MEMOTECH SINGLE DISC SYSTEM OPERATOR'S MANUAL

CONTENTS

Chapter One

INTRODUCTION

The Memotech **SDX** Single Disc is a compact and fast Disc handling system. It is simple to use and in a short period of time even a novice will feel completely at home.

In order to use your **SDX** you will need one of the MTX series computers and a suitable T.V. or Monitor.

The **SDX** consists of a 5.25" drive and Power Supply housed in a neat black case connected by Ribbon Cable to an externally fitted Controller Board which is housed in a brushed aluminium case.

		CP/M Config code
Drive options:	100K.....SS/SD	00
	250K.....SS/DD	02
	500K.....DS/DD	03
	1Mb.....DS/DD	07

The **SDX** system can be expanded by one further drive and upgraded to a full CP/M system using the Memotech Colour 80 Column / Twin RS232 board which is fitted internally in the MTX.

ALTERATIONS - SDX SINGLE DISC SYSTEM ONLY

Chapter Two: Setting up the Hardware

1. Plug the **SDX** Controller into the left hand port of the MTX.

2. Plug the Ribbon Cable into the back of the **SDX** system making sure that pin one (denoted by the red strand on the Ribbon Cable) goes to pin one marked on the back of the **SDX.**

3. Plug the **SDX** into the mains and switch on using the switch at the rear of the system.

4. Switch on your MTX and T.V. or Monitor.

5. Insert System Disc and type **ROM 3** <RET>.

6. Before you go any further **MAKE BACKUPS OF YOUR DISC.** (Ref.Utility programs.)

Chapter Three: Single Disc BASIC

1. **ROM 5** is replaced by **ROM 3**

Chapter Two : Setting up the Hardware

1 Undo the three allen head bolts on the right hand side end plate of the MTX.

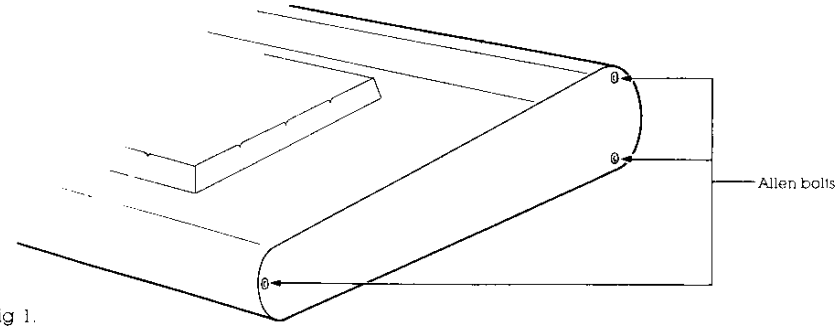


Fig 1.

2 Undo the rear bottom bolt on the left hand side end plate.

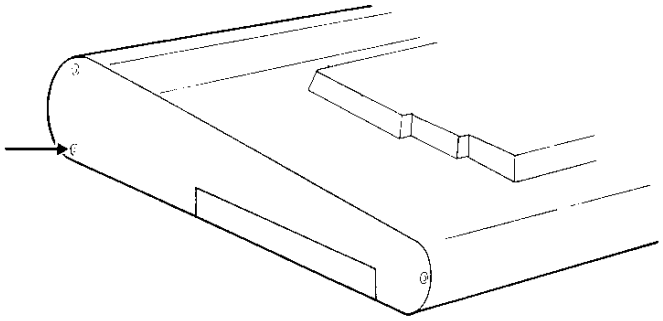


Fig 2

3 Lift the MTX at the rear so that it opens up.

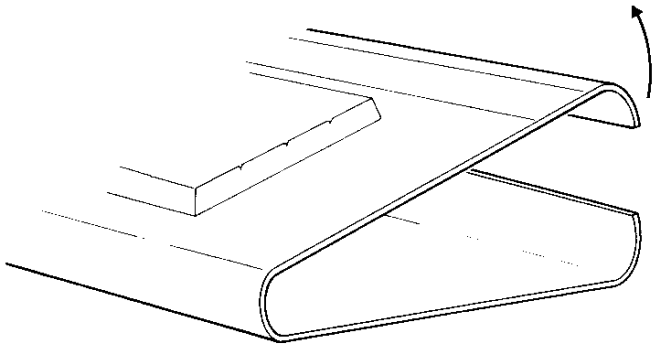


Fig 3.

