Autor: Michail Popov

kommentiert: 13.-14.8.92

(Anm: Beim Befehl MEM kann der Wert von $01 angegeben werden, der jedoch keine Wirkung hat da ROM-Bildschirm-Reset+löschen +RTS)

KILL

Tabelle mit Befehlswörtern und Adressen

Vektoren setzen + BS-Reset
80 Bytes mit 0 ab $C4FF beschreiben LOOP } $C4FF-$C559 löschen (F-Tasten!)
Vektor für BASIC-Interpreter-Low setzen
Vektor für BASIC-Warmstart-Low setzen
Save-Vektor Low setzen
Vektor für Ausdruck werten (Low) setzen
Warmstart-Vektor (High)
Interpretenvektor (High)

neuer BASIC-Warmstart

45 ? „TOOL: (c) 1991 BY M. POPOV"
ausgeben
NEW, BASIC-Speicher löschen
Highbyte Tastaturdekodierung
Highbyte SAVE-Vektor
Highbyte Ausdruck-werten
Lowbyte Tastaturdekodierung
BASIC-Warmstart

neuer BASIC-Interpreter

Zähler Speicher CHRGET
Zähler ablegen
Byte aus Speicher holen
Null? dann Ende sonst kein neuer Befehl
7. Bit (für Ende) löschen
= laufendes Zeichen
ja
sonst Zähler erhöhen
weiterzählen im Wort
im Speicher zurückholen
LOOP, Befehlswort noch nicht zu Ende
21 erhöhen

nicht Null, nächster Adresse der Routine holen
Selbstmodifikation!
Highbyte ablegen
CHRGET-Zeiger aktualisieren
zur Routine ($FFFF? → siehe C64B)
CHRGOT, letztes Zeichen holen
alte Routine, weiter

