COMMODORE 1541 DISK DRIVE

A Friendly Introduction to Your 1541 Disk Drive

USER'S GUIDE

commodore
COMPUTER

USER'S GUIDE STATEMENT

"This equipment generates and uses radio frequency energy. If it is not properly installed and used in strict accordance with the manufacturer's instructions, this equipment may interfere with radio and television reception. This machine has been tested and found to comply with the limits for a Class B computing device peripheral in accordance with the specifications in Subpart J of Part 15 of FCC Rules, which are designed to provide reasonable protection against such interference in a residential installation. If you suspect interference, you can test this equipment by turning it off and on. If you determine that there is interference with radio or television reception, try one or more of the following measures to correct it:

- reorient the receiving antenna
- move the equipment away from the receiver
- change the relative positions of the equipment and the receiver
- plug the equipment into a different outlet so that the equipment and the receiver are on different branch circuits.

If necessary, consult your Commodore dealer or an experienced radio/television technician for additional suggestions. You may also wish to consult the following booklet, which was prepared by the Federal Communications Commission:

"How to Identify and Resolve Radio-TV Interference Problems". This booklet is available from the U.S. Government Printing Office, Washington, D.C. 20402, Stock No. 004-000-00345-4.

COMMODORE 1541 DISK DRIVE USER'S GUIDE

A Friendly Introduction to Your 1541 Disk Drive

commodore
COMPUTER

The information in this manual has been reviewed and is believed to be entirely reliable. No responsibility, however, is assumed for inaccuracies. The material in this manual is for information purposes only, and is subject to change without notice.

Table of Contents

Page

Appendices

1. GENERAL DESCRIPTION

INTRODUCTION

Welcome to the fastest, easiest, and most efficient filing system available for your Commodore 64 or VIC 20 computer, your 1541 DISK DRIVE. This manual has been designed to show you how to get the most from your drive, whether you're a beginner or an advanced professional.

If you are a beginner, the first few chapters will help you through the basics of disk drive installation and operation. As your skill and programming knowledge improves, you will find more uses for your disk drive and the more advanced chapters of this manual will become much more valuable.

If you're a professional, this reference guide will show you how to put the 1541 through its paces to perform just about all the disk drive jobs you can think of.

No matter what level of expertise you have, your 1541 disk drive will dramatically improve the overall capabilities of your computer system.

Before you get to the details of 1541 operation, you should be aware of a few important points. This manual is a REFERENCE GUIDE, which means that unless the information you seek directly pertains to the disk or disk drive you will have to use your Commodore 64 or VIC 20 User's Guides and Programmer's Reference Guides to find programming information. In addition, even though we give you step-by-step instructions for each operation, you should become familiar with BASIC and the instructions (called commands) that help you operate your disks and drives. However, if you just want to use your disk drive unit to load and save prepackaged software, we've included an easy and brief section on doing just that.

Now . . . let's get on with the general information.

The commands for the disk drive come in several levels of sophistication. Starting in chapter three, you can learn how the command that allow you to SAVE and LOAD programs with the disk work. Chapter four teaches you how commands are sent to the disk, and introduces the disk maintenance commands.

Chapter five tells you how to work with SEQuential data files. These are very similar to their counterparts on tape (but much faster). Chapter six introduces the commands that allow you to work with random files, to access any piece of data on the disk, and how you organize the diskette into tracks and blocks. Chapter seven describes the special relative files. RELative files are the best method of storing data bases, especially when they are used along with sequential files.

Chapter eight describes methods for programming the disk controller circuits at the machine language level. And the final chapter shows you how to change the disk device number, by cutting a line inside the drive unit or through software.

APPENDIX A: DISK COMMAND SUMMARY

General Format: PRINT#file#, command

COMMAND	
NEW	“N
COPY	“C:new file= :orginal file
RENAME	“R:new name= old name
SCRATCH	“S:file name
INITIALIZE	“I
VALIDATE	“V
DUPLICATE	not for single drives
BLOCK-READ	“B-R:” channel; drive; track; block
BLOCK-WRITE	“B-W:” channel; drive; track; block
BLOCK-ALLOCATE	“B-A:” drive; track; block
BLOCK-FREE	“B-F:” drive; track; block
BUFFER-POINTER	“B-P:” channel; position
USER1 and USER2	“Un:” channel; drive; track; block
POSITION	“P” CHR$(channel#) CHR$(rec#lo) CHR$(rec#hi) CHR$(position)
BLOCK-EXECUTE	“B-E:” channel; drive; track; block
MEMORY-READ	“M-R” CHR$(address lo) CHR$(address hi)
MEMORY-WRITE	“M-W” CHR$(address lo) CHR$(address hi) CHR$(#chars) “data”
MEMORY-EXECUTE	“M-E” CHR$(address lo) CHR$(address hi)
USER Commands	“Un”