4 Slide in the RS232 board and ensure the edge connector makes good contact with the edge of the Mother Board.
NOTE: If you have an extension RAM board already connected go to Section 9.

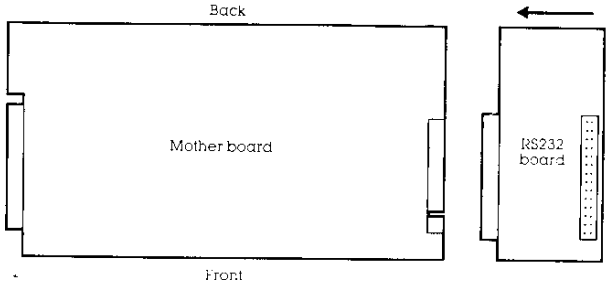


Fig 4.

5 Plug the Interface cable into the 60-way header plug on the RS232 card, ensure pin 1 on the socket (indicated by an arrow head) goes to pin 1 on the RS232 card. The cable should stick out to the right (away from the Mother Board). If it does not, and the pins are lined up correctly, remove the cable and plug the other end into the RS232 card.

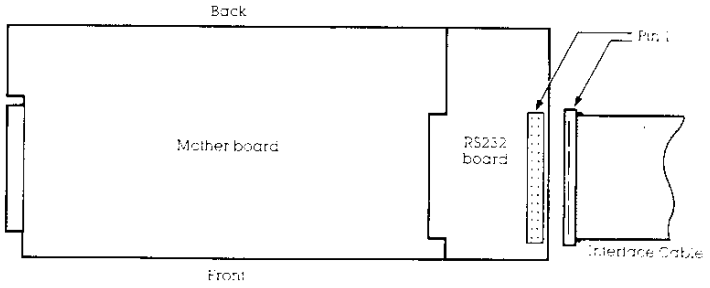
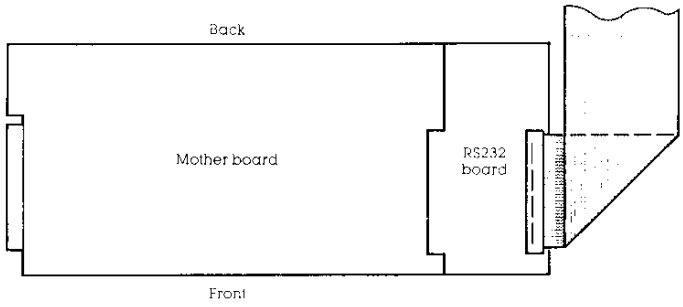
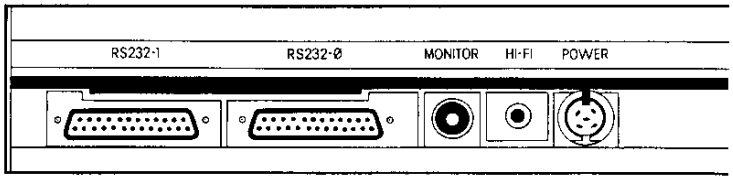


Fig 5.

6 Fold the cable close to the connector at ninety degrees, so the cable now protrudes from the back of the MTX



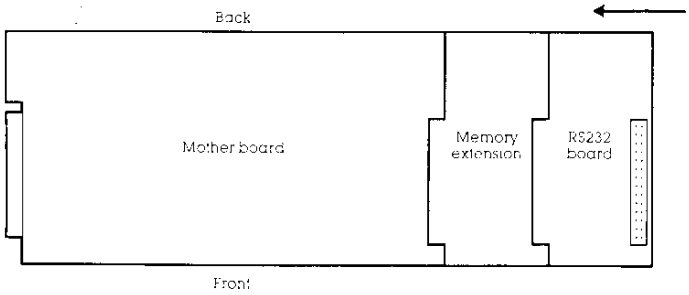
7 The cable will lay in the recess above the RS232 ports.



