

```

1900 GOTO 40
2000 ZL=ZL+(ZL=1)-(ZL=2):T$(55,55)=S
    TR$(ZL):GOSUB 58
2100 GOTO 100
2500 QL=QL+(QL=1)-(QL=2):T$(15,15)=S
    TR$(QL):GOSUB 58
2600 GOTO 100
6000 ? CHR$(125):POSITION 3,2: ? {CO
    DE170/160/170/160/170/160/170/1
    60/160/211/208/197/218/201/193/
    204/173/198/201/204/197/173/203
    /207/208/201/197/160/160/170/16
    0/170/160/170/160/170}
6005 GOSUB 60
6010 TRAP 6010:POSITION 9,4: ? "FILEN
    UMMER: ";INPUT E2;FIL
6105 GOSUB 61
6020 IF DO=1040 THEN GOSUB 52
6030 T=1:GES=0
6040 POSITION 13,6: ? "SEKTOR: 4"
6050 POSITION 13,8: ? "BUFFER: 0"
6100 FOR SEK=4 TO DO
6103 POSITION 16,8: ? SEK
6105 AR=ADR(SE$)+(T-1)*128
6110 X=USR(ADR(S$),QL,SEK,AR,82)
6120 IF SE$(T-1)*128+1,T*128)=LEER$
    THEN G200
6122 IF SE$(T*128,T*128)="[CODE0]" T
    HEN G200
6123 SE=ADR(SE$(T*128,T*128-2))
6124 FN=INT(PEEK(SE)/4)
6125 IF FN<FIL THEN G200
6127 POKE SE,PEEK(SE)-FIL*4
6130 SE=T:SEK=T
6135 POSITION 16,10: ? T
6140 T=T+1:GES=GES+1:IF T=181 THEN G
    OSUB WRITE:GOSUB QUELL
6200 NEXT SEK
6205 GOSUB WRITE
6260 P$(1)="{CODE0}":P$(128)=P$:P$(2
    )=P$
6270 P$(1,1)="{B}":P$(3,3)=CHR$(INT(GE
    S/256)):P$(2,2)=CHR$(GES-(INT(G
    ES/256)*256))
6280 P$(5,5)=CHR$(INT(SE(1)/256)):P$(
    4,4)=CHR$(SE(1)-(INT(SE(1)/256
    )*256))
6290 P$(6,6)="{SPEZIALFILE}"
6300 X=USR(ADR(S$),ZL,361,ADR(P$),87
    )
6310 POSITION 4,16: ? "FUEHREN SIE JE
    Tzt BITTE NOCH"
6320 POSITION 4,18: ? "EINEN VTOC-CHE
    CK DURCH . . .";
6330 FOR G=1 TO 500:NEXT G
6340 GOTO 40
6500 DO=DO+320*(DO=720)-320*(DO=1040
    )
6510 IF DO=1040 THEN T$(154,160)="2.
    0-[CODE178/174/181]":GOTO 6530
6520 T$(154,160)="[CODE178/174/176]-
    2.5"
6530 GOSUB 58
6600 GOTO 100
11000 IF VE=1 THEN POKE 1913,80:T$(1
    14,119)="{AN-[CODE193/213/211]}
    :GOSUB 58:VE=0:GOTO 11100
11010 T$(114,119)="{CODE193/206}-AUS
    "
11050 GOSUB 58:POKE 1913,87:VE=1
11100 GOTO 100
12000 POSITION 7,14: ? "[CODE160/160/
    170/160/170/160/170/160/160/19
    6/201/197/204/196/201/211/203/
    197/212/12/197/160/197/201/20
    6/204/197/199/197/205/160/160/
    170/160/170/160/170/160/170/17
    0/160/160]"
12010 POSITION 7,16: ? "----[2SPACES]
    D I S K . S Y S [2SPACES]-----"
12020 POSITION 7,18: ? "[7SPACES]EINL