APPENDIX B: SUMMARY OF CBM FLOPPY ERROR MESSAGES

0	OK, no error exists.
1	Files scratched response. Not an error condition.
2-19	Unused error messages: should be ignored.
20	Block header not found on disk.
21	Sync character not found.
22	Data block not present.
23	Checksum error in data.
24	Byte decoding error.
25	Write-verify error.
26	Attempt to write with write protect on.
27	Checksum error in header.
28	Data extends into next block.
29	Disk id mismatch.
30	General syntax error
31	Invalid command.
32	Long line.
33	Invalid filename.
34	No file given.
39	Command file not found.
50	Record not present.
51	Overflow in record.
52	File too large.
60	File open for write.
61	File not open.
62	File not found.
63	File exists.
64	File type mismatch.
65	No block.
66	Illegal track or sector.
67	Illegal system track or sector.
70	No channels available.
71	Directory error.
72	Disk full or directory full.
73	Power up message, or write attempt with DOS Mismatch.
74	Drive not ready.

DESCRIPTION OF DOS ERROR MESSAGES

NOTE: Error message numbers less than 20 should be ignored with the exception of 01 which gives information about the number of files scratched with the SCRATCH command.

20: READ ERROR (block header not found)
The disk controller is unable to locate the header of the requested data block. Caused by an illegal block number, or the header has been destroyed.

21: READ ERROR (no sync character)
The disk controller is unable to detect a sync mark on the desired track. Caused by misalignment of the read/writer head, no diskette is present, or unformatted or improperly seated diskette. Can also indicate a hardware failure.

22: READ ERROR (data block not present)
The disk controller has been requested to read or verify a data block that was not properly written. This error message occurs in conjunction with the BLOCK commands and indicates an illegal track and/or block request.

23: READ ERROR (checksum error in data block)
This error message indicates that there is an error in one or more of the data bytes. The data has been read into the DOS memory, but the checksum over the data is in error. This message may also indicate grounding problems.

24: READ ERROR (byte decoding error)
The data or header as been read into the DOS memory, but a hardware error has been created due to an invalid bit pattern in the data byte. This message may also indicate grounding problems.

25: WRITE ERROR (write-verify error)
This message is generated if the controller detects a mismatch between the written data and the data in the DOS memory.

26: WRITE PROTECT ON
This message is generated when the controller has been requested to write a data block while the write protect switch is depressed. Typically, this is caused by using a diskette with a write a protect tab over the notch.

27: READ ERROR (checksum error in header)
The controller has detected an error in the header of the requested data block. The block has not been read into the DOS memory. This message may also indicate grounding problems.

28: WRITE ERROR (long data block)
The controller attempts to detect the sync mark of the next header after writing a data block. If the sync mark does not appear within a pre-determined time, the error message is generated. The error is caused by a bad diskette format (the data extends into the next block), or by hardware failure.

29: DISK ID MISMATCH
This message is generated when the controller has been requested to access a diskette which has not been initialized. The message can also occur if a diskette has a bad header.

30: SYNTAX ERROR (general syntax)
The DOS cannot interpret the command sent to the command channel. Typically, this is caused by an illegal number of file names, or patterns are illegally used. For example, two file names may appear on the left side of the COPY command.

31: SYNTAX ERROR (invalid command)
The DOS does not recognize the command. The command must start in the first position.

32: SYNTAX ERROR (invalid command)
The command sent is longer than 58 characters.

33: SYNTAX ERROR (invalid file name)
Pattern matching is invalidly used in the OPEN or SAVE command.

34: SYNTAX ERROR (no file given)
the file name was left out of a command or the DOS does not recognize it as such. Typically, a colon (:) has been left out of the command.

39: SYNTAX ERROR (invalid command)
This error may result if the command sent to command channel (secondary address 15) is unrecognized by the DOS.

50: RECORD NOT PRESENT
Result of disk reading past the last record through INPUT#, or GET# commands. This message will also occur after positioning to a record beyond end of file in a relative file. If the intent is to expand the file by adding the new record (with a PRINT# command), the error message may be ignored. INPUT or GET should not be attempted after this error is detected without first repositioning.

51: OVERFLOW IN RECORD
PRINT# statement exceeds record boundary. Information is cut off. Since the carriage return is sent as a record terminator is counted in the record size. This message will occur if the total characters in the record (including the final carriage return) exceeds the defined size.

52: FILE TOO LARGE
Record position within a relative file indicates that disk overflow will result.

60: WRITE FILE OPEN
This message is generated when a write file that has not been closed is being opened for reading.

61: FILE NOT OPEN
This message is generated when a file is being accessed that has not been opened in the DOS. Sometimes, in this case, a message is not generated; the request is simply ignored.

62: FILE NOT FOUND
The requested file does not exist on the indicated drive.

63: FILE EXISTS
The file name of the file being created already exists on the diskette.

64: FILE TYPE MISMATCH
The file type does not match the file type in the directory entry for the requested file.

65: NO BLOCK
This message occurs in conjunction with the B-A command. It indicates that the block to be allocated has been previously allocated. The parameters indicate the track and sector available with the next highest number. If the parameters are zero (0), then all blocks higher in number are in use.

