern.
287	647	Farbe unter dem Cursor: Mit den Werten 0 bis 15 werden die jeweiligen Farben gePOKEt.	332/333	818/819	Folgende Kombinationen bewirken ebenfalls einen SAVE-Schutz: 1. POKE 818,116 : POKE 819,196 2. POKE 818,34 : POKE 819,253
289	649	Länge des Tastaturpuffers: Nach POKE 649,0 wird die gedrückte Taste nicht mehr auf den Bildschirm ausgegeben. Die maximale Größe des Tastaturpuffers beträgt 10.	33C-3FB	828-1019	Kassettenpuffer: Nach LOAD oder VERIFY stehen im Kassettenpuffer folgende Informationen: 828: 1 = normales File, 3 = wurde mit SAVE "Name",1,1 abgespeichert. Solche Programme werden bei LOAD automatisch ab der Adresse geladen, von der sie abgespeichert wurden. 829/830: Hier ist die Startadresse des Programms abgelegt (829 ist das Low-Byte, 830 das High-Byte). 831/832: Endadresse des Programms. 833-1019: 186 Zeichen langer Programmname (auf dem Bildschirm werden nur 16 angezeigt, aber es lassen
28C	652	Zähler für REPEAT-Verzögerung: Alle eingePOKEten Werte verzögern die Repeat-Funktion. Höherer Wert = größere Verzögerung.			
28D	653	PEEK für Shift-, Commodore- und für CTRL-Taste:			

Hex	Dezimal	Beschreibung	Hex	Dezimal	Beschreibung
		sich bis 186 Stellen lange Programmnamen abspeichern). Der Kassettenpuffer ist auch gut zum Ablegen eigener Maschinenprogramme geeignet, sofern mit der Floppy gearbeitet wird.	DD00	56576	Mit PEEK(56576) kann man die Pins PB0-PB7 vom User-Port (auf der Unterseite des Ports, siehe Handbuch) auslesen. Mit POKE in diese Speicherstelle kann man auch Ausgaben über den User-Port laufen lassen.
9006	36870	PEEK (36870) liest die Horizontal-Position des Lichtgriffels.	DD02	56578	Datenrichtungsregister für User-Port: Jedes der Bits gibt die Datenrichtung für die Pins PB0-PB7 des User-Ports an (Bit gesetzt = Ausgang, Bit nicht gesetzt = Eingang).
9007	36871	PEEK (36871) liest die Vertikal-Position des Lichtgriffels.	FF81	65409	SYS 65409 setzt den Video-Chip des C64 auf den Ursprungszustand zurück.
	42291	Koppeladressen angleichen: Falls Programme mit NEW gelöscht wurden, kann man mit diesem SYS-Befehl die Byte 2049 und 2050 wieder in Ordnung bringen, wenn vorher etwas anderes als 0 in diese Speicherzellen gePOKEt wird.	FFD5	65493	LOAD-Routine des Betriebssystems. Mit folgender kleiner Routine kann man Unterprogramme nachladen, ohne irgendwelche Basic-Pointer (wie zum Beispiel die Zeiger auf die Endadresse, 45 und 46) zu verändern: POKE 186,1 : POKE 780,0 : POKE 781,0 : POKE 782,96 : POKE 183,0 : SYS 65493 Erklärung: 186,1 = Geräteadresse für Recorder 781 und 782 gibt die Startadresse an, ab der das Programm geladen werden soll. 183,0 = kein Programmname. SYS 65493 = LOAD-Routine.
DO16	53270	VIC-Control-Register: Ein horizontales Softscrolling kann gePOKEt werden. POKE 53270,0 = scrollen nach links POKE 53270,1 = scrollen nach rechts	FFE7	65511	Durch SYS 65511 lassen sich alle Files schließen. So erspart man sich das lästige Eintippen von CLOSE1:CLOSE2:CLOSE3... Dabei sollte aber beachtet werden, daß auf diese Weise nur der Kanal geschlossen wird, aber keine Dateien auf einer Disk.
DO20	53280	POKE 53280,Wert setzt die Rahmenfarbe des Bildschirms.			
DO21	53281	POKE 53281,Wert setzt die Hintergrundfarbe auf dem Bildschirm.			
DC00	56320	Joystick Port 2: WAIT 56320,16,16 wartet auf Feuerknopf WAIT 56320,4,4 wartet auf Linksbewegung des Joysticks WAIT 56320,1,1 wartet auf Joystick nach oben WAIT 56320,2,2 wartet auf Joystick nach unten WAIT 56320,8,8 wartet auf Joystick nach rechts			
DC01	56321	Wie 56320, aber Joystick in Port 1.			
DC05	56325	Die Blinksequenz des Cursors kann in diesem Byte bestimmt werden. Höherer Wert = langsamere Blinksequenz. Mit POKE 56325,58 wird die normale Blinksequenz festgelegt.			
DC90	56464	Nach WAIT 56464,16,16 wartet der Computer auf den Feuerknopf des Joysticks 2.			

Grafik und Sprites wurden im Grafikkurs von H. Ponnath und im 64'er Sonderheft 4 sehr ausführlich beschrieben. Auf PEEKs und POKes für diesen Bereich wurde daher verzichtet.

(M. Kohlen/kn)

Die Modulfabrik

Ganz gleich, ob Sie ein Maschinensprache- oder Basic-Programm in einem EPROM speichern möchten - mit dem Modulgenerator geht es blitzschnell und komfortabel.

Der Modulgenerator zeichnet sich gegenüber vielen zum Teil mit der Hardware (EPROM-Programmiergerät) mitgelieferten und anderen käuflichen Programmen dieser Art durch seine einfache, menügesteuerte Bedienung aus. Er ermöglicht das komfortable Generieren eines aus bis zu fünf Programmen (Basic oder Maschinensprache)

bestehenden Auto-Start-Moduls mit Menüsteuerung. Dieses Modul kann anschließend mit der Treibersoftware des EPROM-Brenners auf ein EPROM übertragen werden.

Vor der Beschreibung des Modulgenerators selbst sollen zuvor der Aufbau und die Funktionsweise der von ihm generierten Programme erläutert werden. Sie belegen den Bereich von \$8000 bis \$A000 und eignen sich somit zum Brennen auf ein 0, beziehungsweise \$AFFF, 8 KByte oder 16 KByte EPROM. Hinter der Modulkennzeichnung »CBM 80« für den Autostart befindet sich ein kurzes Steuerprogramm zur Tastaturabfrage, zum Verschieben und Starten der gespeicherten Programme; dahinter sind eine Liste der Programmnamen sowie eine Tabelle mit wichtigen Adressen:

Byte 1 und 2:	Anfangsadresse des Programms im Modul
Byte 3 und 4:	Endadresse+1 des Programms im Modul
Byte 5 und 6:	Endadresse+1 des Programms nach dem Verschieben
Byte 7 und 8:	Startadresse des Programms