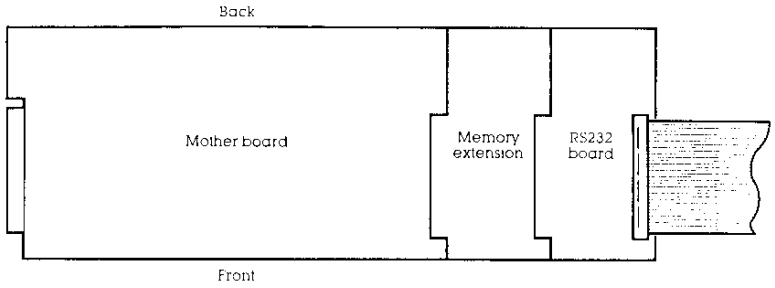
8 Close the MTX and replace the end plates and allen bolts.

9 Users with RAM expansion cards proceed from here. Users without RAM expansion cards go to Section 16

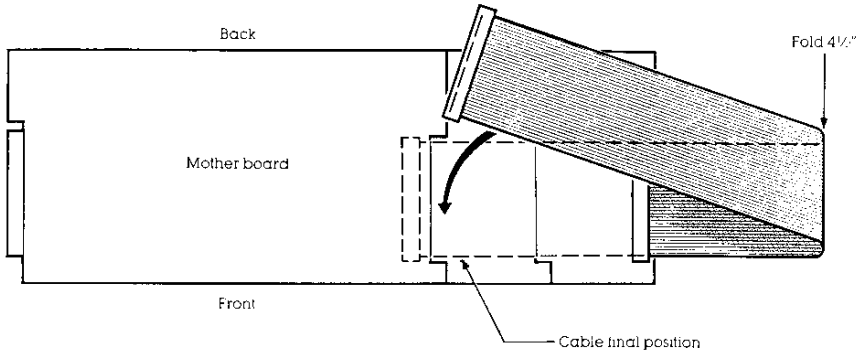
10 Slide in the RS232 board ensuring that a good contact is made between the RS232 board and the memory expansion board. Because of possible oxidisation of the connector contacts, it is a good idea to clean them before inserting. This can be done with a soft wire brush, for instance a suede shoe brush or a soft pencil rubber.



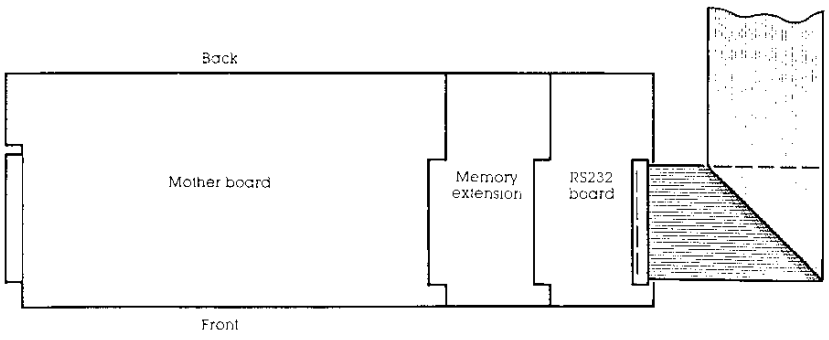
11 Plug in the Interface cable so that the cable points away from the Mother Board and that pin 1 on the cable is lined up with pin 1 on the connector (pin 1 is marked by an arrow head).



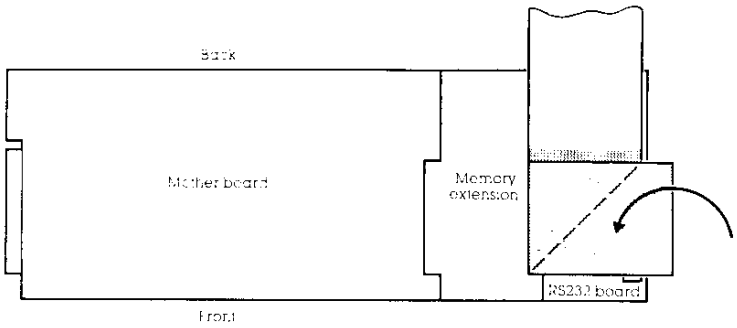
12 Fold the cable back onto itself so that the fold is 4½ inches from the connector.



13 Fold the cable once more so that the front edge of the cable lines up with the fold you have just made. The cable should now lay out of the back of the MTX.