66: ILLEGAL TRACK AND SECTOR
The DOS has attempted to access a track or block which does not exist in the format being used. This may indicate a problem reading the pointer to the next block.

67: ILLEGAL SYSTEM T OR S
This special error message indicates an illegal system track or block.

70: NO CHANNEL (available)
The requested channel is not available, or all channels are in use. A maximum of five sequential files may be opened at one time to the DOS. Direct access channels may have six opened files.

71: DIRECTORY ERROR
The BAM does not match the internal count. There is a problem in the BAM allocation or the BAM has been overwritten in DOS memory. To correct this problem, reinitialize the diskette to restore the BAM in memory. Some active files may be terminated by the corrective action. NOTE: BAM = Block Availability Map

72: DISK FULL
Either the blocks on the diskette are used or the directory is at its entry limit. DISK FULL is sent when two blocks are available on the 1541 to allow the current file to be closed.

73: DOS MISMATCH (73, CBM DOS V2.6 1541)

DOS 1 and 2 are read compatible but not write compatible. Disks may be interchangeably read with either DOS, but a disk formatted on one version cannot be written upon with the other version because the format is different. This error is displayed whenever an attempt is made to write upon a disk which has been formatted in a non-compatible format. (A utility routine is available to assist in converting from one format to another.) This message may also appear after power up.

74: DRIVE NOT READY

An attempt has been made to access the 1541 Single Drive Floppy Disk without any diskettes present in either drive.

APPENDIX C: Demonstration Disk Programs

1. DIR

```
4 OPEN2,8,15
5 PRINT"♡":GOTO 10000
10 OPEN1,8,0,"$0"
20 GET#1,A$,B$
30 GET#1,A$,B$
40 GET#1,A$,B$
50 C=0
60 IF A$<>"" THEN C=ASC(A$)
70 IF B$<>"" THEN C=C+ASC(B$)*256
80 PRINT"🆁"MID$(STR$(C),2);TAB(3);"■";
90 GET#1,B$:IF ST<>0 THEN 1000
100 IF B$<>CHR$(34) THEN 90
110 GET#1,B$:IF B$<>CHR$(34)THEN PRINTB$;:GOTO110
120 GET#1,B$:IF B$=CHR$(32) THEN 120
130 PRINT TAB(18);:C$=""
140 C$=C$+B$:GET#1,B$:IF B$<>""  THEN 140
150 PRINT"🆁"LEFT$(C$,3)
160 GET T$:IF T$<>"" THEN GOSUB 2000
170 IF ST=0 THEN 30
1000 PRINT" BLOCKS FREE"
1010 CLOSE1:GOTO 10000
2000 IF T$="Q" THEN CLOSE1:END
2010 GET T$:IF T$="" THEN 2000
2020 RETURN
4000 REM DISK COMMAND
4010 C$="":PRINT">";
4011 GETB$:IFB$="" THEN4011
4012 PRINTB$;:IF B$=CHR$(13) THEN 4020
4013 C$=C$+B$:GOTO 4011
4020 PRINT#2,C$
5000 PRINT"🆁";
5010 GET#2,A$:PRINTA$;:IF A$<>CHR$(13)GOTO5010
5020 PRINT"■"
10000 PRINT "D-DIRECTORY"
10010 PRINT ">-DISK COMMAND"
10020 PRINT "Q-QUIT PROGRAM"
10030 PRINT "S-DISK STATUS "
10100 GETA$:IFA$=""THEN10100
10200 IF A$="D" THEN 10
10300 IF A$="." OR A$=">" OR A$=">" THEN 4000
10310 IF A$="Q" THEN END
10320 IF A$="S" THEN 5000
10999 GOTO 10100
```