vgl. C000 KILL
Basic-Vektoren $0300-$030B setzen

KILL-Befehl

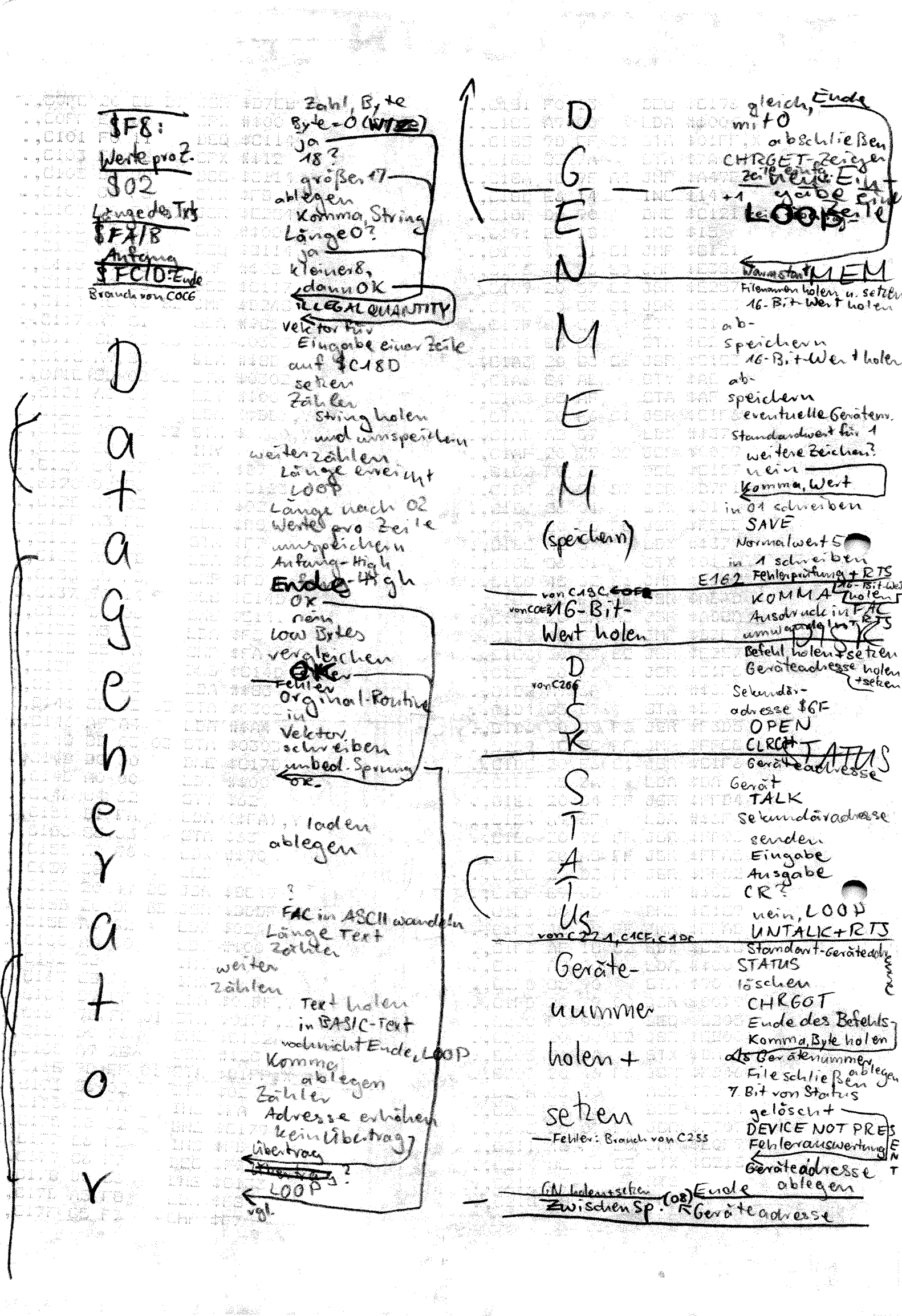
KEY

Byte-Wert
Null? ja! ILLEGAL QUANTITY
größer 9? ja, Fehler
Startwert: 255 = -1
Additionsvorbereiten
+11
herunterzählen bis 0
LOOP
Position ablegen
Komma, String
Länge > 11
ja, Fehler
Position
Länge ablegen
Low-Byte
Highbyte
ablegen
String kopieren
Zahl holen
ab-legen
Komma, Wert
ab-legen
Komma

guter Trick!

DOS

*-Routinen benutzt werden)


$F8:
Werte pro Z.
$02
Länge des Trg
$FA/B
Anfang
$FC/D: Ende
Branch von COC6
Datagenerator
Zahl, Byte
Byte=0 (WIZE)
ja
18?
größer 17
ablegen
Komma, String
Länge 0?
ja
kleiner 8,
dann OK
ILLEGAL QUANTITY
Vektor für
Eingabe einer Zeile
auf $C18D
setzen
Zähler
String holen
und umspeichern
weiterzählen
Länge erreicht
LOOP
Länge nach 02
Werte pro Zeile
umspeichern
Anfang-High
Ende-High
OK
nein
Low Bytes
vergleichen
OK
Fehler
Original-Routine
in
Vektor
schreiben
unbed. Sprung
OK
laden
ablegen
?
FAC in ASCII wandeln
Länge Text
zählen
weiter
zählen
Text holen
in BASIC-Text
wo nicht Ende, LOOP
