

# Neuer Checksummer 64 — blitzschnell und kürzer

**Das Programm Checksummer kennen die meisten unserer Leser bereits aus dem 64'er Sonderheft und den Ausgaben 1 und 2/85.**

**Der Checksummer überprüft jede eingegebene Basic-Zeile und erspart Ihnen deshalb eine langwierige Fehlersuche.**

**Und neu ist der Checksummer, weil die Initialisierung zirka einhundertmal schneller vonstatten geht als bei der alten Version.**

Es gab eine riesige Resonanz auf den Checksummer und zwar ausschließlich positiv. Jedoch brauchte das Programm eine ganze Weile, bis es initialisiert war, eine Herausforderung für eine Unmenge von Lesern! Der nebenstehende Leserbrief setzt sich drastisch, aber nicht ganz so ernst gemeint, mit diesem Problem auseinander und gibt eine Lösung, die zirka einhundertmal schneller ist als die bisherige Version. Dieser Brief und der neue, schnelle Checksummer steht übrigens für eine ganze Reihe von Leserschriften, die auch entsprechende Verbesserungen eingeschickt haben.

Der Checksummer 64 ist ein kleines Maschinenprogramm, das, wenn es aktiviert ist, Sie sofort davon unterrichtet, ob Sie die jeweilige Programmzeile korrekt eingegeben haben.

1. Tippen Sie den Basic-Lader sorgfältig ein. Es gibt zwei Versionen: Eine für den Commodore 64 und eine für den VC 20.

2. Vergessen Sie nicht das Abspeichern vor dem Start (RUN)!

3. Wenn Sie den Checksummer 64 zwischenzeitlich nicht benutzen, können Sie ihn jederzeit mit »POKE 1,55« deaktivieren. Auch durch Drücken der Run-Stop- und der Restore-Taste wird der Checksummer 64 deaktiviert.

Wollen Sie den Checksummer 64 wieder einschalten, so geben Sie bitte »POKE 1,53« ein.

Das Maschinenprogramm bleibt solange erhalten, bis der Computer ausgeschaltet, oder, wenn von anderen Programmen auf das hinter dem ROM liegende RAM zugegriffen wird.

Alle veröffentlichten Listings sind mit einer Checksumme versehen, die am Ende jeder Programmzeile steht. Diese Checksumme steht zwischen < und >. Sie wird beim Eintippen des Programms nicht mit eingegeben. Die Zahl zwischen den beiden Zeichen stellt lediglich eine Information für Sie dar. Wenn Sie diese Checksumme dennoch mit eintippen, werden Sie schnell bemerken, daß Sie etwas falsch gemacht haben. Bei aktiviertem Checksummer 64 wird nämlich nach Eingabe einer Basic-Zeile, die mit Return beendet wird, in die linke obere Bildschirmcke die Checksumme eingeblendet, die mit der Summe aus dem veröffentlichten Listing übereinstimmen muß. Ist das nicht der Fall, haben Sie die Zeile anders eingegeben, als sie im Listing dargestellt ist. Vergessen Sie also bitte nicht, daß die am Ende einer Zeile in < und > stehende Prüfsumme nicht mit eingegeben werden darf.

Der Checksummer 64 ist so ausgelegt, daß er abhängig von der Zeilennummer und dem Text der Zeile eine Checksumme ausgibt. Beim Bilden dieser Checksumme werden Spaces (Leertaste) überlesen, was für Sie bedeutet, daß es egal ist, wieviel Leerzeichen Sie zwischen den Worten lassen, da Sie für den Programmablauf ohnehin keine Bedeutung haben.

## Herrschaften!

(Nehmen Sie mir den Ton dieses Schreibens bitte nicht übel; ich meine es nicht so ganz ernst!)

...da veröffentlicht Ihr in Eurer Zeitschrift Programme wie HYPRA-LOAD und so weiter, weil die geschätzten Fans, überwiegend wohl Hobbyisten, nicht genug Zeit haben (O tempora, o mores!), auf die lahme, dafür aber billige 1541 zu warten, und dann schockt Ihr einen mit einer für den Genuß Eures Magazins lebenswichtigen Utility, nämlich dem Checksummer, während deren Setup man einen ausgedehnten Mittagsschlaf halten kann. Mein C 64-Interpreter (der übrigens Wert darauf legt, auf der 2. Silbe betont zu werden, was einerseits richtig, andererseits wenig bekannt ist) behauptet jedenfalls, er brauche für den Programmlauf 1 Minute 18 Sekunden...