2. VIEW BAM

```
100 REM **********************************
101 REM *  VIEW BAM FOR VIC & 64 DISK   *
102 REM **********************************
105 OPEN15,8,15
110 PRINT#15,"I0":NU$="N/A N/A N/A N/A N/A":Z4=1
120 OPEN2,8,2,"#"
130 Y$="▚▚▚▚▚▚▚▚▚▚▚▚▚▚▚▚▚▚▚▚▚▚▚▚▚▚"
140 X$="▮▮▮▮▮▮▮▮▮▮▮▮▮▮▮▮▮▮▮▮▮▮▮▮▮▮"
150 DEF FNS(Z) = 2↑(S-INT(S/8)*8) AND (SB(INT(S/8)))
160 PRINT#15,"U1:";2;0;18;0
170 PRINT#15,"B-P";2;1
180 PRINT"▛";
190 Y=22:X=1:GOSUB430
200 FORI=0TO20:PRINT:PRINT"▛▛"RIGHT$(STR$(I)+" ",3);:NEXT
210 GET#2,A$
220 GET#2,A$
230 GET#2,A$
240 TS=0
250 FORT=1TO17:GOSUB450
260 Y=22:X=T+4:GOSUB430:GOSUB540:NEXT
270 FORI=1TO2000:NEXT:PRINT"▛"
280 Y=22:X=1:GOSUB430
290 FORI=0TO20:PRINT:PRINT"▛▛"RIGHT$(STR$(I)+" ",3);:NEXT
300 FORT=18TO35
310 GOSUB450
320 Y=22:X=T-13:GOSUB430:GOSUB540:NEXT
330 FORI=1TO1000:NEXT
340 PRINT"▛▚▚▚▚▚"
350 PRINT#15,"B-P";2;144
360 N$="":FORI=1TO20:GET#2,A$:N$=N$+A$:NEXT
370 PRINT" "N$" "TS-17;"BLOCKS FREE"
380 FORI=1TO4000:NEXT
390 PRINT"▛"
400 INPUT"▚▚▮▮ANOTHER DISKETTE N▮▮▮";A$
410 IFA$="Y"THENRUN
420 IFA$<>"Y"THENEND
430 PRINTLEFT$(Y$,Y)LEFT$(X$,X)"▮▮";
440 RETURN
450 GET#2,SC$:SC=ASC(RIGHT$(CHR$(0)+SC$,1))
460 TS=TS+SC
470 GET#2,A$:IFA$=""THENA$=CHR$(0)
480 SB(0)=ASC(A$)
490 GET#2,A$:IFA$=""THENA$=CHR$(0)
500 SB(1)=ASC(A$)
510 GET#2,A$:IFA$=""THENA$=CHR$(0)
520 SB(2)=ASC(A$)
530 RETURN
540 PRINT"▚▮"RIGHT$(STR$(T),1);"▮▛▛";
550 REM  PRINTT"    "SC"  "SB(0)" "SB(1)"  "SB(2)=CHR$(0)
560 IFT>24ANDS=18THEN:PRINTMID$(NU$,Z4,1);:GOTO660
570 FORS=0TO20
580 IFT<18THEN620
590 IFT>30ANDS=17THEN:PRINTMID$(NU$,Z4,1);:GOTO660
600 IFT>24ANDS=18THEN:PRINTMID$(NU$,Z4,1);:GOTO660
610 IFT>24ANDS=19THENPRINTMID$(NU$,Z4,1);:GOTO660
620 IFT>17ANDS=20THENPRINTMID$(NU$,Z4,1);:Z4=Z4+1:GOTO660
630 PRINT"▚":
640 IF FNS(S)=0 THEN PRINT"+";:GOTO660
650 PRINT"▙+";:REMRIGHT$(STR$(S),1);Z4,1);:GOTO72
660 PRINT"▜▙";
670 NEXT
680 RETURN
```