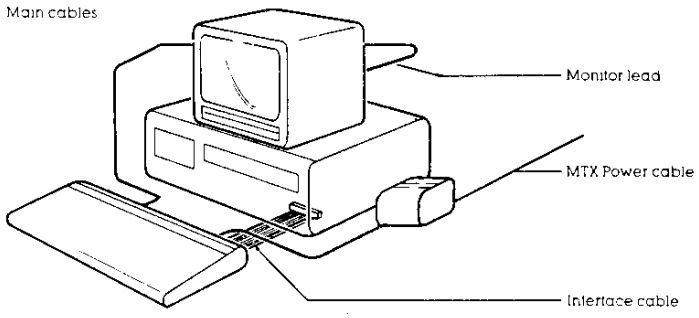


14 Fold the complete cable along the edge of the connector on the RS232 card as shown. The cable should now point out of the back of the MTX, and lay in the recess above the RS232 ports.



15 Close the MTX and replace the end plates and allen bolts.

16 Plug the interface cable from the MTX into the socket in the base of the FDX, as shown.



17 Connect the power lead into the back of the MTX, and plug the Power Supply into the mains.

18 Stand your TV or Monitor on top of the FDX case and then connect the interconnection cable to the rear of the MTX.

19 Plug the Monitor or TV into the mains and switch on.

20 Connect the FDX mains lead, and plug into the mains.

21 Switch on the FDX (red switch on front panel)

22 Observe the following -

A Mains switch illuminated
B Sign on message appears on screen

This document should be regarded as an addition to the MTX Series user manual and should be used in conjunction with that manual. The commands described in the following section are additional to those in standard MTX Basic, and relate to file handling.

The filenames you use should conform to CP/M's requirements for filenames (for example, all Basic programs should use the extension .BAS).

CP/M FILENAME CONVENTIONS

A disc file name consists of two parts: the name and the extension, separated by a fullstop - e.g. NAME.EXT. The extension is optional, but is useful for identifying groups of similar files.

The file name can be 8 characters or less
The extension can be 3 characters or less

An unambiguous file name cannot use any of the following characters:
< > . , ; : = ? * ( )
An ambiguous file name is one which replaces some characters by "?" or "*".

A "?" used in a file name will match any character which falls in the same position. A "*" will match any string which falls in the same position. This is useful when listing or erasing groups of files, e.g.

```
USER DIR "*.BAS"
```

will list all files with extension .BAS

```
USER ERA "PROGRAM?.BAS"
```

would erase the following files:

PROGRAM1.BAS, PROGRAM2.BAS, PROGRAMX.BAS etc....

All file handling commands are prefixed by USER, e.g.

```
10 USER SAVE "PROG.DAT"
```

"USER" is dedicated to the disc commands, and is therefore no longer available to the programmer.

There are two types of program on the disc supplied with the Single Disc System, ie "*.RUN" and "*.BAS".
.BAS files are loaded using USER LOAD
.RUN files are loaded using USER RUN

```
eg:     USER LOAD "POT1.BAS"
        USER RUN "ASTROPAC.RUN"
```

N.B: Function Key F8 produces command word USER.

CLOSE

Format: CLOSE #<channel no>

Purpose: To conclude or close Input/Output to one or all disc files.

Notes: <channel no> is an integer expression whose value is 1,2,3 or 4. The number is associated with the file for as long as it is OPEN, and disc I/O statements use the channel number to refer to the file. If the channel number and # sign are ommitted all currently open files will be closed.

The association between a particular file and channel number terminates upon execution of a CLOSE. The file may be reopened using the same or a different channel number.

A CLOSE for a sequential output file appends an EOF (End Of File) marker to the end of the file.

Examples: 10 USER CLOSE #2

It is important to ensure files are closed before attempting to re-open them or a BDOS ERROR will result.

DIR

Format: DIR (string expression)

Purpose: To list the files on Disc

Notes: DIR uses the CP/M Filename conventions ie:

"*" = any string "?" = any character

Example: 10 USER DIR "*.BAS"

USER DIR "B*.BAS"

USER DIR "PROG?.BAS" could list PROG1.BAS, PROG2.BAS etc.

EOF

Format: EOF #<channel no>,<line no>

Purpose: Tests end of file condition for specified channel.