Wenn Ihr Euch jetzt auf die Füße getrampelt fühlt und um konstruktive Kritik bittet, bitte:

Der erste Blick ins Listing entlarvt den ersten Übeltäter in den Zeilen 130 und 140 (wohlgemerkt: die C 64-Version!):

16 KByte in einer Basic-Schleife kopieren! Puh! Warum nicht gleich von Hand einPOKEN?! Dieser Anschlag auf die Geduld des geneigten Lesers und Anwenders ist um so verwerflicher, als daß doch der Interpreter ab Adresse dezimal 41912 (\$A3B8) eine Block-Verschiebe-Routine anbietet, mit der man nur umzugehen wissen muß.

Also:

Die ersten 3 Befehle der Routine vergessen wir — die werden nur gebraucht, wenn die Routine in der üblichen Weise verwendet wird. Der richtige Einsprung-Punkt für unsere Zwecke ist also dezimal 41919 (\$A3BF).

Die Routine verlangt nun folgende Übergabe-Parameter (in der üblichen Reihenfolge low — high):

a) Startadresse des zu verschiebenden (hier bloß in das darunterliegende RAM zu kopierenden) Blocks in dezimal 95/96 (\$5F/60),

b) Endadresse+1 des zu verschiebenden Blocks in dezimal 90/91 (\$5A/5B),

c) Endadresse+1 des Bereichs, in den verschoben werden soll, in dezimal 88/89 (\$58/59).

Getrennt für jeden ROM-Bereich POKEN wir also die Parameter in die angegebenen Adressen, wobei

a) in unserem speziellen Falle die Endadresse des Quell- und Zielbereichs (+1) natürlich identisch ist,

**Hinweise zum Lesen von Listings**

Die Listings haben sich ein wenig im Ausdruckformat verändert, um Ihnen das Eingeben von Programmen wesentlich zu erleichtern.

— Cursorsteuerzeichen und andere Steuerzeichen, die schwer zu lesen sind, werden von nun an in Klartext in speziellen geschweiften Klammern gesetzt.

Tritt mehrmals hintereinander dasselbe Steuerzeichen auf, so wird diese Steuerzeichen-Sequenz zusammengefaßt, indem zuerst die Anzahl der Wiederholungen dieses Steuerzeichens und dann das Steuerzeichen in Klartext ausgegeben wird.

— Alle Commodore-Grafikzeichen, die über Shift zu erreichen sind, werden nicht mehr als Grafikzeichen, sondern als Klartextzeichen dargestellt. Dabei wird aus dem Zeichen, das Sie auf dem Bildschirm sehen, Wenn Sie die Tastenkombination Shift und »A« ansprechen, wieder ein »A« (nur im abgedruckten Listing). Um dieses »A« vom normalen »A« unter-

scheiden zu können, ist es etwas kleiner als das gewöhnliche »A« und ist außerdem mit einem Unterstreichungszeichen versehen. Diese Vereinbarung gilt auch für sämtliche andere Commodore-Grafikzeichen, die über Shift zu erreichen sind.

— Entsprechendes gilt für sämtliche Commodore-Grafikzeichen, die über die Commodore-Taste zu erreichen sind. Hier wird jedoch das jeweilige Klartextzeichen nicht unterstrichen, sondern überstrichen.

**Checksummer VC 20**

Der Checksummer VC 20 ist im Prinzip genauso aufgebaut wie der Checksummer 64. Da beim VC 20 jedoch nicht die Möglichkeit besteht, das ROM softwaremäßig zu modifizieren, mußte ein anderer Weg als beim Commodore 64 gewählt werden, um die Checksumme zu generieren.

In ihrer Funktionsweise unterscheiden sich der Checksummer VC 20 und der Checksummer 64 nicht. Es gelten folgende Sonderregelungen bei der Benutzung des Checksummer VC 20:

— Da der Basic-Bereich nicht belegt werden soll, ist das Programm im Kassettenpuffer abgelegt.

b) wir im 2. Falle anstelle der richtigen, aber leider unmöglichen, Endadresse+1 einfach naßforsch 0 eingeben, getreu dem — in diesem Falle richtigen — Motto: Für den Computer ist dezimal 65535 (\$FFFF) + 1 = 0!

