

64'er Extra

64'er Extra

Das 64'er Extra bringt geballte Information über Ihren C 64 zum Heraustrennen und Sammeln.

In dieser siebten Ausgabe finden Sie den dritten Teil einer Übersicht über alle ROM-Routinen des C 64. Statt ziellos in ROM-Listings zu blättern, finden Sie hier im Klartext die Funktionsbeschreibung aller irgendwie nutzbaren Routinen.

LINES \$ECFO
Niederwertiges Byte der Tabelle der Bildschirmzeilen.

TALK \$ED09
Kernel-Routine TALK.

LISTN \$ED0C
Kernel-Routine LISTN.

SEND \$ED16
Sendet mit OR behandeltes Zeichen.

SET \$ED22
Sendet 8 Datenbits.

SECDND \$ECC0
Kernel-Routine SECOND.

SCATR \$EDB8
Gibt ATN frei.

TKSA \$EDC7
Kernel-Routine TKSA.

CIOUT \$EDD0
Kernel-Routine CIOUT.

UNTKL \$EDEF
Kernel-Routine UNTALK.

UNLSN \$EDF8
Kernel-Routine UNLSN.

DLABYE \$EE03
Gibt alle Leitungen frei.

ACPTR \$EE13
Kernel-Routine CPTR.

ACPOOC \$EE47
»TIMEOUT« Ausführung.

ACPO1 \$EE56
Empfängt 8 Datenbits.

CLKHI \$EE85
Clock-Leitung high

CLKLO \$EE88
Clock-Leitung low

DATAHI \$EE97
Data-Leitung high

DATALO \$EEA0
Data-Leitung low

DEBCIA \$EEA9
CIA-Entprellung.

WLMS \$EEB3
Verzögerungsschleifen.

RSTRAB \$EEBB
Teil der Routine, die von NMI verwendet wird, wenn er RS232-Übertragung bedient.

RST010 \$EED7
Berechnet Parität. Beim Eintritt ist NXTBIT=0.

RST050 \$EF00
Verarbeitung der Stop-Bits.

RSTBGN \$EF06
Einsprung: Beginnt die Übertragung eines Bytes.

RST060 \$EF13
Bereitet das Senden des nächsten Bytes vor.

DSERR \$EF2E
Legt RS232-Fehler in ST ab.

BITCNT \$EF4A
Ermittelt die Zahl der zu sendenden Bits+1.

RSRCVR \$EF59
Sammelt Bits zu einem Byte während NMI.

RSR030 \$EF97
Überträgt Daten in den Puffer für Paritätsprüfung.

ECERR \$EFCA

Meldet Empfangsfehler.

CK0232 \$EF01

Gibt über die RS232-Schnittstelle am USER-Port eine Datei aus.

BS0232 \$F014

Gibt ein RS232-Zeichen aus.

BS0110 \$F02E

Macht die MNIs des Timers 1 in CIA scharf (unterschiedlich zwischen den Kernelversionen 0 und 3).

CK1232 \$F04D

Liest über die RS232-Schnittstelle (über den Usen-Port) eine Datei ein.

BS1232 \$F086

Liest ein RS232-Zeichen ein.

RSP232 \$F0A4

Schützt seriellen Bus und Bandbetrieb vor NMLs.

SPMSG \$F12B

Gibt die Bandbetriebsmeldung von der Tabelle ab \$F0BD aus, falls durch Flag bei \$9D freigegeben.

GETIN \$F13E

Kernel-Routine GETIN.

NBASIN \$F157

Kernel-Routine CHRIN.

JTGET1 \$F179

Holt ein Zeichen vom Band.

BSCOUT \$F1CA

Kernel-Routine CHROUT.

CASOUT \$F1DD

Sendet ein Zeichen zum Kassettengerät.

NCHKIN \$F20E

Kernel-Routine CHKIN.

NCKOUT \$F250

Kernel-Routine CHKOUT.

NCLOSE \$F291

Kernel-Routine CLOUSE.

NCLALL \$F32F

Kernel-Routine CLALL.

NOPEN \$F34A

Kernel-Routine OPEN.

OP152 \$F38B

OPEN für Kassettendatei.

OPENI \$F305

OPEN für Datei für ein serielles Gerät (Drucker, Diskettenlaufwerk).