    EGEN,DANN"
12030 POSITION 7,20: ? "[7SPACES]{COD
    E160/210/160/197/160/212/160/2
    13/160/210/160/206/160}"
12040 GET E1,A:IF A<155 THEN 40
12050 TRAP 40:RUN "D:DISK.SYS"
12499 STOP
15000 POSITION 3,20: ? "{CODE253/253/
    170/160/170/160/170/160/160/21
    8/201/197/204/196/201/211/203/
    197/212/12/197/160/197/201/20
    6/204/197/199/197/205/160/160/
    170/160/170/160/170/160/170/17
    0/160/160}"
15010 POSITION 15,22: ? "[CODE210/197
    /212/213/210/206] DRUECKEN";
15020 GET E1,A
15030 IF A<155 THEN 15020
15035 POSITION 3,20: ? "[CODE156/156/
    156]"
15040 POSITION 13,10: ? "SEKTOR: "
15050 POSITION 13,12: ? "BUFFER: "
15060 TRAP 40000:TRAP 40
15100 FOR W=1 TO T-1
15110 SE=SE(W)
15113 POSITION 21,10: ? SK
15116 POSITION 21,12: ? W
15120 AR=ADR(SE$)+(W-1)*128
15130 X=USR(ADR(S$),ZL,SK,AR,87)
15140 NEXT W
15150 T=1
15200 RETURN
18000 POSITION 3,20: ? "{CODE253/253/
    170/160/170/160/170/160/160/20
    6/213/197/204/204/196/201/211/
    203/197/212/12/197/160/197/20
    1/206/204/197/199/197/206/160/
    160/170/160/170/160/170}"
18010 POSITION 15,22: ? "[CODE210/197
    /212/213/210/206] DRUECKEN";
18020 GET E1,A
18030 IF A<155 THEN 18020
18035 POSITION 3,20: ? "[CODE156/156/
    156]"
18040 RETURN
29000 POKE 709,15:POKE 710,48
29010 RESTORE 29030
29015 AD=155*256
29020 FOR I=0 TO 9:READ A:POKE AD+I,
    A:NEXT I
29030 DATA 0,0,66,12,155,2,2,2,1
29040 DL=PEEK(560)+PEEK(561)*256:POK
    E DL,1:POKE DL+1,0:POKE DL+2,1
    55
29050 POKE AD+10,PEEK(560)+3:POKE AD
    +11,PEEK(561)
29060 RETURN
30000 T$(1,93)="{QUELLAUFWERK: 1[25SP
    ACES]ZIELAUFWERK: 1[38SPACES]}
    "
30001 T$(94,179)="{[2SPACES]}VERIFY:
    AN-[CODE193/213/211]{25SPACES}
    DOS-ART: {CODE178/174/176}-2.5
    [2SPACES]}"
30002 T$(180,200)="{[2SPACES]}"
30020 S$="hhh[CODE141/1/3]h[CODE141/
    1/3]h[CODE141/10/3]h[CODE141/
    5/3]h[CODE141/4/3]hh[CODE141/2
    /3] S[CODE228/173/3/3/133/212/
    169/0/133/213/96]"
30045 TOZ$="hh[CODE133/204]h[CODE133
    /203]hh[CODE133/205]hh[CODE133
    /207]h[CODE133/208]h[CODE207/
    13/207/160/0/177/203]8[CODE23
    3] [CODE145/207/200/196/205/20
    8/244/96]"
30090 POKE 741,255:POKE 742,154
30100 RETURN
READY I