Aber ... (vor den Ruhm haben die Götter die Schweißfüße gesetzt!) das funktioniert natürlich (von Basic aus) nicht! Der Grund ist für den Kundigen klar: Der Aufruf der SYS-Routine mißhandelt in brutalster Weise die gerade mühsam gePOKEte Startadresse. Das entmutigt uns jedoch in keinster Weise. Wir übergeben die Startadresse in dezimal 780/781 (\$030C/030D) an Akku und X-Register und springen zunächst eine selbstgestrickte Mini-Maschinenroutine von insgesamt 7 Byte Länge an, die die Startadresse abliefern und dann die eigentliche Routine anspricht: STA \$5F, STX \$60, JMP \$A3BF. Diese 7 Byte, die wir natürlich im Programm vorher mittels einer READ-Schleife einPOKEten, verstecken wir nicht-störend vor dem Kassetten-Puffer ab dezimal 820 (\$0334). Wem dieses stille Örtchen nicht paßt: Im beigegeführten Listing (Checksummer für den C 64) habe ich diese Startadresse netterweise parametrisiert.

That's all! Hört sich komplizierter an, als es ist!

Fragt sich (mit Fug und Recht): Was bringt's?

Antwort: Das so verunstaltete Programm behelligt den Computer für nur noch 7 Sekunden anstelle von vorher 1 Minute 18 Sekunden.

Das ist natürlich noch viel zu lang!!!

Ein zweiter Blick ins ursprüngliche Listing entlarvt den zweiten Übeltäter. Die Hex-Zahlen in den DATAs samt den dazugehörigen Programmzeilen. Was Ihr Euch dabei gedacht habt, wißt Ihr und die Götter — ich nicht! Jedenfalls sind hier die Nebenwirkungen schlimmer als der Heileffekt.

Also:

Wir ersetzen die Hex-Zahlen durch stinknormale Dezimalzahlen, schmeißen natürlich die dann überflüssige Umwandlungs-Routine weg und entledigen uns letztlich (spätestens, nachdem wir die korrekte Eingabe der Zahlen gecheckt haben) der Prüfsummen-Befehle.

Ich verweise den entsetzten Leser auch insoweit auf das abgedruckte Listing.

Was bringt's?

Das Programm vergeudet nun nur noch zirka 1 Sekunde! Und damit wollen wir es genug sein lassen.

(Werner Wenzel)

Wenn Sie lesen ! drücken Sie

CTRL steht für Control-Taste, so bedeutet[CTRL-A], daß Sie die Control-Taste und die Taste »A« drücken müssen. Im folgenden steht:

{down}	! Taste neben rechten Shift, Cursor runter
{up}	! Shift-Taste & Taste neben rechtem Shift, Cursor hoch
{clear}	! Shift-Taste & 2. Taste ganz rechts oben
{inst}	! Shift-Taste & Taste ganz rechts oben
{home}	! 2. Taste von ganz rechts oben
{del}	! Taste ganz rechts oben
{right}	! Taste ganz rechts unten
{left}	! Shift-Taste & Taste unten rechts
{space}	! Leertaste
{f1}	! grauer Tastenblock rechts
{f3}	! grauer Tastenblock rechts
{f5}	! grauer Tastenblock rechts
{f7}	! grauer Tastenblock rechts
{f2}	! grauer Tastenblock rechts & Shift
{f4}	! grauer Tastenblock rechts & Shift
{f6}	! grauer Tastenblock rechts & Shift
{f8}	! grauer Tastenblock rechts & Shift
{return}	! Shift-Taste & Return
{black}	! Control-Taste & 1
{white}	! Control-Taste & 2
{red}	! Control-Taste & 3
{cyan}	! Control-Taste & 4
{purple}	! Control-Taste & 5
{green}	! Control-Taste & 6
{blue}	! Control-Taste & 7
{yellow}	! Control-Taste & 8
{rvson}	! Control-Taste & 9
{rvoff}	! Control-Taste & 0
{orange}	! Commodore-Taste & 1
{brown}	! Commodore-Taste & 2
{lig.red}	! Commodore-Taste & 3
{grey 1}	! Commodore-Taste & 4
{grey 2}	! Commodore-Taste & 5
{lig.grenn}	! Commodore-Taste & 6
{lig.blue}	! Commodore-Taste & 7
{grey 3}	! Commodore-Taste & 8

**Die in unseren Listings verwendeten Steuerzeichen und deren Bedeutung**

Wenn Sie sich erst einmal an die in Klartext geschriebenen Steuerzeichen gewöhnt haben, werden Sie den Vorteil dieser Schreibweise erkennen. Der zu dem jeweiligen Steuerzeichen gehörende Klartext ist so verfaßt, daß Sie leicht die Taste beziehungsweise die Tastenkombination finden, die Sie drücken müssen.

