

```

1 ' DIRECT ACCESS ROUTINES
2 ' By Petr Lukes
3 ' The Amstrad User, Oct. '89
4 '
10 :ns="DirAcc LKS 871007/880428 : Disc Direct Access
Routines for AMSTRAD CPCs."+CHR$(10)+CHR$(13)+"P. Lukes
, 26 Noll St., Toowoomba, 4350."
20 ' Initialise
30 INK 0,26:INK 1,0:PAPER 0:PEN 1:MODE 2:PRINT ns
40 PRINT"Caution : This programme does not respect any
file structure"
50 PRINT"and will write to any part of the disc!"
60 WINDOW#1,16,79,17,25:buffer display
70 WINDOW#2,1,15,17,25:PRINT#2,CHR$(24)'record informat
ion in inverse
80 WINDOW#3,1,60,1,16:dialogue
90 blocksize1=512:MEMORY 42619-blocksize1'reserve block
buffer
100 recsize1=128:recblock1=blocksize1\recsize1
110 IF blocksize1 MOD recsize1(>0 THEN PRINT"Non-integr
al number of records per block"
120 sec11=&C1'first sector number for DATA=&C1, SYSTEM=
&41, IBM=1
130 'sec11=&F1'first sector number for special FILE for
mat created by FORMATS utility, TAU Feb 88
140 sectr1=9:IF sec11=1 THEN sectr1=8'9 sectors per tra
ck for DATA, SYSTEM and FILE, 8 for IBM
150 tracks1=40:blocks1=sectr1*tracks1'total blocks
160 recmax1=blocks1+recblock1-1'highest numbered record
170 fill11=&E5'filler byte for AMSDOS formats
180 'fill11=&43'filler byte for special FILE format
190 ruls1=STRING$(recsize1,fill11)'record indicator
200 PRINT" Set up for"recmax1+1"records of"recsize1"byt
es each,";
210 PRINT recblock1"records per block,"
220 PRINT tracks1"tracks,"blocks1"blocks, format ";
230 IF sec11=1 THEN PRINT"IBM"ELSE IF sec11=&41 THEN PR
INT"SYSTEM"ELSE IF sec11=&C1 THEN PRINT"DATA"ELSE IF se
c11=&F1 THEN PRINT"FILE"ELSE PRINT"unknown"
240 buf1=UNT(HIMEM+1)'start of buffer
250 secr1=4+128:secw1=5+128'Command names: read/write s
ector. Locations are used to store addresses after they
are found by efcom
260 DIM asec1(12),faradd1(1)'Arrays for Sec_R/W, 'far
address' for RST 3
270 ' efcom : find command, save address and ROM select

```

```

290 DATA 00,21,FF,FF,e5,c0,d4,bc,d1,d0,eb,73,23,72,79,3
2
290 DATA FF,FF,c9,00
300 c=0:FOR a=0 TO 9:READ x$,y$:mfcom$(a)=VAL("%"+y$+x$
):c=mfcom$(a):NEXT a
310 IF c<>3030 THEN PRINT"Error in data":STOP
320 faradd$(1)=%FF:mfcom$(1)=%secr:mfcom$(8)=%faradd$(
1):CALLmfcom$(0):IF faradd$(1)<>7 THEN STOP'find Read_
sector. AMSDOS ROM is no 7.