Notes: <channel no> is an integer expression whose value is 1,2,3 or 4. The number is associated with the file for as long as it is OPEN, and other disc I/O statements use the channel number to refer to the file.

If the EOF condition is met on channel <channel no>, the program branches to <line no>.
When using random files, use EOF after a REC command to test whether you are reading beyond the end of the file.

Example:
```
10 USER EOF#1,999
999 PRINT "Error - End Of File reached"
```

ERA

Format: ERA (string expression)

Purpose: To erase files on Disc

Notes: ERA uses the CP/M Filename convention.

Example: 10 USER ERA "*.DAT" erases all files of type .DAT

INPUT

Format: INPUT #<channel no>,arguments

Purpose: To read data items from a disc file and assign them to program variables.

Notes: <channel no> is an integer expression whose value is 1,2,3 or 4. The number is associated with the file for as long as it is OPEN, and other disc I/O statements use the channel number to refer to the file.

SEQUENTIAL FILES
The arguments are the variable names that will be assigned to the items in the file.

N.B. The variable type must match the type specified by the variable name in the INPUT command arguments, ie 123 can be read into a string variable or numeric variable, whereas CATS will only be accepted into a string variable.

No question mark is printed.

The data items in the file must appear in the same way they would if data were being typed in response to an INPUT statement.

Carriage return/linefeed, comma or EOF act as field delimiters. (EOF = ASCII 26)

Example:

```
10 USER INPUT #2,A$,B$
```

RANDOM FILES

With random access files a complete record is read into a named string, which **must be** at least the length of the record. Since the record is of a specified length, we recommend that the string be dimensioned to this length. Any CR/LFs or commas will not be treated as field delimiters.

Example:

```
10 USER OPEN #1, "DBASE.DAT","R",100
20 DIM D$(1,100)
30 USER INPUT #1,D$(1)
```

KILL

Format: KILL #<channel no>

Purpose: To close and erase a currently open file.

Notes: <channel no> is an integer expression whose value is 1,2,3 or 4. The number is associated with the file for as long as it is OPEN, and other disc I/O statements use the channel number to refer to the file.

Examples:

```
10 USER OPEN #2, "TEMP.DAT","O"

20 USER KILL #2
```

LINE INPUT #

Format: LINE INPUT #<channel no>,arguments

Purpose: To read an entire line from a disc data file to a string variable.

Notes: <channel no> is an integer expression whose value is 1,2,3 or 4. The number is associated with the file for as long as it is OPEN, and other disc I/O statements use the channel number to refer to the file.

The arguments are the string variable names that will be assigned to the items in the file.

LINE INPUT reads all characters in the file up to a carriage return/line feed. The next LINE INPUT reads all the characters up to the next carriage return/line feed. The carriage return/line feed sequence itself is skipped over.

N.B. If a line feed/carriage return is encountered, it is preserved.

LOAD

Format: LOAD <filename>

Purpose: To read a file from disc into memory

Notes: <filename> is a string expression that conforms to CP/M's requirements for filenames.

<filename> is the name that was used when the file was SAVEd. LOAD deletes all variables and program lines currently residing in memory before it LOADs the designated program.

Information may be passed between programs using their disc data files.

Examples:

```
10 USER LOAD "PROG.BAS"

USER LOAD "PROG.BAS"
```

OPEN

Format: OPEN #<channel no>,<filename>,<type>,(<reclen>)

Purpose: To allow Input/Output to a disc file.

Notes: A disc file must be OPENed before any disc I/O operation can be performed on that file.

OPEN determines the mode of access that will be used with the specified channel.

<channel no> is an integer expression whose value is 1,2,3 or 4. The number is associated with the file for as long as it is OPEN, and other disc I/O statements use the channel number to refer to the file.

<filename> is a string expression containing filenames.

<type> is a string expression whose first character is one of the following:

```
O = sequential output
I = sequential input
R = random input/output
```

<reclen> is a numeric expression which sets the record length for random files. Any values generated by this

expression are truncated to produce an integer.
<reclen> must be included for random files. Otherwise it should not be included in OPEN statements.

Examples:

```
10 USER OPEN #2, "PROG.DAT","O"

20 USER OPEN #3, "RDATA.DAT","R",128
```

PRINT #

Format: PRINT #<channel no>,<list of expressions>

Purpose: To write data to a sequential or random access disc file.

