The Cumana Disk System for the QL Computer

First Edition

A Cumana Publication

Published By:-

Cumana Limited, The Pines Trading Estate,
Broad Street, Guildford, Surrey, GU3 3BH

Parts of Chapter 3, 4 and 5 - Tony Tebby

Contents

Introduction

The Cumana Floppy Disk system for the QL microcomputer consists of interface electronics in a plug-in cartridge and a dual or single disk drive with mains power supply.

The cartridge is intended to be plugged into the Expansion slot of the QL Computer

The cartridge may also be supplied separately from the disk drives. Contact Cumana for the list of suitable alternative disk drives.

If any of the items mentioned above are missing from the package supplied, contact the supplier.

Please fill in and return the warranty registration card enclosed with the package.

Warning: This equipment must be earthed.

IMPORTANT: The wires in the mains lead for the disk drive are coloured in accordance with the following code:

GREEN AND YELLOW.....EARTH
BLUE.....NEUTRAL
BROWN.....LIVE

SEE DIAGRAM OVERLEAF.

The colours of the wires in the mains lead may not correspond with the colours or markings identifying the terminals in your plug, therefore proceed as follows:

The wire coloured green and yellow must be connected to the terminal marked with either the letter E (or the earth symbol) and/or coloured green (or green and yellow).

The wire coloured blue must be connected to the terminal marked with either the letter N, and/or coloured black (or blue).

The wire coloured brown must be connected to the terminal marked with either the letter L, and/or coloured red (or Brown).

If the socket outlet available to you is not suitable for the plug supplied, the plug should be cut cleanly off and disposed of immediately. The exposed mains wires cause a potential shock hazard if the moulded plug were to be plugged in.

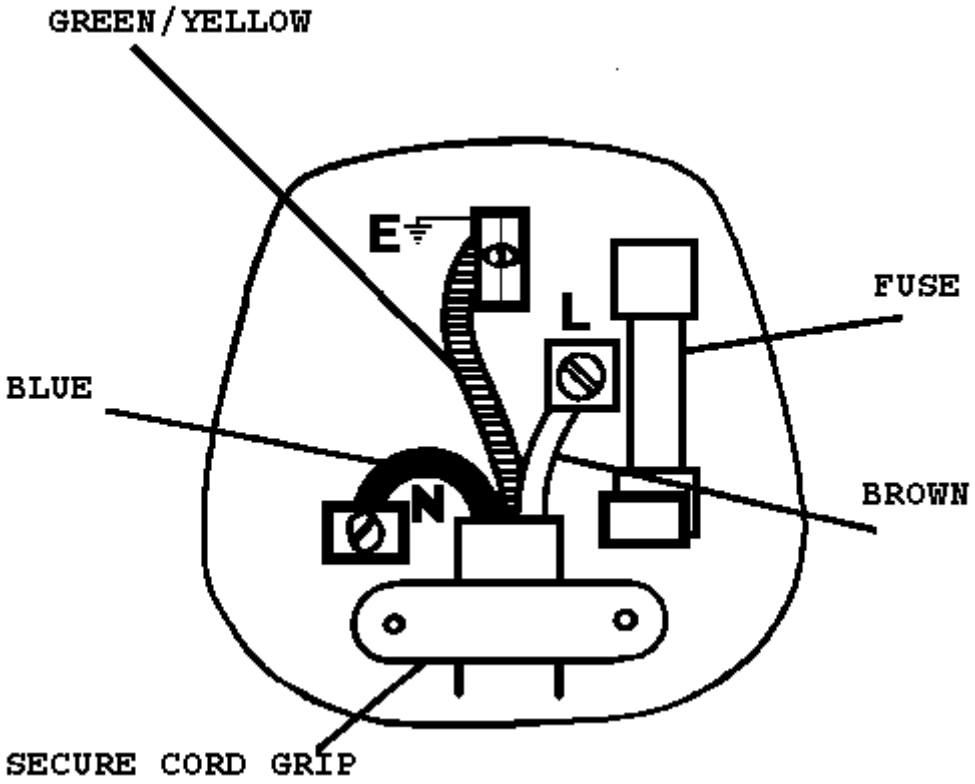
The moulded plug must only be used with the fuse and fuse carrier firmly in place. Plugs and fuse carriers from other manufacturers are not inter-changable. If the plug or fuse carrier should become damaged, or the fuse carrier lost, the plug should be replaced in accordance with the directions above, The plug MUST NOT be used without a suitable fuse carrier. Should the fuse blow, the fault must first be remedied and the fuse replaced with a 3 AMP fuse that is ASTA approved to BS1362. If in doubt consult a qualified electrician.

Warning

There are no user servicable parts inside a CUMANA disk drive. Before attempting to remove the lid be sure to isolate the unit by switching off at the mains and removing the plug from the main supply.

Red/Brown	To Live
Black/Blue	To Neutral
Green/Green&Yellow	To Earth

FIT WITH A 3 AMP FUSE To BS 1362



Under no circumstances is the earth to be removed as damage could result to both computer and floppy disk drive, as well as the possible danger of electrocution.

If in doubt please consult a fully qualified electrician.

Chapter 1

Connecting Up

IMPORTANT NOTICE

DO NOT ATTEMPT TO FIT THE DISK INTERFACE UNIT OR THE DISK DRIVES TO THE QL UNTIL YOU HAVE READ THE FOLLOWING INSTRUCTIONS:

Remove the power from the QL Computer and the disk drives before attempting to install the disk interface.

If you fail to observe this procedure then serious damage may result to the QL and disk interface,

Fitting the Interface

If you require to use the disk interface with other interface boards then you will need to use the QL Peripheral Expansion Unit. To install the disk interface into this unit you should refer to its own manual. However if you are just using the Cumana disk interface then proceed as follows.