3. DISPLAY T & S

```
100 REM********************************
110 REM* DISPLAY ANY TRACK $ SECTOR *
120 REM* ON THE DISK TO THE SCREEN  *
130 REM* OR THE PRINTER              *
140 REM********************************
150 PRINT"■■■──────────────────────────"
160 PRINT"DISPLAY BLOCK CONTENTS"
165 FRINT"─────────────────────────":
170 REM********************************
180 REM* SET PROGRAM CONSTANT        *
190 REM********************************
200 SP$=" ":NL$=CHR$(0):HX$="0123456789ABCDEF"
210 FS$="":FORI=64 TO 95:FS$=FS$+"■"+CHR$(I)+"■":NEXT I
220 SS$="  ":FOR I=192 TO 223:SS$=SS$+"■"+CHR$(I)+"■":NEXT I
240 DIM A$(15),NB(2)
251 D$="0"
253 PRINT"              ■S■CREEN■■■■■■■■■■OR ■■■P■RINTER"
254 GETJJ$:IF JJ$="" THEN254
255 IF JJ$="S"THENPRINT"           ■■SCREEN■"
256 IF JJ$="P"THENPRINT"           ■■PRINTER■"
260 OPEN15,8,15,"I"+D$:GOSUB 650
265 OPEN4,4
270 OPEN 2,8,2,"#":GOSUB 650
280 REM********************************
290 REM* LOAD TRACK AND SECTOR       *
300 REM* INTO DISK BUFFER            *
310 REM********************************
320 INPUT"■■■TRACK, SECTOR";T,S
330 IF T=0 OR T>35 THEN PRINT#15,"I"D$:CLOSE2:CLOSE4:CLOSE15:PRINT"END":END
340 IF JJ$="S" THEN PRINT"■■■TRACK"T" SECTOR"S"■"
341 IF JJ$="P" THEN PRINT#4:PRINT#4,"TRACK"T" SECTOR"S:PRINT#4
350 PRINT#15,"U1:2,"D$:T;S:GOSUB650
360 REM********************************
370 REM* READ BYTE 0 OF DISK BUFFER *
390 REM********************************
400 PRINT#15,"B-P:2,1"
410 PRINT#15,"M-R"CHR$(0)CHR$(5)
420 GET#'5,A$(0):IFA$(0)=""THENA$(0)=NL$
428 IF JJ$="S"THEN430
430 IF JJ$="P"THEN460
431 REM********************************
432 REM* READ & CRT DISPLAY          *
433 REM* REST OF THE DISK BUFFER     *
434 REM**************-*****************
436 K=1:NB(1)=ASC(A$(0))
438 FOR J=0 TO 63:IF J=32 THEN GOSUB 710:IF Z$="N"THEN J=80:GOTO 458
440 FOR I=K TO 3
442 GET#2,A$(I):IF A$(I)="" THEN A$(I)=NL$
444 IF K=1 AND I<2 THEN NB(2)=ASC(A$(I))
446 NEXT I:K=0
448 A$="":B$=":":N=J*4:GOSUB 790:A$=A$+":"
450 FOR I=0 TO 3:N=ASC(A$(I)):GOSUB 790
452 C$=A$(I):GOSUB 850:B$=B$+C$
454 NEXT I:IF JJ$="S" THEN PRINTA$B$
458 NEXT J:GOTO571
```

```
460 REM*******************************
462 REM* READ & PRINTER DISPLAY      *
464 REM*******************************
466 K=1:NB(1)=ASC(A$(0))
468 FOR J=0 TO 15
470 FOR I=K TO 15
472 GET#2,A$(I):IF A$(I)="" THEN A$(I)=NL$
474 IF K=1 AND I<2 THEN NB(2)=ASC(A$(I))
476 NEXT I:K=0
478 A$="":B$=":":N=J*16:GOSUB 790:A$=A$+":"
480 FOR I=0 TO 15:N=ASC(A$(I)):GOSUB 790:IF Z$="N"THEN J=40:GOTO 571
482 C$=A$(I):GOSUB 850:B$=B$+C$
484 NEXT I
486 IF JJ$="P" THEN PRINT#4,A$B$
488 NEXT J:GOTO571
571 REM*******************************
572 REM* NEXT TRACK AND SECTOR       *
573 REM*******************************
575 PRINT"NEXT TRACK AND SECTOR"NB(1)NB(2) "◘"
580 PRINT"DO YOU WANT NEXT TRACK AND SECTOR"
590 GET Z$:IF Z$="" THEN590
600 IP Z$="Y" THEN T=NB(1):S=NB(2):GOTO330
610 IF Z$="N" THEN 320
620 GOTO 590
630 REM*******************************
640 REM* SUBROUTINES                 *
650 REM*******************************
660 REM* ERROR ROUTINE               *
670 REM*******************************
680 INPUT#15,EN,EM$,ET,ES:IF EN=0 THEN RETURN
690 PRINT"◘DISK ERROR■"EN,EM$,ET,ES
700 END
710 REM*******************************
720 REM* SCREEN CONTINUE MESSAGE     *
730 REM*******************************
740 PRINT"◘▶▶▶CONTINUE(Y/N)"
750 GETZ$:IF Z$="" THEN 750
760 IF Z$="N" THEN RETURN
770 IF Z$<>"Y" THEN 750
780 PRINT"□TRACK" T " SECTOR"S "□":RETURN
790 REM*******************************
800 REM* DISK BYTE TO HEX PRINT      *
810 REM*******************************
820 A1=INT(N/16):A$=A$+MID$(HX$,A1+1,1)
830 A2=INT(N-16*A1):A$=A$+MID$(HX$,A2+1,1)
840 A$=A$+SP$:RETURN
850 REM*******************************
860 REM* DISK BYTE TO ASC DISPLAY    *
870 REM* CHARACTER                   *
880 REM*******************************
890 IF ASC(C$)<32 THEN C$=" ":RETURN
910 IF ASC(C$)<128 OR ASC(C$)>159 THEN RETURN
920 C$=MID$(SS$,3*(ASC(C$)-127),3):RETURN
```