330 faradd$(1)=%FF:mfcom$(1)=%secr:mfcom$(8)=%faradd$(
1):CALLmfcom$(0):IF faradd$(1)<>7 THEN STOP'find Write
_sector
340 ERASE mfcom%'no longer needed
350 ' asocio : read/write a sector
360 DATA 00,1e,FF,00,00,2a,FF,00,3a,FF,FF,4f,3a,FF,F
F
370 DATA 57,df,ff,FF,00,32,FF,FF,c9,00
380 c=0:FOR a=0 TO 12:READ x$,y$:msocio$(a)=VAL("%"+y$+
x$):c=msocio$(a):NEXT a
390 IF c<>53097 THEN PRINT"Error in data":STOP
400 dr$="":WHILE dr$<"A"OR dr$<"B":INPUT"Select drive (
A or B) : ",dr$:WEND
410 dr$=ASC(dr$)-65:msocio$(1)=dr$
420 CLS#3:PRINT#3,"Insert disc in drive "dr$", press EN
TER ":
430 LINE INPUT#3,x$
440 :DRIVE,edr$:recn%=0:cbkl%=-1:GOSUB 690'read first s
ector on disc
450 INPUT#3,"Is this the correct disc (n/Y) ":x$:IF x$<
">Y"THEN 420
460 ' Main
470 recn%=-1:WHILE recn%<0 OR recn%>recmax%
480 CLS#3:PRINT#3,"Enter record number (0 to'recmax% CH
R$(B) )":
490 INPUT#3,recn%
500 WEND
510 CLS#3:PRINT#3,"Record number"recn%
520 GOSUB 690'record into z$, display
530 PRINT#3,"Use SHIFT+cursor and COPY/CLR/DEL keys to
transfer and edit"
540 PRINT#3,"old record or enter new record : "
550 LINE INPUT#3,y$:v$=rul$:MID$(v$,1)=y$
560 FOR a=1 TO recsize%:PRINT#3,CHR$(1):MID$(v$,a,1):NE
XT a:PRINT#3
570 INPUT#3,"Is this correct (n/Y) ":x$:IF x$<">Y"THEN
510
580 z$=v$:LOCATE#1,recsize% MOD 64+1,(part%*recsize%)\6
4+1:PRINT#1,CHR$(24);
590 FOR a=1 TO recsize%:PRINT#1,CHR$(1):MID$(z$,a,1):PR
INT#1,CHR$(24):NEXT a
600 INPUT#3,"update buffer (n/Y) ":x$:IF x$<">Y"THEN 67
0
610 b=part%*recsize%-1+buf%
620 FOR a=1 TO recsize%:POKE a+b,ASC(MID$(z$,a)):NEXT a
'update buffer
630 PRINT#1,CHR$(24):CLS#1:PRINT#1,CHR$(24);
640 FOR a=buf% TO buf%+511:PRINT#1,CHR$(1):CHR$(PEEK(a))
):NEXT a'show update
650 INPLT#3,"Write to disc (n/Y) ":x$:IF x$<">Y"THEN 67
0
660 GOSUB 840'record write
670 GOTO 460
680 STOP
690 ' Record read
700 GOSUB 900'convert recn%
710 IF blkn%<>cbkl% THEN 740'read block if not in buffe
r
720 INPUT#3,"Block is in buffer. Read it again (n/Y) ":
x$
730 IF x$<">Y"THEN 790
740 CLS#2:PRINT#2,"Block"blkn%
750 PRINT#2,"Records"blkn%*recblock%to"blkn%*recblock%
+recblock%-1
760 cbkl%=blkn%:faradd$(0)=secr%:GOSUB 950'sector i/o
770 PRINT#1,CHR$(24):CLS#1:PRINT#1,CHR$(24);
780 FOR a=buf% TO buf%+511:PRINT#1,CHR$(1):CHR$(PEEK(a))
):NEXT a
790 z$=rul$:b=part%*recsize%-1+buf%:PRINT#3,"Current re
cord : "
800 FOR a=1 TO recsize%
810 x$=CHR$(PEEK(a+b)):MID$(z$,a)=x$:PRINT#3,CHR$(1):x$
:
820 NEXT a:PRINT#3
830 RETURN
840 ' Record write
850 CLS#2:PRINT#2,"Block"blkn%
860 PRINT#2,"Records"blkn%*recblock%to"blkn%*recblock%
+recblock%-1
870 faradd$(0)=secr%:GOSUB 950'sector i/o
880 PRINT#2,"Block written"
890 RETURN
900 ' convert record number to block, track, sector
910 blkn%=recn%\recblock%:part%=recn% MOD recblock%
920 tracn%=blkn%\sectr%:secn%=secn%+blkn% MOD sectr%
930 'PRINT recn%,blkn%,part%,tracn%,secn%
940 RETURN
950 ' Sector i/o
960 er$=%FF:asecio$(3)=@buf%:asecio$(5)=@secn%:asecio$(
7)=@tracn%
970 asecio$(9)=%faradd$(0):asecio$(11)=@er%:CALLasecio
$(0)
980 PRINT#2,"Track "LOWER$(HEX$(tracn%,2))
990 PRINT#2,"Sector "LOWER$(HEX$(secn%,2)):PRINT#2,"r/w
error: "BIN$(er%,8)
1000 RETURN

```