```

Zeilen-Assembler mit Schnelleinstieg

Rechner: Atari XL/XE. Programmname: ATADIS. Programmlänge: ca. 9000 Bytes. Programmiersprache: BASIC und MCode.

ATADIS ist ein in BASIC geschriebener Assembler / Disassembler. Er kann und soll kein Werkzeug zur Erstellung längerer Maschinenprogramme sein und soll keinen herkömmlichen Assembler / Disassembler ersetzen. Er leistet jedoch gute Dienste bei Aufgaben, bei denen die gewohnte Programmierungsumgebung BASIC nicht verlassen werden soll. Mögliche Anwendungen erstrecken sich vom Erstellen und Debuggen kleinerer Maschinenprogramme bis zum Ein-

deutschen amerikanischer Maschinenprogramme. Aufgrund seiner einfachen Bedienung eignet sich ATADIS auch gut für Anfänger im Arbeiten mit Maschinensprache, die die Ausgabe für einen "richtigen" Assembler noch scheuen.

Das Programm bietet folgende Optionen:

M - Memory Dump :

Speicherinhalte werden sowohl hexadezimal als auch als A(TA)SCII-Zeichen zu jeweils 8 Byte/Zeile ausgegeben. Die Ausgabe erfolgt wahlweise auf dem Bildschirm oder dem Drucker. In letzterem Fall

werden A(TA)SCII-Zeichen, die der Drucker als Steuerzeichen interpretieren könnte, vorher abgefangen.

B - Binary Dump :

Es werden Speicherinhalte als Binärzahlen dargestellt. Dies dient hauptsächlich dem Begutachten von Charactersets oder PM-Daten. Auch hier erfolgt die Ausgabe wahlweise auf Bildschirm oder Drucker.

D - Disassemble :

Disassemblieren eines Speicherbereichs.

Bei relativen Sprüngen wird die Zieladresse direkt angegeben, die Ausgabe erfolgt wahlweise auf Drucker oder Bildschirm. Falls Werte nicht disassemblierbar sind, werden die entsprechenden Zeichen gefolgt von "???" ausgegeben.

W - Write Memory :

Die Speicherinhalte werden einzeln angezeigt und können mit dem neuen Wert überschrieben werden. Die Eingabe von RETURN beläßt die alten Werte in den Speicherzellen.

T - Text Write :

Die A(TA)SCII-Werte eines von der Tastatur eingegebenen Textes werden im Speicher abgelegt. Korrekturmöglichkeiten während der Eingabe sind durch DELETE BACK gegeben. Die Eingabe kann wahlweise durch ESCAPE oder RETURN abgeschlossen werden.

A - Assemble :

Aufruf des zeilenorientierten Assemblers.

Die Syntax entspricht der üblichen, auch in anderen für Atari erhältlichen Assemblern (Assembler/Editor, MAC 65 etc.) verwendeten. An Zahlenformaten stehen dezimal, hexadezimal (mit vorangestelltem "\$") und binär (mit vorangestelltem "%") zur Verfügung.

Da die einzelnen Zeilen direkt nach der Eingabe assembliert werden, ist die Verwendung von symbolischen

Adressen (Labels) leider nicht möglich. Dafür sieht man aber direkt nach der Eingabe das Ergebnis des Assembliervorgangs. Der Assembliervorgang wird durch die Eingabe von END abgeschlossen.

L - Load :

Der verfügbare freie Speicherbereich wird angezeigt.

Anschließend wird der Benutzer nach der Startadresse für das einzuladende Programm und die Filespezifikationen (zum Beispiel D1:FILENAME.EXT oder C:) gefragt.

Es werden Files mit beliebigem Format (BASIC-Programme, Maschinenprogramme, Dateien, ...) geladen.

S - Save :

Der Benutzer wird nach der Startadresse des Programms und der Filespezifikation (zum Beispiel D1:FILENAME.EXT) gefragt.

Bei Kassettenoperationen wird automatisch im schnelleren Format (wie bei CSAVE/CLOAD) aufgezeichnet beziehungsweise geladen.

- Dez/Hex-Wandlung :

Umwandlung einer Dezimalzahl (maximal 65535) in eine Hexadezimalzahl.

- Hex/Dez-Wandlung :

Umwandlung einer Hexadezimalzahl (maximal \$FFFF) in einer Dezimalzahl.

- Printer ON / OFF :

Umschaltung zwischen Drucker- oder Bildschirmausgabe. Ist kein Drucker angeschlossen, wird automatisch auf dem Bildschirm ausgegeben.

SHIFT-CLEAR :

Bildschirm löschen.