Komma
ablegen
Zähler
Adresse erhöhen
kein Übertrag
Übertrag
?
LOOP
vgl.
DGENMEM
(speichern)
von C15C
von C0E8
16-Bit-
Wert holen
DISK
von C266
STATUS
von C221, C1CE, C1DC
Geräte-
nummer
holen +
setzen
—Fehler: Branch von C255
gleich, Ende
mit 0
abschließen
CHRGET-Zeigen
zeigt auf
nächste Eingabe
+1
LOOP
Warmstart
MEM
Filenamen holen u. setzen
16-Bit-Wert holen
ab-
speichern
16-Bit-Wert holen
ab-
speichern
eventuelle Geräten.
Standardwert für 1
weitere Zeichen?
nein
Komma, Wert
in 01 schreiben
SAVE
Normalwert 5
in 1 schreiben
E162 Fehlerprüfung + RTS
KOMMA
Ausdruck in FAC
umwandeln + RTS
Befehl holen + setzen
Geräteadresse holen
+ setzen
Sekundär-
adresse $6F
OPEN
CLRCH
Geräteadresse
Gerät
TALK
Sekundäradresse
senden
Eingabe
Ausgabe
CR?
nein, LOOP
UNTALK + RTS
Standart-Geräteadresse
STATUS
löschen
CHRGOT
Ende des Befehls
Komma, Byte holen
als Gerätenummer
ablegen
File schließen
7 Bit von Status
gelöscht
DEVICE NOT PRES
Fehlerauswertung
Geräteadresse
Geräteadresse ablegen
GM holen setzen
Zwischensp. (08) Ende
R Geräteadresse