OPN232 \$F409

OPEN für eine RS232-Datei.

LOADP \$F49E

Kernel-Routine LOAD.

LD25 \$F4BF

LOAD für Diskette.

LD102 \$F539

LOAD für Kassettengerät.

SAVEP \$F5DD

Kernel-Routine SAVE.

SV21 \$F5FA

SAVE für Diskette.

SV102 \$F65F

SAVE für Kassettengerät.

UDTIMK \$F69B

Kernel-Routine UDTIM.

RDTIMK \$F6D0

Kernel-Routine RDTIM.

SETTMK \$F6E4

Kernel-Routine SETTIM.

NSTOP \$F6ED

Kernel-Routine STOP.

ERROR1 \$F6FB

Tabelle der I/O-Fehlernummern 1...9 und -meldungen.

FAH \$F72C

Lädt nächste Kassettenvorspann.

TAPEH \$F76A

Schreibt Kassettenvorspann.

FAF \$F7EA

Lädt angegebenen Kassettenvorspann.

TRD \$F84A

Lädt vom Kassettengerät.

WTWT \$F867

Schreibt zum Kassettengerät.

READ \$F92C

Kassetten-Lese-Routinen.

WRITE \$FB46

Routinen für Schreiben auf Kassette.

START \$FCE2

Reset-Routine. Einschalten des »64« oder ein Reset-Schalter verursachen einen Sprung hierher. SYS 64738 springt hierher, es sei denn, das RAM unter dem Kernel ist aktiviert. Ist bei \$8000 ein Modul vorhanden, erfolgt JMP (\$8000), andernfalls laufen RAMTAS, RESTOR, IOINIT, CINT und ein Basic-Kaltstart ab. Sonstiges RAM bleibt unverändert; Basic-Programme lassen sich daher wieder gewinnen.

RESTOR \$FD15

Kernel-Routine RESTOR.

VECTOR \$FD1A

Kernel-Routine VECTOR.

RAMTAS \$FD50

Kernel-Routine RAMTAS.

I0INIT \$FD43

Kernel-Routine I0INIT.

SETNAM \$FDFF

Kernel-Routine SETNAM.

SETLFS \$FE00

Kernel-Routine SETFLS.

READST \$FE07

Kernel-Routine READST.

SETMSG \$FE18

Kerneler-Routine SETMSG.

SETTMO \$FE21

Kernel-Routine SETTMO.

MEMTOP \$FE25

Kernel-Routine MEMTOP.

MEMBOT SFE34

Kernel-Routine MEMBOT.

NMI SFE43

NMI-Routine: Alle NMIs laden hier (es sei denn, der Kernel wurde im RAM modifiziert) und werden über den Vektor in (\$0318) gelenkt. STOP-RESTORE, RS232-Betrieb und benutzerdefinierte NMIs werden sämtlich hier verarbeitet.

TIMB SFE66

POST-RESTORE oder die BRKs »Not-Rücksetz-Routine«.

NNM121 SFE66

Prüft RS232-Schnittstelle und sendet Bit, wenn möglich.

NNM130 SFEA3

Prüft RS232-Schnittstelle und empfängt Bit, wenn möglich.

BAUDOT SFE2

Baudaten-Tabelle für RS232

T2NMI SFED6

Behandelt Eingang eines RS232-Bits.

FLNMI SFF07

Behandelt zeitliche Abstimmung des Startbits für RS232.

PULS SFF48

Einsprung für IRQ oder BRK. Alle IRQ-Interrupts laufen über diesen Punkt (es sei denn, der Kernel wurde am RAM modifiziert). IRQ lenkt ein Vektor in (\$0314), BRK-Befehle ein Vektor in (\$0316); Änderungen beider Routinen durch den Benutzer sind daher häufig.

CINT SFF81

Setzt die Interrupt-Frequenz.

IINIT SFF84

Initialisiert die Ein-/Ausgabe-Chips.

RAMTAS SFF87

Testet und setzt RAM.

RESTOR SFF8A

Restauriert Standard-Ein-/Ausgabevektoren.

VECTOR SFF8D

Speichert/setzt Ein/Ausgabevektoren.

SETMSG SFF90

Gibt Meldung zum Bildschirm aus.

SECOND SFF93

Sendet Sekundäradresse nach LISTEN über den seriellen Bus.