! - Exit :

Rückkehr zum Atari-BASIC.


```

6105 IF OP$(D,D)="$(" THEN X=N2:D=D+N
6110 IF OP$(D,D)="$(" THEN H=N1:D=D+N
6120 IF OP$(D,D)="$(" THEN Q=N1:D=D+N
6130 OP$=OP$(D):D=LEN(OP$):IF D<N3 T
HEN 6160
6140 IF OP$(D-N2,D)="$(" THEN Y=N1:
D=D-N3
6145 IF OP$(D-N2,D)="$(" THEN Y=N2:
D=D-N3
6150 IF OP$(D-N1,D)="$(" THEN Y=N3:D
=D-N2
6155 IF OP$(D-N1,D)="$(" THEN Y=N4:D
=D-N2
6160 IF OP$(D,D)="$(" THEN Y=N5:D=D-N
1
6170 OP$=OP$(N1,D)
6180 IF H THEN HEX$=OP$:GOSUB HEXDEZ
:DEZ=N:GOTO 6200
6190 IF Q THEN BIN$=OP$:GOSUB BINDEZ
:GOTO 6200
6195 DEZ=VAL(OP$)
6200 L=N2+(DEZ)=PAGE) OR (LEN(HEX$)
=N4)
6210 K=C*1000+L*100+X*N10+Y:FOR I=N0
TO PAGE-N1:IF A(I)=K THEN B=I:
I=PAGE:NEXT I:GOTO 6300
6220 NEXT I
6230 ? " [CODE253/28/211/249/238/244/
225/248/160/197/242/242/239/242
/161]":FOR I=N0 TO EOL:NEXT I:
? " [CODE156]":GOTO 6005
6300 POKE AD,B:IF L=N2 THEN POKE AD+
N1,DEZ
6310 IF L=3 THEN POKE AD+N1,DEZ=INT(
DEZ/PAGE)*PAGE+POKE AD+N2,INT(D
EZ/PAGE)
6315 ? " [CODE28/156]":J=L:DEZ=AD:L=
N4:GOSUB DEZHEX: ? HEX$:" [2SPACE
S]":
6320 FOR I=N0 TO J-N1:DEZ=PEEK(AD+I)
:L=N2:GOSUB DEZHEX: ? HEX$:" [2
NEXT ?
6330 AD=AD+J:POKE 85,18: ? EDIT$:GOT
O 6005
6995 REM
6996 REM [CODE145/146/146/146/146/14
6/146/133]
6997 REM [CODE252/160/204/239/225/22
8/160/252]
6998 REM [CODE154/146/146/146/146/14
6/146/131]
6999 REM
7000 ? " ? Free Memory: $":DEZ=PEEK
(144):PAGE*PEEK(145):L=N4:GOSUB
DEZHEX: ? HEX$: - $":
7010 DEZ=PEEK(741):PAGE*PEEK(742):L=
N4:GOSUB DEZHEX: ? HEX$
7020 ? " Bufferaddress ":INPUT HEX$:I
F LEN(HEX$)<N4 THEN ? " ? Inval
id Input.":GOTO PROMPT
7030 GOSUB HEXDEZ:BH=INT(N/PAGE):BL=
N-BH*PAGE
7040 ? "Enter Filespec ":INPUT SPEC
$:TRAP PROMPT:GS=N0:IF SPEC$(1,
1)="$(" THEN GS=PAGE/N2
7050 OPEN EN1,N8,GS,SPEC$:POKE IOBAS
E+N2,N7:POKE IOBASE+N8,N0:POKE
IOBASE+N9,PEEK(742)-PEEK(145)
7060 POKE IOBASE+N4,BL:POKE IOBASE+N
5,BH:I=USR(ADR(ML$)):CLOSE EN1
7070 SL=PEEK(IOBASE+N8):SH=PEEK(IOBA
SE+N9):DEZ=SL+PAGE*SH:L=4:GOSUB
DEZHEX: ? Bytes loaded: $":HEX
$:GOTO PROMPT
7995 REM
7996 REM [CODE145/146/146/146/146/14
6/146/133]
7997 REM [CODE252/160/211/225/246/22
9/160/252]
7998 REM [CODE154/146/146/146/146/14
6/146/131]
7999 REM
8000 ? "Startaddress ":INPUT HEX$:IF
LEN(HEX$)<N4 THEN ? " ? Inval
id Input.":GOTO PROMPT