4. CHECK DISK

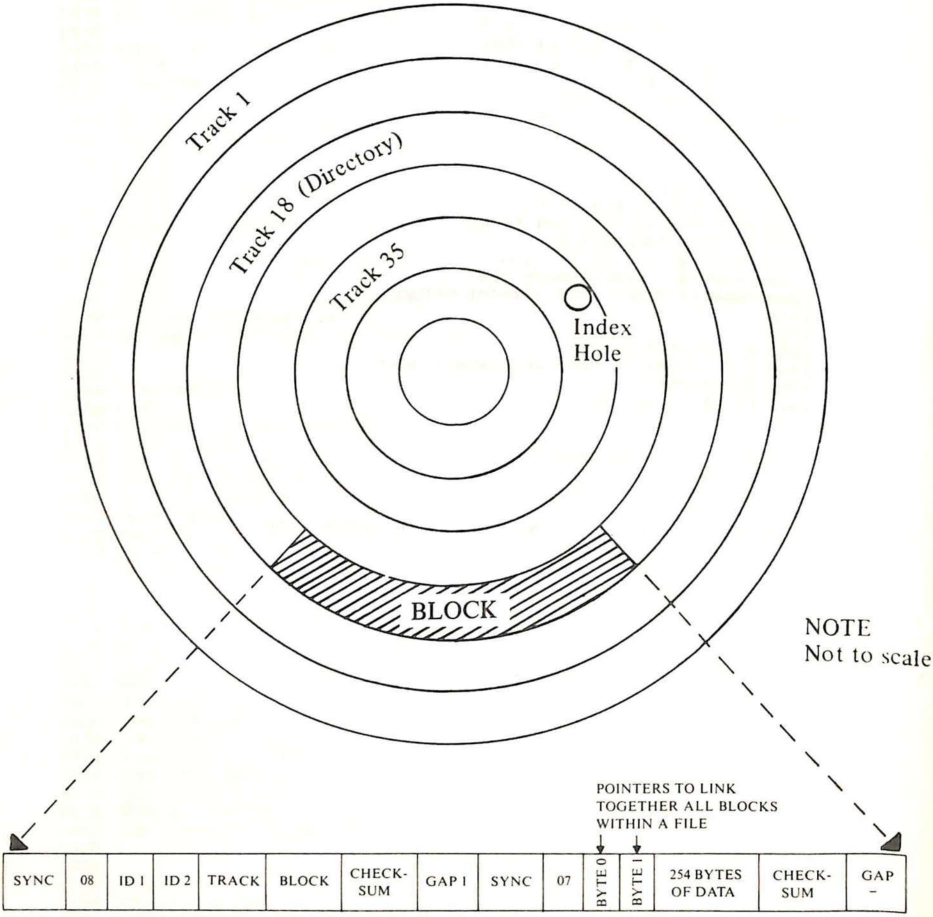
```
1 REM CHECK DISK -- VER 1.4
2 DN=8:REM FLOPPY DEVICE NUMBER
5 DIMT(100):DIMS(100):REM BAD TRACK, SECTOR ARRAY
9 PRINT"▮▮▮▮────────────────────────"
10 PRINT" CHECK DISK PROGRAM"
12 PRINT"──────────────────────"
20 D$="0"
30 OPEN15,DN,15
35 PRINT#15,"V"D$
45 N%=RND(TI)*255
50 A$="":FORI=1TO255:A$=A$+CHR$(255AND(I+N%)):NEXT
60 GOSUB900
70 OPEN2,DN,2,"#"
80 PRINT:PRINT#2,A$;
85 T=1:S=0
90 PRINT#15,"B-A:"D$;T;S
100 INPUT#15,EN,EM$,ET,ES
110 IFEN=0THEN130
115 IFET=0THEN200:REM END
120 PRINT#15,"B-A:"D$;ET;ES:T=ET:S=ES
130 PRINT#15,"U2:2,"D$;T;S
134 NB=NB+1:PRINT" CHECKED  BLOCKS"NB
135 PRINT" TRACK    ▮▮▮▮▮"T;" SECTOR    ▮▮▮▮▮"S"▮"
140 INPUT#15,EN,EM$,ES,ET
150 IF EN=0THEN85
160 T(J)=T:S(J)=S:J=J+1
165 PRINT"▮▮BAD BLOCK:▮▮▮",T;S""
170 GOTO85
200 PRINT#15,"I"D$
210 GOSUB900
212 CLOSE2
215 IFJ=0THENPRINT"▮▮▮▮▮▮NO BAD BLOCKS!":END
217 OPEN2,DN,2,"#"
218 PRINT"▮▮BAD BLOCKS","TRACK","SECTOR"
220 FORI=0TOJ-1
230 PRINT#15,"B-A:";D$,T(I);S(I)
240 PRINT,,T(I),S(I)
250 NEXT
260 PRINT"▮"J"BAD BLOCKS HAVE BEEN ALLOCATED"
270 CLOSE2:END
900 INPUT#15,EN,EM$,ET,ES
910 IF EN=0 THEN RETURN
920 PRINT"▮▮ERROR #"EN,EM$;ET;ES""
930 PRINT#15,"I"D$
```