Notes: <channel no> is an integer expression whose value is 1,2,3 or 4. The number is associated with the file for as long as it is OPEN, and other disc I/O statements use the channel number to refer to the file.

The expressions in <list of expressions> are the numeric and/or string expressions that will be written to the file.

When PRINTing to a random file, a single string is specified. If the string is too long it will be truncated at the record length. If the string is too short the record will be padded out with zeros.

Example:

```
10 USER OPEN #1,"DATA","R",20
20 LET F$="  "
30 FOR X=1 TO 3
40 PRINT "RECORD NO ";X;INPUT ">";A$
50 USER REC #1,X
60 USER PRINT #1,A$+F$
70 NEXT
80 USER CLOSE #1
90 PRINT:PRINT:PRINT:PRINT
100 USER TYPE "DATA"
```

READ

Format: READ "<filename>",<start address>

Purpose: To READ a block of memory from disc.

Notes: <filename> is a string expression containing a name that conforms to CP/M's rules for disc filenames.

<start address> and <no of bytes> are in decimal and can be numeric expressions.

Example

```
10 USER READ "TEST.DAT",17000
20 USER READ "MEM.DAT",12*(4096)
```

ie read MEM.DAT into memory starting at #C000

REC #

Format: REC #<channel no>,<logical record no>

Purpose: Positions the pointer in the file at the record number specified. This only applies to random files.

Notes: <channel no> is an integer expression whose value is 1,2,3 or 4. The number is associated with the file for as long as it is OPEN, and other disc I/O statements use the channel number to refer to the file.

The next record INPUT or PRINTed after this command will be the one specified in <logical record no>.

Example: 10 USER REC #1,7

REN

Format: REN (string expression)=(string expression)

Purpose: To rename a file on Disc

Notes: REN uses the CP/M Filename convention.

Example: 10 USER REN "NEWNAME.DAT"="OLDNAME.DAT"

ROM 5

Format: ROM 5

Purpose: If the disc in the disc drive is changed, ROM 5 can be used to reset the system and read the names of the files on the new disc into memory.

Example: If a disc is changed without the use of a ROM 5 command, the system will still hold the directory from the previous disc, and will not be aware that the disc has been changed.

10 ROM 5

ROM 5

RUN

Format: RUN <filename>

Purpose: To load programs from disc and run them.

Notes: <filename> is a string expression that conforms to CP/M's requirements for filenames.

The program to be loaded and run must be preceded by four bytes.

The first two bytes of the program must be the address at which the program is to be loaded, and the second two bytes must be the size of the program. The four bytes themselves are not loaded into memory.

Normally written and saved Basic programs cannot use this command.

Example: ASSEM 10

```
        DW START                    ;START ADDRESS
        DW 16                       ;PROGRAM LENGTH
START:  RST 10
        DB #8C,13,10,"It works",13,10
        RET
```

(return to Basic)

If you are using an MTX500, type

```
USER WRITE "TEST.RUN",32775,20
```

If you are using an MTX512, type

```
USER WRITE "TEST.RUN",16391,20
```

then type

```
NEW

USER RUN "TEST.RUN"
```

Note: This code is location independent, so changing

DW START to DW #BFF0

will load the program at #BFF0, and will allow it to be called without destroying any current programs.

SAVE

Format: SAVE <filename>

Purpose: To write a program that is currently in memory to Disc.

Notes: <filename> is a string expression that conforms to CP/M's requirements for filenames.
If <filename> already exists, the existing file will be overwritten.
If BASIC is unable to save the file a "no space" error will be generated, and existing files retained.

Examples: 10 USER SAVE "PROG.BAS"

USER SAVE "PROG.BAS"

NB: Attempting to write to a R/O will result in a BDOS ERROR. Use ^C and ROM 5 to recover.

TYPE

Format: TYPE (string expression)

Purpose: To write a file to the screen

Notes: TYPE uses the CP/M Filename convention.

Example: 10 USER TYPE "NAMES.DAT"

<BREAK> terminates listing. The <PAGE> key functions normally to stop and start listing.

NB: Typing any file containing control characters, including NewWord or Basic files will cause screen errors.