DIR

D

I

R

1
8 (Default)
1
Filpcw
[Gerätadresse holen] setzen
Sekundäradresse
Null setzen (+ Flag f. LOAD)
BASIC-Ende-High
zwischenspeichern
Low-Byte

"
LOAD
Fehler
Zeiger

BASIC-Zeilen neu binden
LIST +RTS MERGE
LOAD/SAVE Parameter
BASIC-Ende Low holen

MERGE

minus 2
nach x
Highbyte
Übertrag
nach ?
Flag für LOAD
LOAD
zählen Y=0
Filenamen holen
= @ für Überschreiben
nein
weiter
zählen Y=1
2. Byte
":" ?
nein
= 0
"S" (Scratch)
in Filenamen
SCRATCH senden
Zeiger
+2
kein Übertrag
erhöhen
ablegen
Länge
minus 2
10 Zeichen
"SCRATCHING"
ausgeben
LOOP
Filenamen ausgeben
SAVE

SAVE

neue SAVE-Routine

SCRATCHING

TEXT

neue Tastatur

Tastaturdecodierung

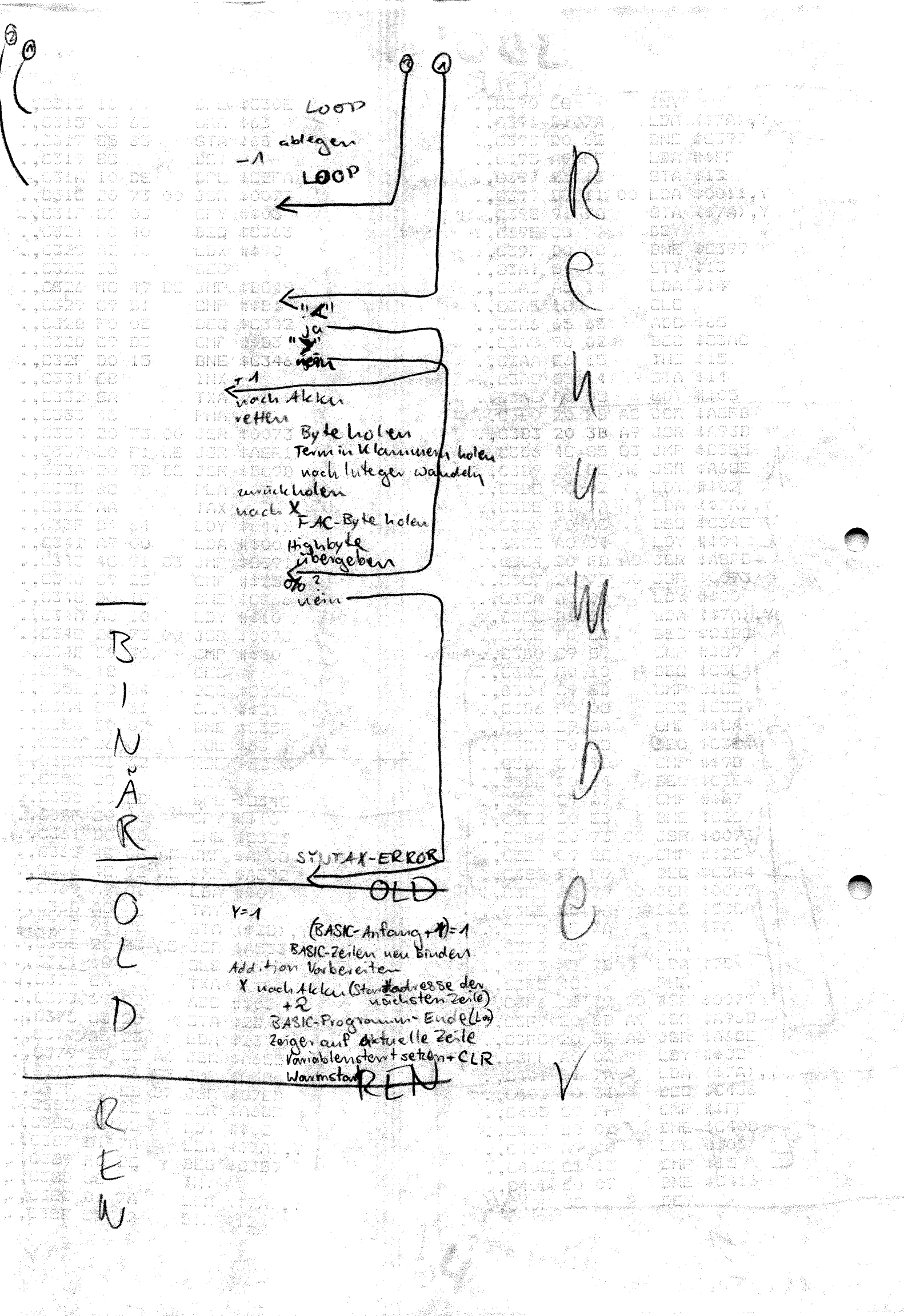
neue „Ausdruck-werten" numerisch ROUTINE
setzen
FAC
setzen
CHRGET
Orginalroutine
Dollar ?
nein
4 Zeichen
Zeichen holen
Ziffer
Buchstabe ?
nein
F
größer dann Fehler
minus 7
minus $2F
3
Links
links

-1

$

①

②

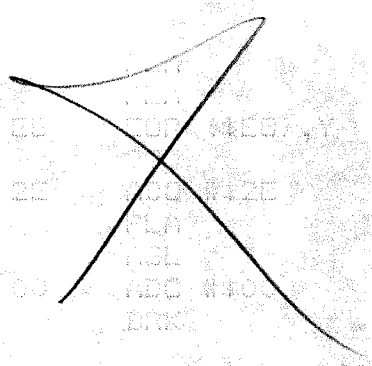

LOOP
ablegen
-1
LOOP
"("
"X"
nein
+1
nach Akku
retten
Byte holen
Term in Klammern holen
nach Integer wandeln
zurückholen
nach X
FAC-Byte holen
Highbyte
übergeben
%?
nein
BINÄR
SYNTAX-ERROR
OLD
Y=1
(BASIC-Anfang+1)=1
BASIC-Zeilen neu binden
Addition vorbereiten
X nach Akku (Startadresse der nächsten Zeile)
+2
BASIC-Programm-Ende (Low)
Zeiger auf aktuelle Zeile
Variablenstart setzen + CLR
Warmstart
REN
OLD
REN
RENUMBER