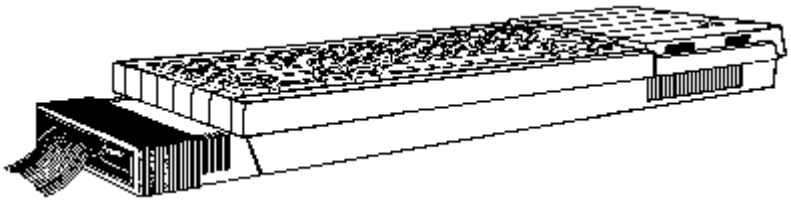


Fig 1.1 Diagram of QL Computer Drive and Disk Interface

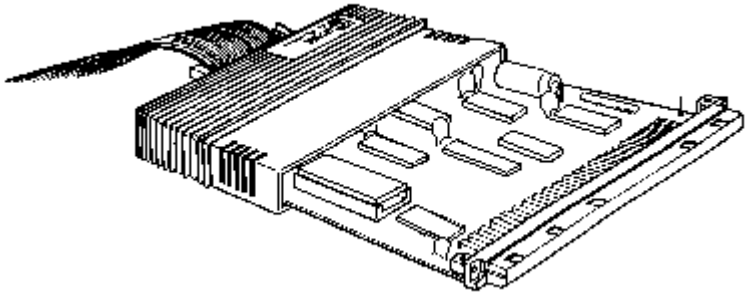


Fig 1.2 Diagram of disk Interface only showing Disk Drive connection to interface

On the left hand end of the QL, you will find a rectangular plastic cover with a small clip which covers the QL peripheral expansion slot. To remove this cover pull the clip away from the computer. The cover is sometimes a very tight fit, and may require some effort to remove.

As you will see, the Disk Interface Board has a cover over approximately half its length - the other half being a bare circuit board with a plastic connector at its end, Slot this end of the interface card into the end of the QL. The components on the board

should be face up, and it should slide in under the retaining slots on the QL.

Now push the Disk Interface firmly home - you should be able to feel when the card is firmly in place and be able to confirm that it is seated in a level position. See Fig. 1.1.

IMPORTANT: Always ensure the power is switched off before installing or removing the interface board from the QL or serious damage may result to your controller and/ or Computer system.

Ensure the QL and the Disk Drives are still NOT connected to the mains/power supply, The connecting lead supplied with the Disk Drives will already be correctly wired and tested and simply pushes into the socket protruding from the Disk Interface. See Fig. 1.2.

Note: Do not attempt to force the cable socket into the connector on the interface. If the socket refuses to go in, check for possible bent pins in the connector or incorrect orientation of the socket.

IMPORTANT: YOU MUST POWER-UP IN THE FOLLOWING SEQUENCE:

1. Display monitor or TV set,

2. Connect the Disk Drives to the power source (Some Cumana disk drives can be heard settling after being switched on).

3. QL Computer

If you do not follow this sequence for connecting the power, then damage may be caused to the QL and the Disk units.

Do not leave floppy disks in the drives while powering up. It may result in corrupting data on the disk.

After Power up of your system you should see the normal TV/Monitor Selection Screen with the message: Cumana Disk Interface V1.14 (C) 1984

You will then be asked to select function keys Fl or F2 to choose either Monitor or TV operation. Select as appropriate and then press the 'Enter' key. If TV/Monitor selection works normally then your QL and Disk Drives are ready for use,

Removing the Disk Interface

If you need to remove the Disk Drive(s) and Interface for any reason, first DISCONNECT THE QL FROM THE MAINS and the Disk Drive(s) from their power source, if you fail to do this you may damage your QL and Disk Interface unit. Then reverse the procedure for installation. Unplug the cable from the Disk Drive(s) to the Disk Interface and gently pull the Disk Interface unit from the expansion slot on the QL (Put it in its original packing for safe keeping). Lastly, replace the cover on the expansion slot of the QL - this may again need some pressure.

Chapter 2

Formatting

Formatting is a process which segments each track into a number of sectors and this has to be done before a new disk can be used by the computer. However you must be sure that the disk you are formatting does not contain any useful files or information as ultimately they would be over written and totally destroyed.

Refer to the text in Appendix B for a description of the QL format.

To format a disk, place a diskette in drive 1 and type the following:-

FORMAT FLP1_xxx <ENTER>.

Where FORMAT is the command, FLP1 is the device name and number, xxx is the name given to the disk which can be up to ten characters long, and <ENTER> means a press of the enter key on your keyboard.

The disk name can be useful for identifying disks quickly without studying the files in the directory e.g. PAYROLL, GAMES, ASSEMBLIES, etc.

After entering the command the computer and disk drive will format each track in turn and when it is complete the following message will be displayed on your TV set or monitor

1440/1440

The two sets of digits displayed on the screen refer to the number of useful sectors and total sectors on the disk respectively. These digits will vary in size depending on the capacity of the disk drive being used.

Note: If after formatting the two figures shown differ i.e. the left hand figure is smaller, then the format process was not entirely successful and some sectors were found to be bad, these sectors are marked on the disk and the computer would never use them.

However, it is a wise move to try and reformat the disk and if the same fault occurs then the diskette may be faulty. In this case try another diskette and discard the first as it may lead to problems in the future.

As the disk is used then the left hand figure will naturally decrease showing the number of sectors available for use, against the number of sectors formatted.

If you have more than one drive mechanism then you may wish to format on an alternative drive, Simply change the drive number i.e.

FORMAT FLP2_xxx <ENTER>.

This will format a diskette in drive 2.