WRITE

Format: WRITE "<filename>",<start address>,<no of bytes>

Purpose: To WRITE a block of memory to disc.

Notes: <filename> is a string expression containing a name that conforms to CP/M's rules for disc filenames.
<start address> and <no of bytes> are in decimal and can be numeric expressions.

Example

```
10 USER WRITE "TEST.DAT",17000,100

20 USER WRITE "MEM.DAT",10*(4096),100
ie save 100 bytes starting at #A000
```

NB: Attempting to to write to a R/O Disc will result in a BDOS ERROR. Use ^C and ROM 3 to recover.

ERRORS

Sometimes a message BDOS ERROR will occur.

This indicates that an error has occurred in the low-level disc operating system.

Common causes are:-

1. A file has not been closed.
2. A disc door is left open.
3. A disc is in the wrong way.
4. A disc has not been formatted.
5. A disc has been corrupted, or mistreated.
6. A disc drive had been selected that does not exist.

EXAMPLE PROGRAMS

The following BASIC programs are designed to illustrate the way in which Memotech Single Disc BASIC works, with particular reference to string handling.

```
10 REM TO READ AND WRITE A SEQUENTIAL FILE
20 USER OPEN#1,"NAMES.DAT","O"
30 INPUT "DO YOU WANT TO ENTER ANOTHER NAME? ";Y$
40 IF Y$<>"Y" AND Y$<>"y" THEN  GOTO 100
50 INPUT "TYPE IN A NAME:";N$
60 USER  PRINT #1,N$
70 CLS
80 GOTO 30
100 USER CLOSE#1
110 USER OPEN#1,"NAMES.DAT","I"
120 USER EOF#1,200
130 USER  INPUT #1,N$
140 PRINT N$
150 GOTO 120
200 USER CLOSE#1
```

```
10 REM TO READ AND WRITE A SEQUENTIAL FILE
20 USER OPEN#1,"NAMES.DAT","O"
30 INPUT "DO YOU WANT TO ENTER ANOTHER NAME? ";Y$
40 IF Y$<>"Y" AND Y$<>"y" THEN  GOTO 100
50 INPUT "TYPE IN A NAME:";N$
60 USER  PRINT #1,N$
70 CLS
80 GOTO 30
100 USER CLOSE#1
110 USER TYPE"NAMES.DAT"
```

```
10 REM TO KEEP A BACKUP COPY OF AN ALTERED FILE
11 REM NAMES.DAT IS THE MOST RECENT FILE
12 REM NAMES.BAK IS THE BACKUP FILE
13 REM NAMES.TMP IS A TEMPORARY FILE
20 USER OPEN#1,"NAMES.DAT","I"
30 USER OPEN#2,"NAMES.TMP","O"
40 USER EOF#1,100
50 USER  INPUT #1,N$
60 USER  PRINT #2,N$
70 GOTO 40
100 USER CLOSE#1
110 INPUT "DO YOU WANT TO ENTER ANOTHER NAME? ";Y$
120 IF Y$<>"Y" AND Y$<>"y" THEN  GOTO 200
130 INPUT "TYPE IN A NAME:";N$
140 USER  PRINT #2,N$
150 CLS
160 GOTO 110
200 USER CLOSE#2
210 USER OPEN#1,"NAMES.BAK","O"
220 USER KILL#1
230 USER REN"NAMES.BAK"="NAMES.DAT"
240 USER REN"NAMES.DAT"="NAMES.TMP"
250 USER TYPE"NAMES.DAT"
```

```
10 REM EXAMPLE OF RANDOM ACCESS FILES
15 REM
20 REM A$ IS A WORKING RECORD IN MEMORY
25 REM
30 REM EACH RECORD IS 100 CHARACTERS LONG
35 REM
40 REM ONLY ONE STRING CAN BE PRINTED OR INPUT
50 REM FROM A RANDOM FILE AT EACH TIME
55 REM
60 REM A$ IS A FIXED LENGTH ARRAY OF 100 CHARACTERS
65 REM IT IS INITIALISED TO ******
100 DIM A$(1,100)
101 FOR I=1 TO 100
102 LET A$(1,I)="*"
103 NEXT
110 PRINT A$(1)
115 REM
116 REM OPEN THE RANDOM FILE WITH RECORD LENGTH 100
117 REM
120 USER OPEN#1,"DATA.TXT","R",100
130 INPUT "ENTER NAME: ";N$
132 REM
133 REM FILL OUT N$ WITH *****
134 REM
135 IF N$="" THEN  GOTO 200
136 FOR I=LEN (N$)+1 TO 20
137 LET N$(I)="*"
138 NEXT
139 REM
140 INPUT "ENTER RECORD NUMBER";R
145 IF R=0 THEN  GOTO 300
149 REM ASSIGN N$ TO THE FIRST 20 CHARACTERS OF N$ TO A$
150 LET A$(1,1,20)=N$(1,20)
157 REM
158 REM SELECT RECORD  R
159 REM
160 USER REC#1,R
170 USER  PRINT #1,A$(1)
177 REM
178 REM PRINT THE ENTIRE ARRAY A$ TO DISC
179 REM
180 GOTO 130
200 INPUT "ENTER REC";I
205 IF R=0 THEN  GOTO 300
210 USER REC#1,I
220 USER  INPUT #1,A$(1)
230 PRINT A$(1)
240 GOTO 200
300 USER CLOSE#1
```