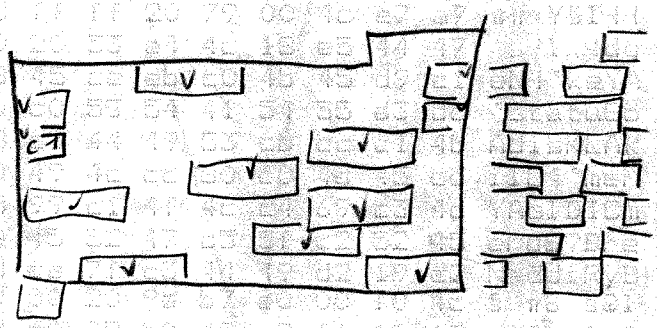
```
.,C410 B1 7A     LDA ($7A),Y
.,C412 D5 14     CMP $14
.,C414 F0 02     BEQ $C418
.,C416 A0 00     LDY #$00
.,C418 20 FD AD  JSR $ADFD
.,C41B 20 F2 A7  JSR $A7F2
.,C41E 4C FF C3  JMP $C3FF
.,C421 88        DEY
.,C422 84 D3     STY $D3
.,C424 20 CC E5  JSR $E5CC
.,C427 A0 05     LDY #$05
.,C429 B1 7A     LDA ($7A),Y
.,C42B AA        TAX
.,C42C C8        INY
.,C42D B1 7A     LDA ($7A),Y
.,C42F 20 CD BD  JSR $BDCD
.,C432 84 F7     STY $F7
.,C434 A0 00     LDY #$00
.,C436 68        PLA
.,C437 85 7B     STA $7B
.,C439 68        PLA
.,C43A 85 7A     STA $7A
.,C43C C0 02     CPY #$02
.,C43E F0 A4     BEQ $C3E4
.,C440 A5 15     LDA $15
.,C442 A6 14     LDX $14
.,C444 20 CD BD  JSR $BDCD
.,C447 84 F7     STY $F7
.,C449 A5 F9     LDA $F9
.,C44B 38        SEC
.,C44C E5 F7     SBC $F7
.,C44E F0 03     BEQ $C453
.,C450 85 F8     STA $F8
.,C452 08        PHP
.,C453 A5 7A     LDA $7A
.,C455 85 5F     STA $5F
.,C457 A5 7B     LDX $7B
.,C459 86 60     STX $60
.,C45B 38        SEC
.,C45C E5 FC     SBC $FC
.,C45E F0 01     BEQ $C461
.,C460 E8        INX
.,C461 86 14     STX $14
.,C463 84 15     STY $15
.,C465 A5 2D     LDA $2D
.,C467 85 FA     STA $FA
.,C469 A6 2E     LDX $2E
.,C46B 86 FB     STX $FB
.,C46D 18        CLC
.,C46E 65 F8     ADC $F8
.,C470 90 01     BCC $C473
.,C472 E8        INX
.,C473 85 2D     STA $2D
.,C475 86 2E     STX $2E
.,C477 A4 F7     LDY $F7
.,C479 88        DEY
.,C47A B1 5F     LDA ($5F),Y
.,C47C 91 FA     STA ($FA),Y
.,C47E 98        TYA
.,C47F D0 F8     BNE $C479
.,C481 C6 FB     DEC $FB
.,C483 C6 F9     DEC $F9
.,C485 A6 F9     LDX $F9
```

