

Ein besonderer Disassembler

Dieser Disassembler für den C64 arbeitet ausschließlich mit Dezimalzahlen

Bei vielen Hobby-Programmierern besteht der einzige Kontakt zur Maschinensprache im gelegentlichen Eingeben von Basic-Ladern. Da tut man sich insbesondere mit den Hexadezimalzahlen schon mal etwas schwer.

Hier ist nun ein Disassembler, der sowohl Adressen als auch Befehls- und Datenbytes in dezimaler Form ausgibt.

Das Programm ist sehr einfach anzuwenden und erklärt sich im übrigen von selbst, so daß eine weitergehende Beschreibung hier überflüssig ist.

Bei der Adressenabfrage gibt es drei mögliche Eingaben:

1. Die direkte Anfangsadresse (dezimal)
2. »W« für weiter
3. »E« für Ende

Bei der folgenden Frage nach der Ausgabe auf Drucker kann man wählen zwischen:

1. »J« für Ja = Druckausgabe
2. »N« für Nein (Return reicht)

Das Programm läuft auf dem C64, kann aber leicht auch für den VC20 umgeschrieben werden (in Zeile 10 werden nur die Bildschirmfarben eingestellt).

Die verwendeten Steuerzeichen stellen nur Farben dar und können folglich frei gewählt werden.

(Stefan Heine / ev)

```

1 REM DEZIMAL-DISASSEMBLER <065>
2 REM <145>
3 REM STEFAN HEINE <188>
4 REM WIESENREDDER 1 <069>
5 REM 2330 ECKERNFOERDE <201>
6 REM <149>
7 REM <150>
10 POKE 53280,0:POKE 53281,0 <047>
20 PRINT "ICLEAR,GREY 3,SPACE9)DISASSEMBLER
   [SPACE2]1.5" <049>
30 PRINT "[SPACE12]BY[SPACE]ST.HEINE" <212>
40 PRINT:PRINT:PRINT <103>
50 PRINT "[YELLOW]*[SPACE]ADRESSEN[SPACE]BITTE
   [SPACE]DEZIMAL[SPACE]" <170>
60 PRINT "[SPACE2]EINGEBEN!!!":PRINT <140>
70 PRINT "[GREEN]*[SPACE]ADRESSEN
   :[SPACE]W=WEITER[SPACE]E=ENDE":PRINT <207>
80 PRINT"*[SPACE]DRUCKER[SPACE]
   :[SPACE]'RETURN'=NEIN" <054>
90 PRINT:PRINT:PRINT <153>
100 REM COPYRIGHT * ST.HEINE * <062>
110 DIM M$(255),L$(255),F$(255),F2$(12) <087>
120 FOR I=0 TO 255:READ M$(I):NEXT
   :FOR I=0 TO 255:READ L$(I):NEXT
   :FOR I=0 TO 255:READ F$(I) <134>
130 NEXT:FOR I=0 TO 12:READ F1$(I),F2$(I):NEXT
   :PC=0 <180>
140 IF IC=1 THEN GOSUB 360:IC=0 <251>