5. PERFORMANCE TEST

```
1000 REM  PERFORMANCE TEST  2.0
1010 :
1020 REM  VIC-20 AND COMMODORE 64
1030 REM  SINGLE FLOPPY DISK DRIVE
1040 :
1050 OPEN 1,8,15:OPEN15,8,15
1060 LT=35
1070 LT$=STR$(LT)
1080 NT=30
1090 PRINT"♥◘────────────────────────"
1100 PRINT"   PERFORMANCE TEST"
1110 PRINT"────────────────────────"
1120 PRINT
1130 PRINT"  INSERT SCRATCH"
1140 PRINT
1150 PRINT"     DISKETTE IN DRIVE"
1160 PRINT
1170 PRINT"◘    PRESS ■RETURN▬"
1180 PRINT
1190 PRINT"             WHEN READY◘"
1200 FOR I=0 TO 50:GET A$:NEXT
1210 GET A$:IF A$<>CHR$(13) THEN 1210
1220 :
1230 :
1240 TI$="000000"
1250 TT=18
1260 PRINT#1,"N0:TEST DISK,00"
1270 C1$="   DISK NEW COMMAND   "+CHR$(13)
1280 C2$="◘ WAIT ABOUT 80 SECONDS"
1290 CC$=C1$+C2$:GOSUB 1840
1300 IF TI<NTTHEN1370
1310 PRINT"◘SYSTEM IS"
1320 PRINT"◘          NOT RESPONDING"
1330 PRINT" CORRECTLY TO COMMANDS"
1340 GOSUB 1880
1350 :
1360 :
1370 PRINT"◘DRIVE PASS"
1380 PRINT"         MECHANICAL TEST◘"
1390 TT=21
1400 OPEN 2,8,2,"0:TEST FILE,S,W"
1410 CC$="OPEN WRITE FILE"        :GOSUB 1840
1420 CH=2:CC$="WRITE DATA"        :GOSUB 1930
1430 CC$="CLOSE "+CC$             :GOSUB 1840
1440 OPEN 2,8,2,"0:TEST FILE,S,R"
1450 CC$="OPEN READ FILE"         :GOSUB 1840
1460 CH=2:GOSUB 1990
1470 PRINT#1,"S0:TEST FILE"
1480 CC$="SCRATCH FILE◘":TT=1     :GOSUB 1840
1490 :
1500 :
1510 TT=21
1520 OPEN 4,8,4,"#"
1530 NN%=(1+RND(TI)*254+NN%)AND255:PRINT#1,"B-P";4;NN%
1540 NN$="":FOR I=1 TO 255:NN$=NN$+CHR$(I):NEXT
1550 PRINT# 4,NN$;
1560 PRINT# 1,"U2:";4;0;LT;0
1570 CC$="WRITE TRACK"+LT$:GOSUB 1840
1580 PRINT#1,"U2:";4;0;1;0
1590 CC$="WRITE TRACK 1"          :GOSUB 1840
1600 PRINT#1,"U1:";4;0;LT;0
1610 CC$="READ TRACK"+LT$         :GOSUB 1840
1620 PRINT#1,"U1:";4;0;1;0
1630 CC$="READ TRACK 1"           :GOSUB 1840
1640 CLOSE 4
1650 :
1660 :
```

```
1670 PRINT"▮ UNIT HAS PASSED"
1680 PRINT"      PERFORMANCE TEST!"
1690 PRINT"▮ PULL DISKETTE FROM"
1700 PRINT"▮  DRIVE BEFORE TURNING"
1710 PRINT"   POWER OFF."
1720 END
1730 :
1740 :
1750 PRINT"   ▮CONTINUE (Y/N)?";
1760 FOR I=0 TO 50:GET A$:NEXT
1770 GET A$:IF A$="" THEN 1770
1780 PRINT A$"▮"
1790 IF A$="N" THEN END
1800 IF A$="Y" THEN RETURN
1810 GOTO 1760
1820 :
1830 :
1840 PRINT CC$
1850 INPUT# 1,EN,EM$,ET,ES
1860 PRINTTAB(12)""EN;EM$;ET;ES;""
1870 IF EN<2 THEN RETURN
1880 PRINT"▮ UNIT IS FAILING"
1890 PRINT"▮   PERFORMANCE TEST"
1900 TM$=TI$:GOSUB 1750:TI$=TM$:RETURN
1910 :
1920 :
1930 PRINT"WRITING DATA"
1940 FOR I=1000 TO 2000:PRINT#CH,I:NEXT
1950 GOSUB1850
1960 CLOSE CH:RETURN
1970 :
1980 :
1990 PRINT"READING DATA"
2000 GETA$
2010 FOR I=1000 TO 2000
2020 INPUT# CH,J
2030 IF J<>I THEN PRINT"▓READ ERROR:■":GOSUB 1850
2040 NEXT
2050 GOSUB 1850
2060 CLOSE CH:RETURN
```

APPENDIX D: DISK FORMATS



SYNC	08	ID 1	ID 2	TRACK	BLOCK	CHECK-SUM	GAP 1	SYNC	07	BYTE 0	BYTE 1	254 BYTES OF DATA	CHECK-SUM	GAP –

1540/1541 Format: Expanded View of a Single Sector

BLOCK DISTRIBUTION BY TRACK

Track number	Block Range	Total
1 to 17	0 to 20	21
18 to 24	0 to 18	19
25 to 30	0 to 17	18
31 to 35	0 to 16	17

1540/1541 BAM FORMAT

Track 18, Sector 0.

BYTE	CONTENTS	DEFINITION
0,1	18,01	Track and block of first directory block.
2	65	ASCII character A indicating 4040 format.
3	0	Null flag for future DOS use.
4—143		Bit map of available blocks for tracks 1—35.

*1 = available block
0 = block not available
(each bit represents one block)

1540/1541 DIRECTORY HEADER

Track 18, Sector 0.

BYTE	CONTENTS	DEFINITION
144—161		Disk name padded with shifted spaces.
162—163		Disk ID.
164	160	Shifted space.
165—166	50,65	ASCII representation for 2A which is DOS version and format type.
166—167	160	Shifted spaces.
177—255	0	Nulls, not used.