TKSA SFF96

Sendet Sekundäradresse nach TALK über den seriellen Bus.

MEMTOP SFF99

Liest/setzt die obere Grenze des Speichers für Basic.

MEMBOT SFFB2

Liest/setzt die untere Grenze des Speichers für Basic.

SCNKEY SFF9F

Naturabfrage.

SETTMO SFFA2

Setzt TIMEOUT für seriellen Bus.

ACPTR SFFA5

Holt ein Byte von einem seriellen Gerät (gewöhnlich Floppy-Disk).

CIOUT SFFAB

Gibt ein Byte zum seriellen Bus aus (gewöhnlich für Drucker oder Floppy-Disk).

UNTALK SFFAB

desaktiviert sendende Geräte am seriellen Bus.

UNLSN SFFAE

desaktiviert empfangende Geräte am seriellen Bus.

LISTN SFFB1

Schaltet Gerät am seriellen Bus auf Empfang (gewöhnlich Drucker oder Floppy-Disk).

READST SFFB7

liest das Status-Byte nach A ein.

SETLFS SFFB8

Setzt Dateinummer, Gerät, Sekundäradresse.

SETNAM FFBD

Setzt Dateinamen.

OPEN SFFCO

Öffnet Datei zum Lesen/Schreiben.

CLOSE SFFC3

Schließt eine Datei.

CHKIN SFFC6

Eröffnet Eingabekanal.

CHKOUT SFFC9

Eröffnet Ausgabekanal.

CLRCHN SFFCC

Schließt Kanal, stellt die Standardbedingungen für I/O wieder her.

CHRIN SFFCF

Holt ein Zeichen.

CHROUT SFFD2

Gibt ein Zeichen aus.

LOAD SFFD5

Lädt ein Programm (von Diskette oder Band).

SAVE SFFD8

Speichert einen Bereich als Programm (auf Diskette oder Band).

SETTIM SFFDB

Setzt Taktzähler. (BASIC 2 VERIFY)

RDTIM SFFDE

Liest Taktzähler. (BASIC 2 SYS)

STOP SFFE1

Prüft auf die STOP-Taste.

GETIN SFFE4

Holt ein Zeichen (gewöhnlich von der Tastatur — GET).

CLALL SFFE7

Beendet jeglichen I/O-Betrieb und schließt alle Dateien.

UDTIM SFFEA

Addiert zum Taktzähler; auf Null stellen, wenn 240 000.

SCREEN SFFED

Holt Anzahl von Zeilen und Spalten.

PLOT SFFFF

Liest/setzt Cursor.

OBASE SFFF3

Startadresse des Tastatur-VIA.

Nützliche PEEKs und POKEs

PRINT PEEK(17)Anzeige von 0: die letzte Eingabe wurde über die Anweisung INPUT vorgenommen
Anzeige von 64: die letzte Eingabe wurde über die Anweisung GET vorgenommen
Anzeige von 152: die letzte Eingabe wurde über READ vorgenommen**PRINT PEEK(43)+256*PEEK(44)**

Ausgabe der Adresse, von wo aus das Basic-Programm gespeichert ist

PRINT PEEK(45)+256*PEEK(45)

Ausgabe der Adresse, bis wohin das Basic-Programm reicht

PRINT PEEK(47)+256*PEEK(48)

Ausgabe der Adresse, bis wohin die Variablen reichen

PRINT PEEK(49)+256*PEEK(50)

Ausgabe der Adresse, bis wohin die Felder reichen

PRINT PEEK(55)+256*PEEK(56)

Ausgabe der Adresse, bis wohin der Basic-Arbeitsspeicher reicht

PRINT PEEK(57)+256*PEEK(58)

Ausgabe der Nummer der momentan bearbeiteten Programmzeile

PRINT PEEK(63)+256*PEEK(64)

Ausgabe der Nummer derjenigen DATA-Zeile, aus der Daten entnommen werden (READ-Befehl)

PRINT CHR\$(PEEK(69)AND127) + CHR\$(PEEK(70)AND127)

Ausgabe des zuletzt benutzten Variablennamens

PRINT PEEK(198)

Ausgabe der Anzahl der im Tastaturpuffer gespeicherten Zeichen

POKE 198,0

Der Tastaturpuffer wird geleert

PRINT PEEK(214)