— Angeschaltet wird der Checksummer VC 20 mit »SYS 955« und abgeschaltet mit »SYS 58459«.

**ACHTUNG:** Nehmen Sie keine Kassetten-Operationen vor, wenn der Checksummer VC 20 eingeschaltet ist. Da das Betriebssystem den Kassettenpuffer mit Daten belegt, kann der Checksummer VC 20 überschrieben werden, was zur Folge hat, daß sich der Rechner bei aktiviertem Checksummer VC 20 »aufhängt«. Wollen Sie deshalb ein Programm auf (von) Kassette abspeichern (laden), so müssen Sie erst den Checksummer VC 20 abschalten (SYS 58459).

Daraufhin kann der Kassettenpuffer mit Daten überschrieben werden, ohne daß der Rechner »aussteigt«.

```

10 REM***** <057>
20 REM* * <247>
30 REM* CHECKSUMMER * <056>
40 REM* VERSION VC20 * <044>
50 REM* * <021>
60 REM***** <107>
70 PRINT "[CLEAR,SPACE2,RVSON]CHECKSUMMER[SPACE2]
VC-20[RVOFF]" <185>
80 PRINT <233>
90 PRINT "[DOWN]EINEN[SPACE]MOMENT,[SPACE]
BITTE..." <181>
100 FOR I=827 TO 993:GOSUB 180:POKE I,A <177>
110 PS=PS+A+1:NEXT I <049>
120 IF PS>20612 THEN PRINT "[DOWN]
PRUEFSUMMENFEHLER[SPACE]!":END <130>
130 SYS 955:PRINT "CHECKSUMMER[SPACE]AKTIVIERT."
<242>
140 PRINT "AN[SPACE]:SYS955" <212>
150 PRINT "[DOWN]AUS:SYS58459,[SPACE]BEI[SPACE]
CAS-[SPACE4]SETTE[SPACE]ZUSAETZLICH[SPACE5]
RUN/STOP[SPACE]&[SPACE]RESTORE" <068>
160 PRINT "[DOWN]BEI[SPACE]AKTIVIERTEM[SPACE]
CHECK-SUMMER[SPACE]KEIN"; <105>
170 PRINT "[SPACE]CASSETTEN-BETRIEB[SPACE](LOAD,
[SPACE]SAVE)[SPACE2]ERLAUBT!":NEW <051>
180 READ A$:IF LEN(A$)>2 THEN PRINT "TIPPFehler
[SPACE]IN[SPACE]ZEILE"PEEK(63)+PEEK(64)*256
:END <161>
190 A1=ASC(A$):A2=ASC(RIGHT$(A$,1)) <176>
200 IF A1<48 OR A1>57 THEN IF A1<65 OR A1>70 TH
EN 270 <095>
210 IF A2<48 OR A2>57 THEN IF A2<65 OR A2>70 TH
EN 270 <109>
220 IF A1>64 THEN A1=A1-55:GOTO 240 <159>
230 IF A1<58 THEN A1=A1-48 <087>
240 IF A2>64 THEN A2=A2-55:GOTO 260 <184>
250 IF A2<58 THEN A2=A2-48 <110>
260 A=A1*16+A2:RETURN <098>
270 PRINT "UNGUELTIGER[SPACE]HEXCODE[SPACE]IN
[SPACE]ZEILE"PEEK(63)+PEEK(64)*256:END <237>
280 DATA 20,5F,03,86,7A,84,7B,20 <061>
290 DATA 73,00,AA,F0,F3,A2,FF,86 <130>
300 DATA 3A,90,0A,A2,00,86,FF,20 <097>
310 DATA 79,C5,4C,E1,C7,A2,01,86 <127>
320 DATA FF,4C,9C,C4,A6,FF,E0,01 <199>
330 DATA F0,03,4C,60,C5,A0,02,A9 <125>
340 DATA 00,85,FE,B1,5F,F0,0F,C9 <186>
350 DATA 20,D0,03,C8,D0,F5,18,65 <141>
360 DATA FE,85,FE,4C,76,03,C0,04 <193>
370 DATA 30,F1,C6,D6,A5,D6,48,A2 <198>
380 DATA 03,A9,20,9D,01,04,BD,B7 <180>
390 DATA 03,20,D2,FF,CA,10,F2,A6 <217>
400 DATA FE,A9,00,20,CD,DD,A9,3E <016>
410 DATA 20,D2,FF,68,85,D6,20,87 <220>
420 DATA E5,A9,8D,20,D2,FF,A2,00 <003>
430 DATA 86,FF,F0,AE,09,3C,12,13 <002>
440 DATA A9,3B,8D,02,03,A9,03,8D <249>
450 DATA 03,03,A5,BA,C9,01,D0,10 <235>
460 DATA A9,74,8D,30,03,8D,32,03 <239>
470 DATA A9,C4,8D,31,03,8D,33,03 <007>
480 DATA AD,88,02,8D,90,03,60 <113>