8010 GOSUB HEXDEZ:BH=INT(N/PAGE):BL=
N-BH*PAGE
8020 ? "Filelength ":INPUT HEX$:IF
LEN(HEX$)<N4 THEN ? " ? Invali
d Input.":GOTO PROMPT
8030 GOSUB HEXDEZ:SH=INT(N/PAGE):SL=
N-SH*PAGE
8040 ? "Enter Filespec ":INPUT SPEC
$:TRAP PROMPT:GS=N0:IF SPEC$(1,
1)="$(" THEN GS=PAGE/N2
8050 OPEN EN1,N8,GS,SPEC$:POKE IOBAS
E+N2,N10+N1:POKE IOBASE+N8,SL:P
OKE IOBASE+N9,SH
8060 POKE IOBASE+N4,BL:POKE IOBASE+N
5,BH:I=USR(ADR(ML$)):CLOSE EN1
8070 PROMPT
8995 REM
8996 REM [CODE145/146/146/146/146/14
6/146/146/146/146/133]
8997 REM [CODE252/160/208/242/233/23
8/244/229/242/160/252]
8998 REM [CODE154/146/146/146/146/14
6/146/146/146/146/131]
8999 REM
9000 IF V=N3 THEN Y=N4: ? "PRINTER ON
"
9010 IF V=N4 THEN Y=N3: ? "PRINTER OF
F.":GOTO PROMPT
9995 REM
9996 REM [CODE145/146/146/146/146/14
6/146/146/146/146/146/146/146/1
33]
9997 REM [CODE252/160/196/229/250/16
0/173/190/160/200/229/248/160/2
52]
9998 REM [CODE154/146/146/146/146/14
6/146/146/146/146/146/146/146/1
31]
9999 REM
10000 TRAP PROMPT: ? "Decimal ":INPU
T DEZ:L=N4:GOSUB DEZHEX: ? Hex
": ? HEX$:GOTO PROMPT
10995 REM
10996 REM [CODE145/146/146/146/146/14
6/146/146/146/146/146/146/146/1
33]
10997 REM [CODE252/160/196/229/250/1
60/188/173/160/200/229/248/160
/252]
10998 REM [CODE154/146/146/146/146/14
6/146/146/146/146/146/146/146/1
31]
10999 REM
11000 ? "Hex ":INPUT HEX$:GOSUB HEX
DEZ:IF EF=N1 THEN GOTO PROMPT
11010 ? "Decimal ":N:GOTO PROMPT
29995 REM
29996 REM [CODE145/146/146/146/146/14
6/146/146/146/146/146/146/146/1
31]
29997 REM [CODE252/160/201/238/233/2
44/233/225/236/233/243/233/229
/242/245/238/231/160/252]
29998 REM [CODE154/146/146/146/146/14
6/146/146/146/146/146/146/146/1
31]
29999 REM
30000 RESTORE 30000:POKE 559,0
30010 READ N0,N1,N2,N3,N4,N5,N6,N7,N
8,N9,N10,N12,ESC,ADRES,BOL,HE
XDEZ,PAGE,DEZHEX,GETKEY,DEZBIN
,BINDEZ,BREAK,KEY,PROMPT
30020 DATA 0,1,2,3,4,5,6,7,8,9,10,12
,27,100,155,200,256,300,400,50
0,600,700,764,800
30030 DIM ML$(N6),SPEC$(N10+N5):ML$=
" [CODE162/161/LV(CODE228)]:IOB
ASE=848:V=N3
30100 DIM $$(171),A(PAGE-N1),L$(N12)
,T$(N10+N8),HEX$(N4),NIBBLE$(N
1),BNS$(N8),BRACK$(N10+N1),EDI
T$(N10+N2),OPS(N12)
N$(N1,99)="ACANDASLBCBCCSBEQB
ITMBIBNEBPLBRKBVCBSCCLDCLC
LVCMPXPCPYDCDCDEYSEORINCINXI
NYJMPJSRLDADLDXLYLSR"
N$(100,171)="NOPORAPHAPPLPLPL