Ausgabe der Nummer der Zeile, in der sich der Cursor befindet; dabei werden die Zeilen von 0 bis 24 (= 25 Zeilen) gezählt

POKE 214,Z : POKE 211,S : SYS 58640

Diese Befehlssequenz setzt den Cursor unmittelbar auf die mittels Z und S angegebene Zeilen- beziehungsweise Spaltenposition, wobei die Zeilen von 0 bis 39 und die Spalten von 0 bis 24 gezählt werden

PRINT PEEK(641)+256*PEEK(642)

Ausgabe der Adresse, von wo ab der für Basic nutzbare RAM-Bereich beginnt

PRINT PEEK(643)+256*PEEK(644)

Ausgabe der Adresse, bis wohin der für Basic nutzbare RAM-Bereich reicht

POKE 641,AL : POKE 642,AH : SYS 58260

Die Anfangsadresse für Basic-Programme wird auf den Wert AL+256*AH festgelegt

POKE 643,EL : POKE 644,EH : SYS 58260

Die Endadresse für den Basic-Arbeitsspeicher wird auf den Wert EL+256*EH festgelegt

PRINT PEEK(646)

Anzeige der Nummer der momentanen Zeichenfarbe

POKE 646,F

Einstellen der Zeichenfarbe, wobei F der Farbnummer entspricht

POKE 649,MX

MX ist eine Zahl zwischen 0 und 10; diese POKE-Anweisung legt fest, wieviele Zeichen sich maximal im Tastaturpuffer befinden dürfen (der Grundwert ist 10); wenn also MX=0 gewählt wird, ist die Tastatur völlig abgeschaltet und keine Eingabe mehr ist möglich (Vorsicht!)

POKE 650,128

Die Tasten-Wiederholungs-Automatik wird auf alle Tasten ausgedehnt

POKE 650,64

Die Tasten-Wiederholungs-Automatik wird für alle Tasten abgeschaltet

POKE 650,0

Die Tasten-Wiederholungs-Automatik wird für die INST DEL-, die Leertaste und die Cursortasten eingeschaltet (Grundwert)

PRINT PEEK(653) AND 7

Anzeige von 0: keine der Tasten SHIFT, CBM oder CTRL gedrückt

Anzeige von 1: die SHIFT-Taste ist gedrückt

Anzeige von 2: die CBM-Taste befindet sich in gedrückter Haltung

Anzeige von 4: die CTRL-Taste befindet sich in gedrückter Haltung

Anzeige von 3,5,6,7: diese Zahlen ergeben sich aus der Addition zweier oder aller drei oben beschriebenen Werte und geben somit an, daß sich die den Summanden zugeordneten Tasten in gedrückter Haltung befinden

POKE 657,128

Umschaltungsverriegelung von dem Großschrift/Grafik-Modus in den Klein-/Großschrift-Modus

POKE 657,0

Aufheben der Verriegelung (siehe voriger POKE-Befehl)

POKE 775,200

LIST-Schutz einschalten

POKE 775,167

LIST-Schutz ausschalten

POKE 788,52

RUN-STOP-Taste wirkungslos machen

POKE 788,49

RUN-STOP-Taste reaktivieren

POKE 801,0 : POKE 802,0 :

Nach dieser Befehlssequenz wird das Speichern eines Programms unmöglich gemacht

POKE 808,225

RUN-STOP- + RESTORE-Taste wirkungslos machen, außerdem erscheint ein »verrücktes« Listing

POKE 808,237

RUN-STOP- + RESTORE-Taste reaktivieren

POKE 53265,11

Bildschirminhalt unsichtbar machen, ohne daß dessen Inhalt verloren geht

POKE 53265,27

Bildschirminhalt wieder sichtbar machen

WAIT 56320,16,16

Warten, bis beim Joystick Port 2:

der Feuerknopf gedrückt wird,

WAIT 56320,4,4

Linksbewegung vorgenommen wird,

WAIT 56320,1,1

Bewegung nach oben erfolgt,

WAIT 56320,2,2

Bewegung nach unten erfolgt,

WAIT 56320,8,8

Rechtsbewegung vorgenommen wird

WAIT 56321,16,16

Warten, bis beim Joystick Port 1:

der Feuerknopf gedrückt wird,

(Richtungsabfragen analog zu Joystick Port 2)