Insertion

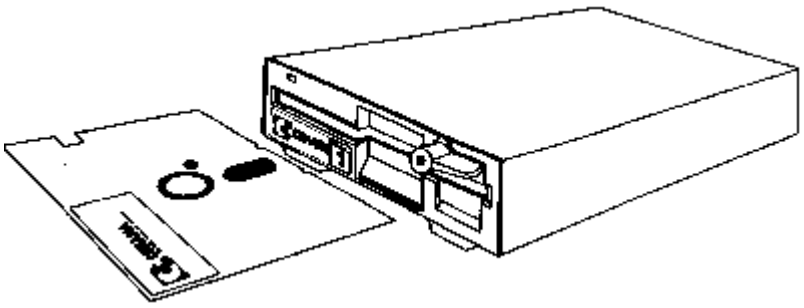


Fig. 2.1. Inserting a 5 1/4" floppy disk into the drive.

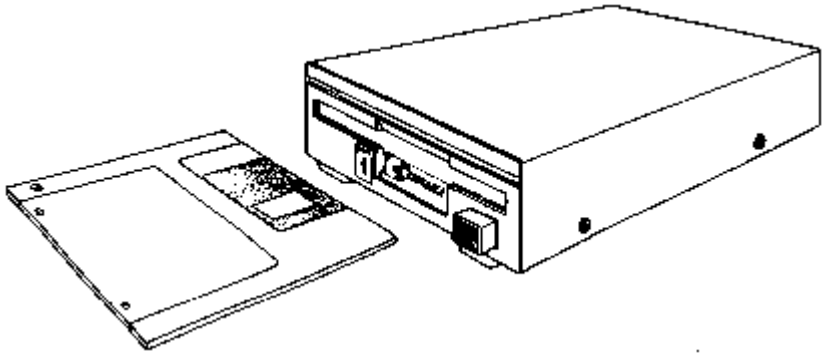


Fig. 2.2. Inserting a 3 1/2" floppy disk into the drive.

Depending on the type of drive you have purchased either:-

a) Push the diskette fully home and turn the locking lever clockwise as shown in Figure 2.1.

or

b) Push the diskette fully home until a click is heard as shown in Figure 2.2.

Chapter 3

Making a Backup

Because diskettes are exposed to every day environments there is a chance that one will get damaged or corrupted. It is therefore advisable to ensure you have a copy of all your important files.

In order that you may do this, a command has been supplied that will make a complete physical backup copy of any diskette you choose.

The command line is as follows:-

WCOPY FLP1 to FLP2 <ENTER>

Will copy the entire contents of the disk in drive 1 to a formatted disk in drive 2.

(Example PSION Program QUILL)

(1) Insert the Quill cartridge into Microdrive 1 (mdvl)
    Insert a formatted disk into Disk Drive 1 (flp1)

Transferring software from cartridge to disk.

(2) Type WCOPY MDVl TO FLPl and enter. Microdrive 1 and disk drive 1 should now run briefly.
    You will see displayed mdvl_clone TO flpl_clone..Y/N/A/Q?
    Now type A to copy this file and all the rest.
    Each file will be displayed as it is copied.
    When complete you will see the flashing cursor.

You will now have a copy of QUILL on disk, if you type CLS and enter, DIR flpl and enter, you will see this verified on the screen. Because the disk file BOOT is an exact copy of the Microdrive file, the computer will still be looking for mdv.

Before loading the program type FLP_USE and enter
To load and run the program type LRUN MDV1_BOOT and enter
The command FLP_USE can be entered after you first switch on the computer, it will then remain valid until you switch off.

Note: See Chapter 4 for full explanation of the FLP USE ctommand.

Chapter 4

Disk Filing System

The QL computer is delivered with two 'mass storage' devices: the Microdrives. These devices have the same function as the floppy disks on more expensive personal computerse being designed for the permanent storage of programs and data. Other devices which behave in the same way as Microdrives (such as floppy or hard disks) may be added to the QL 'transparently'. This means that QDOS will ensure that a program does not need to 'know' where its data is stored. A Microdrive looks, to a program, exactly the same as a floppy disk. This 'device independence' is a built in characteristic of the QDOS operating system.

The simplest way of using a floppy disk system on the QL is to copy all programs and data to floppy disks, and either add the command 'FLP_USE MDV' to all BOOT files, or type this command at the start of a session on the QL. The effect of this command is to make the floppy disks pretend to be rather large and fast Microdrives.

For example, a modified BOOT file for executing the PSION program Quill could look like:

```
100 FLP_USE mdv
110 CLOSE #1: CLOSE #2
120 EXEC_W mdvl_quill

....
....
```

On the other hand, it is just as easy to use the floppy disks without changing the name. All the filing system commands described in the 'Microdrives' section of the QL Concept Reference Guide will work with floppy disks, provided the filenames start with 'FLP' instead of 'MDV'.

FORMAT flp1_diskl	formats a new floppy disk in drive 1
DIR flp1	directory listing of floppy disk 1
SAVE flp1_myprog	save the current SuperBASIC 'myprog' in floppy disk 1
OPEN_NEW #3, flp2_data	creates and opens a new file 'data' in floppy disk 2
COPY mdv1_x TO flpl_x	copies file x from Microdrive 1 to floppy disk 1
LOAD flpl_x	loads file x from disk drive 1 into the computer
DELETE flp_x	deletes file x from floppy disk 1

Note: Delete is a command already available under QDOS for use with microdrives.

Floppy Disk Compatibility

The QJUMP Floppy Disk driver software provides an easy upgrade path from Microdrives to floppy disk speed and storage capacity, It not only provides all the built-in Microdrive filing system operations, but includes all the extended filing system operations provided in the Sinclair QL Toolkit for Microdrives. This allows all the SuperBASIC extensions provided in the QL Toolkit (e.g. FOP_OVER, RENAME etc.) to be used with the floppy disks

OPEN OVERWRITE Trap #2, DO=l, D3=3

This variant of the OPEN call opens a file for write/read whether it exists or not. The file is truncated to zero length before use.

RENAME Trap #3, DO=4A, Al points to new name

This call renames a file. The name should include the drive name (e,g. FLPl_NEW_NAME).

TRUNCATE Trap #3, DO=4B

This call truncates a file to the current byte position.