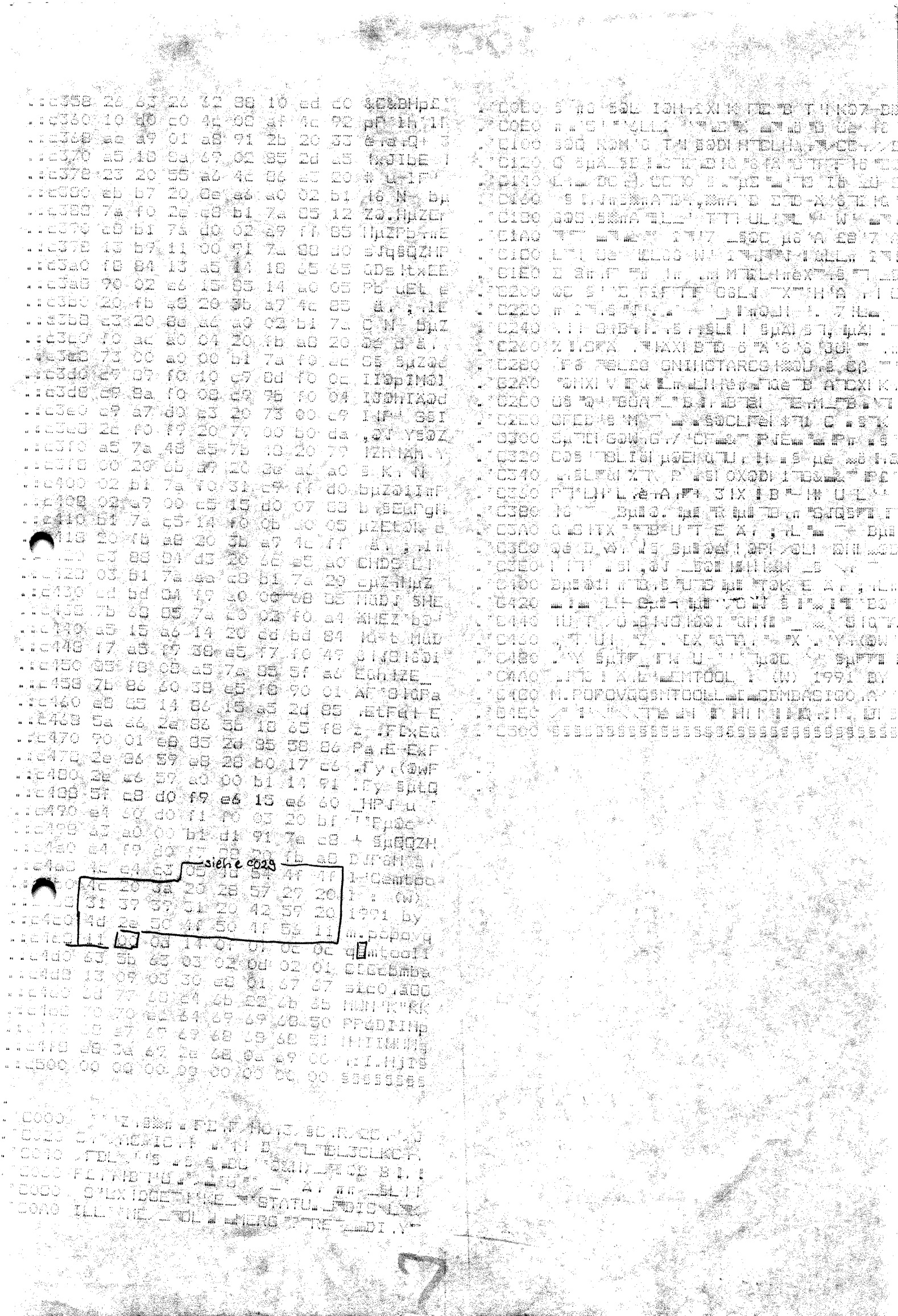
```
.,C1D5 B1 14     LDA ($14),Y
.,C1D7 91 5F     STA ($5F),Y
.,C1D9 C8        INY
.,C1DA D0 F9     BNE $C1D5
.,C1DC E6 15     INC $15
.,C1DE E6 60     INC $60
.,C1E0 E4 61     CPX $61
.,C1E2 D0 F1     BNE $C1D5
.,C1E4 F0 03     BEQ $C1E9
.,C1E6 4C 3F A3  JMP $A33F
.,C1E9 A0 00     LDY #$00
.,C1EB B1 D1     LDA ($D1),Y
.,C1ED 91 7A     STA ($7A),Y
.,C1EF C8        INY
.,C1F0 C4 F9     CPY $F9
.,C1F2 D0 F7     BNE $C1EB
.,C1F4 88        DEY
.,C1F5 20 FB A8  JSR $A8FB
.,C1F8 4C E4 C3  JMP $C3E4
```

```
.,C1FB 38        SEC
.,C1FC E9 04     SBC #$04
.,C1FE 48        PHA
.,C1FF A5 7A     LDA $7A
.,C201 38        SEC
.,C202 E9 04     SBC #$04
.,C204 85 7A     STA $7A
.,C206 B0 02     BCS $C20A
.,C208 C6 7B     DEC $7B
.,C20A A0 03     LDY #$03
.,C20C B1 7A     LDA ($7A),Y
.,C20E 91 FC     STA ($FC),Y
.,C210 88        DEY
.,C211 10 F9     BPL $C20C
.,C213 68        PLA
.,C214 85 FC     STA $FC
.,C216 A5 7B     LDA $7B
.,C218 E5 FD     SBC $FD
.,C21A 90 0A     BCC $C226
.,C21C A5 FD     LDA $FD
.,C21E 85 7B     STA $7B
.,C220 A5 FC     LDA $FC
.,C222 85 7A     STA $7A
.,C224 A0 00     LDY #$00
.,C226 A5 FC     LDA $FC
.,C228 C5 2F     CMP $2F
.,C22A A5 7B     LDA $7B
.,C22C E5 30     SBC $30
.,C22E B0 0E     BCS $C23E
.,C230 20 2D C6  JSR $C62D
.,C233 A5 FC     LDA $FC
.,C235 85 2F     STA $2F
.,C237 A5 7B     LDA $7B
.,C239 85 30     STA $30
.,C23B 84 FE     STY $FE
.,C23D B1 7A     LDA ($7A),Y
.,C23F 85 FD     STA $FD
.,C241 C8        INY
.,C242 B1 7A     LDA ($7A),Y
.,C244 85 FC     STA $FC
.,C246 C8        INY
.,C247 B1 7A     LDA ($7A),Y
.,C249 85 FB     STA $FB
.,C24B C8        INY
.,C24C B1 7A     LDA ($7A),Y