```

```

150 INPUT "[LIG.BLUE]#[SPACE]ANFANGSADRESSE";A#
   <193>
160 INPUT "[GREY 3]#[SPACE]AUSGABE[SPACE]AUF
   [SPACE]DRUCKER";GC# <244>
170 HC#=LEFT$(GC#,1) <047>
180 IF HC#="J"THEN IC=1 <068>
190 J=LEN(A#):IF J>5 THEN 140 <100>
200 IF A#="W"THEN 230 <065>
210 IF A#="E"THEN 900 <061>
220 PC=VAL(A#) <156>
230 FOR I=1 TO 24:DZ=PC:PRINT "[GREEN]";
   :P1=PEEK(PC) <061>
240 ON L$(P1)GOSUB 260,270,330 <112>
250 NEXT:GOTO 140 <212>
260 PRINT TAB(1)"[RED]";PC;:PRINT TAB(9)"[GREEN]
   ";P1;:PRINT "[BLUE]";TAB(24);M$(P1):PC=PC+1
   :RETURN <011>
270 PRINT TAB(1)"[RED]";PC;:PRINT TAB(9)"[GREEN]
   ";P1;:P2=PEEK(PC+1):PRINT TAB(13)P2; <238>
280 PRINT "[BLUE]";TAB(24);M$(P1);"[SPACE],
   LIG.BLUE]"; <025>
290 IF F$(P1)=9 THEN P2=PEEK(PC+1)
   :IF P2<128 THEN P2=PC+P2+2:GOTO 310 <148>
300 IF F$(P1)=9 THEN P2=PC-(255-P2)+1 <238>
310 PRINT F1$(F$(P1));P2;:DZ=P2:GOTO 320 <085>
320 PRINT F2$(F$(P1)):PC=PC+2:RETURN <186>
330 PRINT TAB(1)"[RED]";PC;:PRINT TAB(9)"[GREEN]
   ";P1;:P2=PEEK(PC+1):PRINT TAB(14)P2; <043>
340 P3=PEEK(PC+2):PRINT TAB(19)P3;
   :PRINT "[BLUE]";TAB(24);M$(P1)"[LIG.BLUE]";
   :P4=(P3*256)+P2 <198>
350 PRINT P4;F1$(F$(P1));:DZ=P3*256+P2
   :PRINT F2$(F$(P1)):PC=PC+3:RETURN <081>
360 OPEN 4,4:PRINT#4,CHR$(15) <076>
370 VB=1024:VC=55296 <020>
380 FOR I=VB TO VB+999 STEP 40:P#="" <211>
390 FOR J=0 TO 39:X=PEEK(I+J) <216>
400 IF X<32 THEN X=X+64:GOTO 440 <004>
410 IF X<64 THEN 440 <218>
420 IF X<96 THEN X=X+32:GOTO 440 <029>
430 IF X<128 THEN X=X+64:GOTO 440 <088>
440 P#=P#+CHR$(X) <109>
450 NEXT:PRINT#4,P#;NEXT:CLOSE 4:RETURN <125>
460 REM*** DATAS BEFEHLSLISTE *** <063>
470 DATA BRK,ORA,???,???,???,ORA,ASL,???,PHP,
   ORA,ASL,???,???,ORA,ASL,???, <008>
480 DATA BPL,ORA,???,???,???,ORA,ASL,???,CLC,
   ORA,???,???,???,ORA,ASL,???, <216>
490 DATA JSR,AND,???,???,BIT,AND,ROL,???,PLP,
   AND,ROL,???,BIT,AND,ROL,???, <095>
500 DATA BMI,AND,???,???,???,AND,ROL,???,SEC,
   AND,???,???,???,AND,ROL,???, <205>
510 DATA RTI,EOR,???,???,???,EOR,LSR,???,PHA,
   EOR,LSR,???,JMP,EOR,LSR,???, <158>
520 DATA BVC,EOR,???,???,???,EOR,LSR,???,CLI,
   EOR,???,???,???,EOR,LSR,???, <054>
530 DATA RTS,ADC,???,???,???,ADC,ROR,???,PLA,
   ADC,ROR,???,JMP,ADC,ROR,???, <079>
540 DATA BVS,ADC,???,???,???,ADC,ROR,???,SEI,
   ADC,???,???,???,ADC,ROR,???, <239>
550 DATA???,STA,???,???,STY,STA,STX,???,DEY,???,
   TXA,???,STY,STA,STX,???, <239>
560 DATA BCC,STA,???,???,STY,STA,STX,???,TYA,
   STA,TXS,???,???,STA,???,???, <200>
570 DATA LDY,LDA,LDX,???,LDY,LDA,LDX,???,TAY,
   LDA,TAX,???,LDY,LDA,LDX,???, <217>
580 DATA BCS,LDA,???,???,LDY,LDA,LDX,???,CLV,
   LDA,TSX,???,LDY,LDA,LDX,???, <176>
590 DATA CPY,CMP,???,???,CPY,CMP,DEC,???,INY,
   CMP,DEX,???,CPY,CMP,DEC,???, <197>
600 DATA BNE,CMP,???,???,???,CMP,DEC,???,CLD,
   CMP,???,???,???,CMP,DEC,???, <025>
610 DATA CPX,SBC,???,???,CPX,SBC,INC,???,INX,
   SBC,NOP,???,CPX,SBC,INC,???, <221>
620 DATA BEQ,SBC,???,???,???,SBC,INC,???,SED,
   SBC,???,???,???,SBC,INC,???, <053>
630 REM*** DATAS 1,2 ODER 3 BIT *** <059>
640 DATA 1,2,1,1,1,2,2,1,1,2,1,1,1,3,3,1,2,2,1,
   1,1,2,2,1,1,3,1,1,1,3,3,1 <139>
650 DATA 3,2,1,1,2,2,1,1,2,1,1,3,3,3,1,2,2,1,
   1,1,2,2,1,1,3,1,1,1,3,3,1 <154>
660 DATA 1,2,1,1,1,2,2,1,1,2,1,1,3,3,3,1,2,2,1,
   1,1,2,2,1,1,3,1,1,1,3,3,1 <161>
670 DATA 1,2,1,1,1,2,2,1,1,2,1,1,3,3,3,1,2,2,1,

```