In addition the FS.FLUSH call fora file, not only flushes all the file buffers, but, unlike the Microdrive driver, updates the map and the directory, This means that a new file can be created, and if it is flushed, then in the event of the QL being turned off or reset before the file is closed, then all of the file (up to the point where it was last flushed), is readable. In effect a FLUSH call is just the same as a CLOSE call, except that the file remains open and the file pointer remains unchanged.

Auto-boot

If there is a disk in drive 1 when the QL is turned on (this may be risky with some makes of floppy disk drive, particularly those with permanently loaded heads) or reset (this should be safe with all drives), then QL will boot from the disk in drive 1, otherwise the QL will boot from Microdrive 1 as usual.

When a 'directory devicee such as a floppy diske is accessed for the first time, QDOS will allocate a block of memory for the device. In the case of a floppy disk, the Sinclair standard format requires a block of memory about 1.6 Kilobytes long. This is rather larger than the Microdrive block which is only about 0.6 kilobytes long. The auto-boot procedure used ensures that ifthere is no disk in drive 1 when the QL is resete then the 1.6 kilobyte block for disk drive 1 will be allocated. Programs that are too large to execute when floppy disks are being used, should still execute from microdrives.

Microdrive Emulation

The standard driver also includes a SuperBASIC procedure FLP_USE to change the name of the floppy disk driver.

FLP_USE mdv or FLP_USE 'mdv'

resets the name of the floppy disk driver to 'mdv', so that all subsequent open calls for Microdrives will use the floppy disks instead. Thus the commands

```
FLP_USE mdv

....
....
OPEN #3, mdv_myfile
```

will actually open the file 'myfile' on floppy disk 1, rather than trying to open a file on Microdrive 1

Any three letters may be used as a new device name, in particular

```
FLP_USE flp
```

will reset the drive to its normal state.

Note: If you use the Microdrive Emulation, you must first use the reset command FLP USE MDV before the MDV command may be used again.

Floppy Disk Options

There are three parameters of the floppy disk system which are available as user options.

The security level is selectable to allow a user to choose higher speed of access at the cost of reduced immunity to erroneous disk swapping. There are three levels of security, the lowest level still being at least as secure as common disk based operating systems (e.g. MSDOS and CPM).

A user may specify the time taken for the disk drive motor to get the disk speed to within the specification.

A user may specify the number of tracks to be formatted on a disk.

These parameters are specified by a single command:

```
FLP_OPT security_level (,start_up_time (, nr_of_tracks))
```

Security

The Microdrive filing system is unusual in that, although the data is stored in 'sectors' in just the same way as on a floppy disk, each sector holds information which identifies the cartridge. When a cartridge is changed the filing system will recognise the change the next time any access is made to Microdrive. Standard floppy disk formats do not allow this type of security, so the format used for QL floppy disks includes identifying information in Track 0 Sector 1 of the disk. Clearly if this were checked every time any access were made to the disk, then the floppy disk system would be very slow indeed, Security, in the context of this user option, is the extent to which the floppy disk system may be abused by changing disks, while they are in use, without destroying data stored on the disks,

There are four operations which affect the security: the first is the operation to check if the disk has been changed, the second is the operation to flush the slave blocks, the third is the operation to update the map and the fourth is the operation to update the directory.

In these definitions, the term 'the drive has stopped' is usually taken to mean that the motors have stopped and no drive select light is visible. On some floppy disk systems, if an attempt is made to access a drive which has no disk in place, or has the door open, then the motors will continue to run. If a drive select light is still on, then the motors may be stopped by inserting a disk and closing the door. In any case, the drives are deemed to be stopped if 5 seconds have passed without a disk access.

Security Level 0

The disk is only checked when a file is opened and the drive has stopped since the last time it was checked and there are no files already open on the drive.

The map is only updated after a file is closed (or flushed) when half a second has elapsed without any other disk operation.

At this lowest level of security, confusion or loss of data can be expected if a disk is changed while there are still files open or the motor is running.

Security Level 1

The disk is checked when a file is opened, or data or the map is to be written, and the drive has stopped since the last time it was checked.

The map is only updated after a file is closed (or flushed) when half a second has elapsed since the previous disk operation.

At this level of security, disks should only be changed while the motor is stopped (all select lights off).If a disk is changed while there are files open, then read operations will be confused but any write operations will be aborted. This should maintain the integrity of the data on the disk.

Security Level 2

The disk is checked whenever a file is opened or whenever the map or data to be read from or written to the disk and the drive has stopped since the last time the disk was checked.

The map and directory are updated and the buffers are flushed immediately after a file is closed, or after an FS.FLUSH call.

This is the default security level and data should be quite secure unless a disk is changed while the motors are running,

Security System Errors

There are two error messages which may be written to the screen by the floppy disk filing system. These are in the form of the disk name followed by the message itself. The first message indicates that an attempt to read or write a sector on the disk has failed:

disk - name read/write failed

The second message indicates that a disk has been changed while it is still in use:

disk_name files still open

If the floppy disk system attempts to write to a disk which has been changed, then you may get both messages indicating that the attempt to write the data has been aborted, and that files were still open when the disk was changed.

Start Up Time

The floppy disk system will always try to read data from a disk as soon as it can. However, to preserve the data integrity of the disk, write operations are held up until the disk has been 'run up' for long enough for the speed to be stable. As a default this is set to 0.6 second which is more than enough for most modern drives, The start_up_time parameter is in 20 millisecond units, so the default value is 30. A value of 13 (260 milliseconds) is adequate for the most recent direct drive 3.5 inch drives, while some older drives may require a value of about 60 (1.2 seconds). A value of 90 (1.8 seconds) or more may cause problems with some disk systems, as the motors may stop automatically before the start up time has elapsed!