PROLRTIRTSBSCSECSSEISTASTXST
YTAXTATXTXTATXTATYAROR???"
L$=(CODE31/30)E(CODE30/31){(C
ODE31/30) A: ? $=" [CODE31/30
1,X)],Y,X,Y } [2SPACES]:"BREAK
$=" ? BREAK " ? FOR I=N0 TO PA
GE-N1:READ A(I):A=NEXT I:RET
URN
30150 OPEN EN2,N12,N0,"E":OPEN EN3,
N8,N0,"E":TRAP 30160:OPEN EN4
,N8,N0,"P":GOTO 30170
30160 CLOSE EN4:OPEN EN4,N8,N0,"E":
30170 GRAPHICS N0:SETCOLOR N1,N0,N10
:SETCOLOR N2,N12+N2,N2:SETCLO
R N4,N12+N2,N2:POKE 82,N12
30180 ? " [CODE125/8/160/193/212/193/
196/201/211/160/177/174/183/16
0/101]": ? " [CODE160/195/239/160
/249/242/233/231/232/244/160/2
26/249/160]": ? " [CODE138/160/1
60/212/200/207/210/211/207/198
/212/160/160/136/253]":POKE 82
,N0:GOTO PROMPT
30200 DATA 11100,35221,0,0,0,35200,3
200,0,37100,35210,3130,0,0,353
0,0,3300,0,10200,35222,0,0,0,35
203,3203,0,14100
30210 DATA 35304,0,0,0,35303,3303,0,
29300,2221,0,0,7200,2200,40200
,0,39100,2210,40130,0,7300,230
0,40300,0,8200
30220 DATA 2222,0,0,0,2203,40203,0,4
4100,2304,0,0,0,2303,40303,0,4
1100,24221,0,0,0,24200,33200,0
,36100,24210
30230 DATA 33130,0,28300,24300,33300
,0,12200,24222,0,0,0,24203,332
03,0,16100,24304,0,0,0,24303,3
3303,0,42100,1221
30240 DATA 0,0,0,1200,56200,0,38100,
1210,56130,0,28325,1300,56300,
0,13200,1222,0,0,0,1203,56203,
0,46100,1304
30250 DATA 0,0,0,1303,56303,0,0,4722
1,0,0,49200,47200,48200,0,2310
0,0,53100,0,49300,47300,48300,
0,4200,47222,0,0
30260 DATA 49203,47203,48204,0,55100
,47304,54100,0,0,0,47303,0,0,322
10,30221,31100,0,32200,30200,3
1200,0,51200,30210
30270 DATA 50100,0,32300,30300,31300
,0,5200,30222,0,0,32203,30203,
31204,0,0,17100,30304,52100,0,32
303,30303,31304
30280 DATA 0,20210,18221,0,0,20200,1
8200,21200,0,27100,18210,22100
,0,20300,18300,21300,0,9200,18
222,0,0,0,18203
30290 DATA 21203,0,15100,18304,0,0,0
,18303,21303,0,19210,43221,0,0
,19200,43200,25200,0,26100
30300 DATA 43210,34100,0,19300,43300
,25300,0,6200,43222,0,0,0,4320
3,25203,0,45100,43304,0,0,0,43
303,25303,0

```

* BEIM ABTIPPEN ARBEIT SPAREN *
 * können Sie sich, wenn Sie mit *
 * der Anforderungskarte am Heft- *
 * ende die Programme auf Daten- *
 * träger anfordern. *
 * Anwenden und auf Ihre Belange *
 * abändern lassen sich diese *
 * Programme allerdings nur mit *
 * Hilfe dieser CHIP-SPECIAL- *
 * Ausgabe. *