Note: ASCII characters may appear in locations 180 thru 191 on some diskettes.

SEQUENTIAL FORMAT

BYTE	DEFINITION
0—1	Track and block of next sequential data block.
2—256	265 bytes of data with carriage return as record terminators.

PROGRAM FILE FORMAT

BYTE	DEFINITION
0,1	Track and block of next block in program file.
2—256	265 bytes of program info stored in CBM memory format (with key words tokenized). End of file is marked by three zero bytes.

DIRECTORY FORMAT

Track 18, Block 1 for 1540/1541	
BYTE	DEFINITION
0—1	Track and block of next directory block.
2—31	*File entry 1
34—63	*File entry 2
66—95	*File entry 3
98—127	*File entry 4
130—159	*File entry 5
162—191	*File entry 6
194—123	*File entry 7
226—255	*File entry 8

*STRUCTURE OF SINGLE DIRECTORY ENTRY

BYTE	CONTENTS	DEFINITION
0	128 + type	File type OR'ed with $80 to indicate properly closed file. TYPES: 0 = DELeted 1 = SEQential 2 = PROGram 3 = USER 4 = RELative
1—2		Track and block of 1st data block.
3—18		File name padded with shifted spaces.
19—20		Relative file only: track and block for first side sector block.
21		Relative file only: Record size.
22—25		Unused.
26—27		Track and block of replacement file when OPEN@ is in effect.
28—29		Number of blocks in file: low byte, high byte.

RELATIVE FILE FORMAT

DATA BLOCK	
BYTE	DEFINITION
0,1	Track and block of next data block.
2—256	254 bytes of data. Empty records contain FF (all binary ones) in the first byte followed by 00 (binary all zeros) to the end of the record. Partially filled records are padded with nulls (00).

SIDE SECTOR BLOCK	
BYTE	DEFINITION
0—1	Track and block of next side sector block.
2	Side sector number (0-5)
3	Record length
4—5	Track and block of first side sector (number 0)
6—7	Track and block of second side sector (number 1)
8—9	Track and block of third side sector (number 2)
10—11	Track and block of fourth side sector (number 3)
12—13	Track and block of fifth side sector (number 4)
14—15	Track and block of sixth side sector (number 5)
16—256	Track and block pointers to 120 data blocks

INDEX

COMMODORE SALES CENTERS

Commodore Business Machines, Inc.
1200 Wilson Drive
West Chester, PA 19380, U.S.A.

Commodore Business Machines Limited
3370 Pharmacy Avenue, Agincourt
Ontario, M1W 2K4, Canada

Commodore Business Machines (UK) Ltd.
1, Hunters Road, Weldon
Corby, Northants, NN17 1QX, England

Commodore Bueromaschinen GmbH
PO BOX 710126, Lyonerstrasse 38
6000 Frankfult 71, West Germany

Commodore Italiana S.P.A.
Via Fratelli Gracchi 48
20092 Cinisello Balsamo, Milano, Italy

Commodore Business Machines Pty Ltd.
5 Orion Road
Lane Cove, NSW 2066, Australia

Commodore Computer B.V.
Kabelweg 88
1014 BC AMSTERDAM, Netherlands

Commodore AG(Scheweiz)
Aeschenvorstadt 57
Ch-4010 Basel, Switzerland

Commodore Computer NV-SA
Europalaan 74
1940 ST-STEVENS-WOLUWE, Belgium

Commodore Data AS
Bjerrevej 67
Horsens, Denmark

DISK DRIVES

MODEL SFD 1001

1 Megabyte double sided, floppy disk drive.
Uses double density, double sided diskettes.
IEEE interface.

MODEL 2031LP

Media compatible with 1541 and 4040 diskettes.
170 K of memory per single side disk.
Rugged and stylish housing.
IEEE interface.

PRINTERS

MODEL MPS-801

High quality dot matrix printing.
80 column printing.
Prints 50 characters per second.
Serial port interface.

MODEL MCS-801

Seven color, dot matrix printing.
Up to 80 columns printed per page.
Prints 38 characters per second.
Perfect for everything from personal letters to important business reports.
Serial port interface.

MODEL 1526

80 column, dot matrix printing.
Prints 80 characters per second.
Durability at an affordable price.
Serial Port interface.

PRINTER/PLOTTER

MODEL 1520

Four color, printer/plotter for all graphic needs.
Print bar graphs. . .create pie charts. . .plot time graphs.
High quality printing in a variety of styles and sizes.
Great for graphic artists.

MODEMS

VICMODEM

Timely financial information, news and reference libraries are as near as your phone.
An affordable introduction to the growing world of phone/computer services.
Communicate with a wide variety of computer users who already own a modem.

AUTOMODEM

The convenience of automatic ansering and dialing.
The resource power of telecomputing services.
Turn your phone into a telecomputing information network.

commodore
COMPUTER

P/N 1540031-02 ⑤

Printed in Japan