Number of Tracks

The QL format for disks allows the number of tracks on a disk to be read from the disk itself. However, the number of tracks must be determined when a disk is to be formatted. Normally the disk system will do this itself by checking if there are at least 55 tracks on a disk. If there are, then there are assumed to be 80 tracks, otherwise it is assumed that there are 40 tracks. This internal check may be overwritten, allowing 37 track and 75 track drives to be formatted as well as saving possible wear or damage to a 40 track drive when seeking track 55 (somewhere in the middle of the jacket).

Direct Sector Read/Write

The software includes provision for reading sectors of a disk using direct addressing. To do this a special file is opened on the disk. The name is

```
FLP1_*Dsd  where s is the sector length  0 = 128 bytes
                                         1 = 256 bytes
                                         2 = 512 bytes
                                         3 = 1024 bytes

           and d is the density          S = single (FM)
                                         D = double (MFM)
```

When this file is open, no other file may be open on the drive, The only IO calls supported for this type of file are IO.FSTRG, IO.SSTRG IO.POSAB and IO.POSRE, to read or write complete sectors or to set the position. The parameter (Dl) to the POSRE call is ignored, but the current position is returned. Reading or writing a sector does not change the file position.

If the attempt to read or write a sector fails, D0 will be returned as a standard error message pointer (read/write failed).

The position is a composite of the required sector, side and track:

sector number + side * 256 + track * 65536

To ensure compatibility with string IO the length specified in the SSTRG and FSTRG calls may be one of three values:

sector length	the complete sector is read or written
2	returns the sector length (IO.FSTRG) ignored (IO.SSTRG)
2 + sector len.	returns the sector length followed by the sector (IO.FSTRG) skips the first two bytes, and writes the rest to the sector (IO,SSTRG)

This variety enables sectors to be read and written in SuperBASIC using the normal string IO in the QL Toolkit, as well as by assembler programs. For example, sector 1 of side 1 on track 2 may be read into the string A$ using the following command:

```
GET #n \1+256+2*65536, a$
```

When using the direct sector read/write calls for a 40 track disc in an 80 track drive, the track number should be doubled. Seek errors will not be detected. If a read/write error is returned from a direct sector read/write call, then it will be safest to make another call to read from track zero. Calls to read from or write to track zero will cause a 'restore' rather than a seek, and will thus reset the drive to a known state.

Disk Drive Specifications

It is a requirement that disk drives used with this version of the disk driver should be set to have the motor on when provided with a 'motor on' signal and there is a disk in the drive. Drives which turn the motor off when the drive is not selected will not give reliable service.

The disk driver will automatically adjust itself to use any mixture of disk drives. 40 or 80 track, single or double sided. In addition it will adjust itself to use slow step rate drives. Disks need not have been formatted and written on the same specification drive as a drive being used to read them.

Compatibility chart

Disk format -->	40T	40T+40T	80T	80T+80T
Drive				
40T	C	?	X	X
40T+40T	C	C	X	X
80T	R	?	C	?
80T+80T	R	R	C	C

C = compatible
R = compatible (read only)
X = incompatible
? = incompatible but may not be detected correctly on some types of drive

The format procedure automatically checks the drive specification and will format the drive in an appropriate manner. Note that 40 track drives which do not have an end stop, or which would suffer damage

when stepped beyond the 40th track (to track 55) should not be formatted unless the number of tracks has been specified in an FLP_OPT command, It is possible to force the disk driver to format a disk as single sided on a double sided drive by making the 11th character (it is invisible) of the disk name an asterisk: e.g.

FORMAT 'FLPl_DISK_NAME *'

Chapter 5

Additional Commands

Extensions to SuperBASIC

The Cumana disk system incorporates additional extensions to the SuperBASIC interpreter. Where these extensions are the same as extensions in the Sinclair QL Toolkit, the routines are functionally identical. There is thus no conflict between The QL Toolkit and the floppy disk software. The command:

FLP_EXT (no parameters)

is used to link in most of the extensions, This command may be included in a BOOT file or typed on the keyboard at any time. Repeated use of this command will inevitably use up space in the QL's memory.

If some of the extensions have been referred to before the FLP_EXT command is used, then the NEW command may have to be typed before these extensions will become available.

The EXTRAS command

To use the EXTRAS command simply type:

EXTRAS

and press the ENTER key. This command will list on the screen all other additional commands (Extensions) provided by the Cumana Disk Interface Unit, together with any commands loaded into RAM at power up (boot).

BGET BPUT GET PUT FPOS - Direct IO

In QDOS, files appear as a continuous stream of bytes. On directory devices (Microdrives, hard disks etc,) the file pointer can be set to any position in a file. This provides 'direct access' to any data stored in the file. Access implies both read access and, if the file is not open for read only (OPEN_IN from SuperBASIC, IO.SHARE in QDOS), write access. Parts of a file as small as a byte may be read from, or written to any position within a file. QDOS does not impose any fixed record structures upon files: applications may provide these if they wish.

Procedures are provided for accessing single bytes, integers, floating point numbers and strings. There is also a function for finding the current file position.

The general form of the direct I/O commands is:

command #n [\pointer] {,item)
or
It is usual (although not essential - the default is #1) to give a channel number for the direct I/O commands. If the pointer is given, the file position is set before processing the list of I/O items; if the pointer is a floating point variable rather than an expression, then, when all items have been read from or written to the file, the pointer is updated to the current file position.

Byte I/O

```
BGET #n [\pointer] {,item)
BPUT #n [\pointer] {,item)
```

BGET gets 0 or more bytes from the channel, BPUT puts 0 or more bytes into the channel. For BGET, each item must be floating point or integer variable; for each variable, a byte is fetched from the channel. For BPUT, each item must evaluate to an integer between 0 and 255- for each item, a byte is sent to the output channel,

For example the statements

```
abcd=2.6
zz%=243
BPUT #3, abcd+1, '12', zz%
```

will put the byte values 4, 12 and 243 after the current file position.