.,C24E 85 FA     STA $FA
.,C250 A5 7A     LDA $7A
.,C252 18        CLC
.,C253 69 05     ADC #$05
.,C255 85 F9     STA $F9
.,C257 A5 7B     LDA $7B
.,C259 69 00     ADC #$00
.,C25B 85 FA     STA $FA
.,C25D 38        SEC
.,C25E A5 FC     LDA $FC
.,C260 E5 7A     SBC $7A
.,C262 85 FC     STA $FC
.,C264 A5 FB     LDA $FB
.,C266 E5 7B     SBC $7B
.,C268 85 FB     STA $FB
.,C26A 38        SEC
.,C26B A5 F9     LDA $F9
.,C26D E5 7A     SBC $7A
.,C26F 85 F9     STA $F9
.,C271 A5 FA     LDA $FA
.,C273 E5 7B     SBC $7B
.,C275 85 FA     STA $FA
.,C277 A5 FE     LDA $FE
.,C279 F0 0A     BEQ $C285
.,C27B A0 00     LDY #$00
.,C27D B1 7A     LDA ($7A),Y
.,C27F 91 FD     STA ($FD),Y
.,C281 C8        INY
.,C282 D0 F9     BNE $C27D
.,C284 C8        INY
.,C285 B1 7A     LDA ($7A),Y
.,C287 91 FD     STA ($FD),Y
.,C289 C8        INY
.,C28A C0 04     CPY #$04
.,C28C D0 F7     BNE $C285
.,C28E 98        TYA
.,C28F 18        CLC
.,C290 65 7A     ADC $7A
.,C292 85 7A     STA $7A
.,C294 90 02     BCC $C298
.,C296 E6 7B     INC $7B
.,C298 A5 FD     LDA $FD
.,C29A 85 2F     STA $2F
.,C29C A5 FE     LDA $FE
.,C29E 85 30     STA $30
.,C2A0 60        RTS
```

```
.,C2A1 A0 00     LDY #$00
.,C2A3 B1 7A     LDA ($7A),Y
.,C2A5 91 FC     STA ($FC),Y
.,C2A7 C8        INY
.,C2A8 C4 F9     CPY $F9
.,C2AA D0 F7     BNE $C2A3
.,C2AC 88        DEY
.,C2AD B1 7A     LDA ($7A),Y
.,C2AF 91 FC     STA ($FC),Y
.,C2B1 C8        INY
.,C2B2 B1 7A     LDA ($7A),Y
.,C2B4 91 FC     STA ($FC),Y
.,C2B6 C8        INY
.,C2B7 B1 7A     LDA ($7A),Y
.,C2B9 91 FC     STA ($FC),Y
.,C2BB 98        TYA
.,C2BC 18        CLC
.,C2BD 65 FC     ADC $FC
.,C2BF 85 FC     STA $FC
.,C2C1 90 02     BCC $C2C5
.,C2C3 E6 FD     INC $FD