MEMOTECH
SINGLE DISC BASIC
UTILITY PROGRAMS

STAT

Format: USER STAT

The STAT command has three different modes.

1 To find the ammount of space left on the disc.
2 To find the size of a file.
3 To set a file to be read only.

1 Type: USER STAT

This mode gives the ammount of space on the disc.
e.g.

```
Space 100k
```

2 Type: USER STAT "FILE.EXT"

This mode gives the size of the file FILE.EXT in both the number of 128 Byte records and the size of the program in Kbytes. It also displays whether the file has been set to read only (RO), or can be written to (RW).
e.g.

```
USER STAT "ABC.BAS"

ABC.BAS 16K 124 Records RW
Space 20K
```

3 USER STAT "FILE.EXT",RO

This mode is used to protect a file by setting it to be read only (RO), or unprotect it by setting it to read/write (RW).
e.g.

```
USER STAT "ABC.BAS",RO
```

will produce on the screen:

```
ABC.BAS set to RO
```

FORMAT

Format: USER FORMAT

Before a new disc can be used it must be FORMATted.
The command writes information onto the disc so that the computer can keep a directory of files and know exactly where on the disc it has put them.

To FORMAT a disc, insert the system disc and type..

USER FORMAT

The computer will give the message

Ready to format
Insert Disc and
type <RET> to format disc
type any other key to abandon

If <RET> is pressed, the disc will be formatted. This will take about 40 secs. The message ..

WAIT....FORMATTING

will appear.

When the disc has been formatted, the computer will give the option to insert and format another disc.

SYSCOPY

Format: USER SYSCOPY

Before a disc can be used, it must be formatted using the FORMAT command. Then the disc system must be copied onto it using the SYSCOPY command.

When the SYSCOPY command is used, a disc should be inserted which already contains a system, ie a system disc. The computer loads the system from this disc into memory, ready for copying. The computer then gives the instruction to insert the destination disc onto which the system is to be copied.

e.g.

USER SYSCOPY

Insert Source Disc.....Press a key <COMPUTER READS SYSTEM>

Insert Destination Disc...Press a key <COMPUTER WRITES SYSTEM>

<RET> To continue, any other to quit.

If <RET> is pressed the system can be copied to another disc.

Note: Attempting to SYSCOPY a write protected disc will cause a BDOS error. To recover use Control C and then ROM 3.3.

COPY

Format: USER COPY "NEWFILE"="OLDFILE"

COPY is used to make a copy of a file.

The disc containing the file to be copied is called the source disc.
The disc to which the file is to be copied is called the destination disc.

Because the files may be very large, the COPY program copies 16K at a time from the source disc to the destination disc until the entire file is copied. At all times the computer displays which disc should be inserted.

e.g.
USER COPY "NEWFILE"="OLDFILE"

Insert Source Disc ..... Press a key (COMPUTER LOADS 16K)

Insert Destination Disc... Press a key (COMPUTER SAVES 16K)

This process is repeated until the file is copied.

If the break key is pressed during a copy, there will be a file called NEWFILE.$$$ containing as much of the file that has been copied up to that point.