Unformatted I/O

It is possible to put or get values in their internal form. The PRINT and INPUT commands of SuperBASIC handle formatted IO, whereas the direct I/O routines GET and PUT handle unformatted I/O. For example, if the value 1.5 is PRINTed the byte values 49 ('1'), 46 ('.') and 53 ('5') are sent to the output channel. Internally, however, the number 1,5 is represented by 6 bytes (as are all other floating point numbers)- These six bytes have the value 08 01 60 00 00 00 (in hexadecimal). If the value is PUT, these 6 bytes are sent to the output channel.

The internal form of an integer is 2 bytes (most significant byte first). The internal form of a floating point number is a 2 byte exponent to base 2 (offset by hex 81 F), followed by a 4 byte mantissa, normalised so that the most significant bits (bits 31 and 30) are different. The internal form of a string is a 2 byte positive integer, holding the number of characters in the string, followed by the characters.

```
GET #n [\pointer] {,item)
PUT #n [\pointer] {,item)
```

GET gets data in internal format from the channel, PUT puts data in internal format into the channel, For GET, each item must be a integer, floating point, or string variable. Each item should match the type of the next data item from the channel, For PUT the type of data, put into the channel, is the type of the item in the parameter list. The commands

```
fpoint=54
...
wally%=42: salary=78000: name$='Smith'
PUT #3\fpoint, wally%, salary, name$
```

will position the file, open on #, to the 54th byte, and put 2 bytes (integer 42), 6 bytes (floating point 78000), 2 bytes (integer 5) and the 5 characters 'Smith'. Fpoint will be set to 69 (54+2+6+2+5).

For variables or array elements the type is self evident, while for expressions there are some tricks which can be used to force the type:

```
.... + 0     will force floating point type;
.... & "     will force string type;
.... || 0    will force integer type,

xyz$='ab258.z'

BPUT #3\37,xyz$(3 to 5) || 0
```

will position the file opened on channel #3 to the 37th byte and then will put the integer 258 on the file in the form of 2 bytes (value 1 and 2, i.e. 1*256+2).

Provided no attempt is made to set a file position, the direct I/O routines can be used to send unformatted data to devices which are not part of the file system, If, for example, a channel is opened to an Epson compatible printer (channel #3) then the printer may put into condensed underline mode by either

```
BPUT #3,15,27,45,1
or
PRINT #3,chr$(l5);chr$(27);'-';chr$(1);
```

There is one function to assist in direct access I/O: FPOS returns the current file position for a channel. The syntax is:

```
FPOS (#n)
```

For example:

```
PUT #4\102,valuel,value2
ptr = FPOS (#4)
```

will set 'ptr' to 114 (=102+6+6).

The file pointer can be set by using any of GET BGET, PUT or BPUT with no items to be got or put. If an attempt is made to put the file pointer beyond the end of file, the file pointer will be set to the end of file and no error will be returned. Note that setting the file pointer does not mean that the required part of the file is actually in a buffer, but that the required part of the file is being fetched. In this way, it is possible for an application to control prefetch of parts of a file where the device driver is capable of prefetching.

CLOCK - Resident Clock

```
CLOCK [#channel]              default clock, 2 rows of 10 chars
CLOCK [#channel,] string      user defined clock
```

CLOCK is a procedure to set up a resident digital clock. If no window is specified, then a default window is set up in the top RHS of the monitor mode default channel 0. This window is 60 by 20 pixels and is only suitable for four colour mode. The clock may be invoked to execute within a window set up by BASIC. In this case the clock job will be removed when the window is closed.

The string is used to define the characters written to the clock window: any character may be written except $ or %. If a dollar sign is found in the string then the next character is checked and

$d or $D will insert the three characters of the day of week,
$m or $M will insert the three characters of the month.

If a percentage sign is found then

%y or %Y will insert the two digit year
%d or %D will insert the two digit day of month
%h or %H will insert the two digit hour
%m or %M will insert the two digit minute
%s or %S will insert the two digit second

The default string is '$d %d $m %h:%m:%s' a newline should be forced by padding out a line with spaces until the right hand margin of the window is reached.

Example:

```
MODE 8
OPEN #6,'scr_156x10a32x16'
INK #6,0: PAPER #6,4
CLOCK #6,'QL time %h:%m'
```

DATA_USE - Data File Default

Default directories may be set for use with many Toolkit commands.

DATA_USE directory_name

If the directory_name supplied does not end with '_', '_' will be appended to directory name. The directory name can be more detailed than just a device name, For example:

DATA_USE flpl_project5_library

```
WDIR
ferr=FOP_NEW (#3,fred)
```

will produce a directory listing of all filenames starting with 'flpl_project5_library' and then open a new file called 'flpl_project5_library_fred'. The default set by this command is optional and is only used if the name supplied to a Toolkit command is not a valid file or device name. Thus:

ferr=FOP_NEW (#3,flp2_fred)

will open file 'flp2_fred' (not 'flpl_project 5_library_flp2_fred'!)

File Open Functions - FOPEN FOP_IN FOP_NEW FOP_OVER FOP_DIR

This is a set of functions for opening files. These functions differ from the OPEN procedures in ROM in two ways: firstly, if a file system error occurs (e.g. 'not found' or 'already exists') these functions return the error code and continue; secondly the functions use the DATA_USE directory default

```
FOPEN (#3,name)      open for read/write
FOP_IN (#3,name)     open for read only
FOP_NEW (#3,name)    open a new file
FOP_OVER (#3,name)   open a new file, or overwrite old file
FOP_DIR (#3,name)    open a directory
```

Directory entries may be read using GET to get information. Each entry is 64 bytes long, the length of the file is at the start of the entry, there is a standard string starting at the 14th byte of the entry giving the filename and there is the update date as a long integer starting at the 56th byte,

Example of file Open

A file may be opened for read only with an optional extension using the following code

```
ferr=FOP_IN (#3,name$&'_ASM')      :REMark try to open ASM file
IF ferr=-7: ferr=FOP_IN (#3,name$) :REMark ERR.NF, try no_ASM
```

File Enquiry Functions - FLEN FTYP FDAT

There are three functions to extract information from the header of a file. Note that in current versions of the Microdrive handler, the header is only updated on an FS.HEADS call or on closing the file, the QJUMP Floppy Disk Driver also updates the header on a call to flush the disk buffers. This means that the file length read from the header will usually be the file length as it was when the file was opened.