.,C2C5 A5 7A     LDA $7A
.,C2C7 C5 2F     CMP $2F
.,C2C9 A5 7B     LDA $7B
.,C2CB E5 30     SBC $30
.,C2CD B0 0E     BCS $C2DD
.,C2CF 20 2D C6  JSR $C62D
.,C2D2 A5 FC     LDA $FC
.,C2D4 85 2F     STA $2F
.,C2D6 A5 FD     LDA $FD
.,C2D8 85 30     STA $30
.,C2DA 84 FE     STY $FE
.,C2DC B1 7A     LDA ($7A),Y
.,C2DE 85 FD     STA $FD
.,C2E0 C8        INY
.,C2E1 B1 7A     LDA ($7A),Y
.,C2E3 85 FC     STA $FC
.,C2E5 C8        INY
.,C2E6 B1 7A     LDA ($7A),Y
.,C2E8 85 FB     STA $FB
.,C2EA C8        INY
.,C2EB B1 7A     LDA ($7A),Y
.,C2ED 85 FA     STA $FA
.,C2EF A5 7A     LDA $7A
.,C2F1 18        CLC
.,C2F2 69 05     ADC #$05
.,C2F4 85 F9     STA $F9
.,C2F6 A5 7B     LDA $7B
.,C2F8 69 00     ADC #$00
.,C2FA 85 FA     STA $FA
.,C2FC 38        SEC
.,C2FD A5 FC     LDA $FC
.,C2FF E5 7A     SBC $7A
.,C301 85 FC     STA $FC
.,C303 A5 FB     LDA $FB
.,C305 E5 7B     SBC $7B
.,C307 85 FB     STA $FB
.,C309 38        SEC
.,C30A A5 F9     LDA $F9
.,C30C E5 7A     SBC $7A
.,C30E 85 F9     STA $F9
.,C310 A5 FA     LDA $FA
.,C312 E5 7B     SBC $7B
.,C314 85 FA     STA $FA
.,C316 A5 FE     LDA $FE
.,C318 F0 0A     BEQ $C324
.,C31A A0 00     LDY #$00
.,C31C B1 7A     LDA ($7A),Y
.,C31E 91 FD     STA ($FD),Y
.,C320 C8        INY
.,C321 D0 F9     BNE $C31C
.,C323 C8        INY
.,C324 B1 7A     LDA ($7A),Y
.,C326 91 FD     STA ($FD),Y
.,C328 C8        INY
.,C329 C0 04     CPY #$04
.,C32B D0 F7     BNE $C324
.,C32D 98        TYA
.,C32E 18        CLC
.,C32F 65 7A     ADC $7A
.,C331 85 7A     STA $7A
.,C333 90 02     BCC $C337
.,C335 E6 7B     INC $7B
.,C337 A5 FD     LDA $FD
.,C339 85 2F     STA $2F
.,C33B A5 FE     LDA $FE
.,C33D 85 30     STA $30
.,C33F 60        RTS
```






siehe c029