```

Der Checksummer für den VC 20

Als Sicherung wird bei der Initialisierung geprüft, ob das zuletzt angesprochene Peripherie-Gerät der Kassettenrecorder war. Ist das der Fall, so werden die Betriebssystemroutinen LOAD und SAVE für die Benutzung gesperrt. Der Rechner meldet bei Aufruf einer dieser beiden Routinen READY, ohne weitere Aktionen durchzuführen. Diese Sicherung kann man nach der Tipparbeit aufheben, wenn man den Checksummer VC 20 mit SYS 58459 abschaltet. Dadurch wird der Kassettenpuffer für andere Daten freigemacht.

Weiterhin wird dann durch gleichzeitiges Drücken der Tasten »Run-Stop & Restore« erreicht, daß die Betriebssystemroutinen LOAD und SAVE wieder eingerichtet werden.

— Bei Benutzung einer Diskettenstation brauchen Sie nicht darauf zu achten, daß bei LOAD beziehungsweise SAVE der Checksummer VC 20 überschrieben wird, da der Kassettenpuffer für die Diskettenstation normalerweise nicht genutzt

```

10 REM ***** <175>
20 REM * * <247>
30 REM * CHECKSUMMER 64 * <162>
33 REM * * <004>
36 REM * (VERSION 2.0) * <014>
40 REM * * <011>
50 REM * * <061>
60 REM * * <031>
70 REM * COMMODORE 64 * <056>
80 REM * * <051>
90 REM ***** <255>
100 PRINT "[CLR,13SPACE,RVSON]CHECKSUMMER 6
4[RVOFF]" <025>
110 PRINT <007>
121 SA=820:FOR I=SA TO SA+6:READ A:POKE I,
A:NEXT I <073>
122 DATA 133,95,134,96,76,191,163 <179>
130 POKE 88,0:POKE 89,192:POKE 90,0:POKE 9
1,192:POKE 780,0:POKE 781,160:SYS SA <244>
140 POKE 88,0:POKE 89,0:POKE 90,0:POKE 91,
0:POKE 780,0:POKE 781,224:SYS SA <039>
150 POKE 1,53:POKE 42289,96:POKE 42290,228 <001>
160 FOR I=58464 TO 58554:READ A:POKE I,A:N
EXT I <100>
190 PRINT "{4DOWN,9SPACE}CHECKSUMMER AKTIVI
ERT." <247>
200 PRINT "{2DOWN}AUSSCHALTEN : POKE1,55" <050>
210 PRINT "{DOWN}ANSCHALTEN{2SPACE}: POKE1,
53":NEW <171>
320 DATA 160,2,169,0,133,2,177,95 <103>
330 DATA 240,15,201,32,208,3,200,208 <239>
340 DATA 245,24,101,2,133,2,76,110 <153>
350 DATA 228,192,4,48,241,198,214,165 <090>
360 DATA 214,72,162,3,169,32,157,1 <191>
370 DATA 4,189,183,228,32,210,255,202 <096>
380 DATA 16,242,166,2,169,0,32,205 <206>
390 DATA 189,169,62,32,210,255,104,133 <168>
400 DATA 214,32,108,229,169,141,32,210 <168>
410 DATA 255,76,128,164,92,72,32,201 <093>
420 DATA 255,170,104,144,1,138,96,9 <051>
430 DATA 60,18,19 <195>

```

64'er

Der neue Checksummer für den Commodore 64

wird. Deshalb können Sie die beiden Routinen weiterhin normal nutzen, sofern der Rechner bei der Initialisierung des Checksummer VC 20 feststellt, daß das zuletzt angesprochene Peripherie-Gerät nicht der Kassettenrecorder war.

— Bedingt durch den anderen Aufbau des Checksummer VC 20 wird anders als beim Checksummer 64 nach der LOAD-Routine keine Checksumme ausgegeben.

— Wird eine Zeile gelöscht, also eine Zahl zwischen 0 und 63999 eingegeben, und danach Return gedrückt, so wird eine Checksumme ausgegeben, die aber keine Bedeutung hat.

Viel Spaß beim Eintippen von Programmen mit dem neuen Checksummer!

(F. Lonczewski/gk)