If a file is being extended, the file length can be found by using the FPOS function to find the current file position. (If necessary the file pointer can be set to the end of file by the command GET #n\ 999999)

```
FLEN (#n)    returns the file length,
FTYP (#n)    returns the file type (O=normal 1=EXEC),
FDAT (#n)    returns the data space for EXEC files.
```

JOBS AJOB SPJOB RJOB - Job Control

As QDOS is a multitasking operating system, it is possible to have, at one time, in the QL a number of competing or co-operating jobs. Jobs compete for resources in line with their priority, and they may co-operate using pipes or shared memory to communicate. The basic attributes of a job are its priority and its position within the tree of jobs (ownership). A job is identified by two numbers: one is the job number which is an index into the table of jobs, and the other is a tag which is used to identify a particular job so that it cannot be confused with a previous job occupying the same position in the job table. Within QDOS the two numbers are combined into the job ID which is job number + tag*65536. For these job control routines, where job-id is a parameter of one of the job control routines, it may be given as either a single number (the job ID, as returned from OJOB or NXJOB of the QL Toolkit) or as a pair of numbers (job number, job tag). Thus the single parameter 65538 (2+1 *65536) is equivalent to the two parameters 2,1.

JOBS is a command to list all the jobs running in the QL at the time. If there are more jobs in the machine than can be listed in the

output window, the procedure will freeze the screen (CTRL F5) when it is full. The procedure may fail if jobs are removed from the QL while the procedure is listing them. The following information is given for each job:

the job number
the job tag
the job's owner job number
a flag 'S' if the job is suspended
the job priority
the job (or program) name.

The syntax of the command is:

JOBS [#n] where #n is the channel for the listing.

There are three procedures for controlling Jobs in the QL:

```
AJOB job_id,priority      activates a job;
SPJOB job_id,priority     sets a job's priority;
RJOB job_id,error_code    removes a job from the QL.
```

If there is a job waiting for the completion of a job removed by RJOB, it will be released with DO set to error_code,

Listing Extensions

EXTRAS [#channel_number]

The SuperBASIC interpreter is extendable. The procedure EXTRAS may be used to list the extra procedures and functions linked into the interpreter, EXTRAS will freeze the screen (CTRL F5) when the output window has been filled.

RENAME and TRUNCATE

The RENAME and TRUNCATE procedures operate on files on floppy disks or, if the Sinclair QL Toolkit has been added, on microdrives. If either of these procedures are used on a standard QL to operate on a microdrive file, the result will be a 'bad parameter'.

```
RENAME old,new       rename a file, the DATA_USE default
                     directory is used for both filenames.
TRUNCATE #n          truncate the file open on #n to the
                     current file position.
```

VIEW - Examining a File

VIEW is procedure intended to allow a file to be examined in a window on the QL display.

VIEW [#n,]name view a file: lines are truncated to fit in the window and, when the window full, CTRL F5 is generated.

SPL SPL_USE - File Spooler

The SPL command sets up a job to copy a file. Only the source need be given: the destination may be defaulted, The source file has the default set up by the DATA_USE command. As supplied, the default destination is SER. The SuperBASIC interpreter will continue after the Job has been set up, with the file being copied in the

background. SPL differs from COPY not only in that it operates as a job in the background, but also in the handling of file headers. The COPY procedure copies both the file and its header: to copy a file to a device like a printer, the variant COPY_N is used to copy without the header.

SPL will, however, not copy the header from an ordinary data file, but it will copy the header of a file which is one of the special types (e.g, executable program file). Furthermore, when using SPL to copy from file to file, if the destination file already exists, then it will be overwritten.

The command syntax is

SPL source_file [TO destination_file]

The source and destination files may be given as names, or as a SuperBASIC channel number (e,g, #3).

The default set by the DATA-USE command is used to find the source file, and there is a special command, SPL USE, to set the default destination. The default destination device or directory may be up to 32 characters long.

SPL_USE device_name / directory_name

A device_name does not end in '_': a directory_name must end in '_'.

If the SPL command is given with only one parameter (the source filename) the output file (or device) will be derived from the current default set by SPL_USE as follows:

1) directory name&source filename
or
2) device name

If the SPL command is given with 2 parameters the output file or device will be derived as follows:

1) destination_filename
or
2) directory_name&destination_filename

WCOPY Wild Card Copying

WCOPY source_wild_name TO destination_wild_name
WCOPY source_wild_name, destination_wild_name

When using WCOPY each source and destination filename is written to the chosen channel, and the user is requested to press one of:

Y (yes) copy this file
N (no) do not copy this file
Q (quit) do not copy this or any more files
A (all) copy this and all the next matching files.

If the destination file already exists, the user is requested to press one of:

Y (yes) copy this file, overwriting and the old file

```
N (no)      do not copy this file
Q (quit)    do not copy this or any more files
A (all)     overwrite the old file, and overwrite any other
            files requested to be copied.
```

WCOPY may be used to copy whole directories. The destination name is made up from the actual source file name and the destination wild name. If a missing section of the source wild name is matched by a missing section of the destination wild name, then that part of the actual source file name will be used as the corresponding part of the actual destination name. Otherwise the actual destination file name is taken from the destination wild name, If there are more sections in the destination wild name than in the source wild name, then these extra sections will be inserted after the drive name, and vice versa.

For example, if the default data directory is mdv2_, then

```
WCOPY fred,mog                  would copy

mdv2_fred                       to mdv2_mog
mdv2_freda_list                 to mdv2_moga_list

WCOPY _fred, _mog               would copy

mdv2_fred                       to mdv2_mog
mdv2_freda_list                 to mdv2_moga_list
mdv2_old_fred                   to mdv2_old_mog
mdv2_old_freda_list             to mdv2_old_moga_list

WCOPY _list, old_ _list         would copy

mdv2_jo_list                    to mdv2_old_jo_list
mdv2_freda_list                 to mdv2_old_freda_list

WCOPY old_ _list, mdv1_ _list   would copy

mdv2_old_jo_list                to mdvl_jo_list
mdv2_old_freda_list             to mdvl_freda_list
```

SPL will often be used to copy files in the background, but it can be used as a true spooler when used with the default output device. In this case, if the output device is in use, the SPL job will suspend itself until the device is available.

SPL Examples

```
SPL myfile                      using the supplied defaults this will
                                spool MDV2_MYFILE to SER

SPL flp2_demo_myfile TO ser2    the file FLP2_DEMO_MYFILE will be
                                spooled to SER2

DATA_USE flp2_demo              this will also spool the file
                                FLP2_DEMO_MYFILE to SER2
...
SPL myfile

SPL mdv2_myfile,mdv1_myfile     does the obvious

SPL_USE mdv1_                   using the supplied DAT A USE default,
...                             this will also spool mdv2_ myfile to
```

	mdvl_myfile
SPL myfile	
SPL myfile to #3	will spool myfile to the file or device already opened as #3

VIEW - Examining a File

VIEW is procedure intended to allow a file to be examined in a window on the QL display.

VIEW [#n,] name	view a file: lines are truncated to fit in the window and, when the window is full, CTRL F5 is generated.

Wild Card Commands

There is a set of directory maintenance commands using a 'wild card' definition of the file name (based on the DATA USE default irectory).

STAT [#n,][name]	print medium name, number of free sectors, total number of sectors.
WDIR [#n,] [wild_name]	list directory, generates CTRL F5 when the window is full.
WSTAT [#n,] [wild_name]	list file name, length and last update date, generates CTRL F5 when the window is full.
WDEL [#n,] [wild_name]	delete files (requests confirmation).
WDEL_F [wild_name]	delete files (forced).

When using WDEL, each filename is written to the chosen channel, and the user is requested to press one of the keys:

Y (yes)	delete this file
N (no)	do not delete this file
Q (quit)	do not delete this or any of the next files
A (all)	delete this and all the next matching files

The wild - name in these procedures may refer to more than one file. To do this file names are divided into sections (flp2_fred_bin has three sections) and a wild name may have missing sections (e.g. flp2_ _old_ _list has one missing section). All those files whose names have sections matching the sections in the wild name are referenced by the commands, In the following examples, flp2_ is assumed to be the default data directory.

Wild name	Typical matching files
fred	flp2_fred flp2_freda_list
_fred	flp2_fred flp2_freda_list flp2_old_fred flp2_old_freda_list
flp1_old_ _list	flpl_old_jo_list flpl_old_freda_list

Appendix A

Data Storage

There are various devices in which data can be stored, and in general they are referred to as memory. Memory size is normally measured in Kilobytes of information, where one byte is eight bits and a kilobyte is 1024 bytes.

The QL Computer has 128 Kilobytes of user memory known as RAM or Random Access Memory, and as the name suggests, it can be used in a number of different ways. The RAM normally holds the current program and its variables. When a new program is needed it can be loaded into RAM that was previously occupied by the old program, thus overwriting it.

RAM can only hold information whilst the power is switched on. When the computer is switched off any program or data contained in its RAM will be lost.

The QL also contains a certain amount of ROM or Read Only Memory. This memory contains the program or operating system that the QL needs to communicate and interface with the outside world It is normally known as firmware, and unlike RAM, it cannot be overwritten and it will not be lost when the machine is turned off.

Other types of memory associated with computers are those known as media storage peripherals: disk drives, magnetic tape drives, punched tape and card machines, etc. These devices are normally used for long-term data retention. The most popular devices use magnetic media for data storage.

Data stored within a computer can be saved onto magnetic media devices before the machine is turned off or before another program is run. Similarly, data stored on a peripheral can be loaded into a computer for manipulation.

Floppy disk drives are a popular type of magnetic media peripheral. Floppy disk drives use diskettes, also known as floppy disks, as the data storage media.

There are many advantages to using Disk Drives compared with Microdrives, the more obvious being the fact that Disk Drives are faster and more convenient to use. A useful feature of the Cumana QL Disk Interface unit is that it allows you to copy files (data and programs) from Microdrive cartridges to Floppy Disks and vice versa. This means that all the existing software that you have been using on your QL can be transferred to floppy disk, (Including the PSION suite of packages).

The Cumana Disc Interface unit can be used with three types of disc drive using different types of floppy disc:

3 inch, 3 1/2 and 5 1/4 inch.

Each of these can be either single or double sided, and 40 or 80 track.

Appendix B

40/80 - Track Theory

A floppy disk is a thin circular piece of magnetic media contained within a flexible cardboard or plastic jacket. When inserted in a disk drive it rotates at 300 rpm. With a stationary read/write head this would produce a circular track around the diskette, However, the head inside a disk drive is able to move in steps, either towards the centre of the disk, or away from the centre, thereby producing a number of tracks, typically 40 or 80. Since the total distance covered by the head is the same for both 40- and 80-track systems, the head inside the 80-track drive must step twice as many times to go the same distance as the head inside the 40-track drive. This distance is measured in Tracks Per Inch or TPI. A 40-track drive has a pitch of 48 TPI and an 80-track drive 96 TPI.

Fig. B.1. below shows the basic outline of a 5 1/4" floppy disk. Its features are explained below.

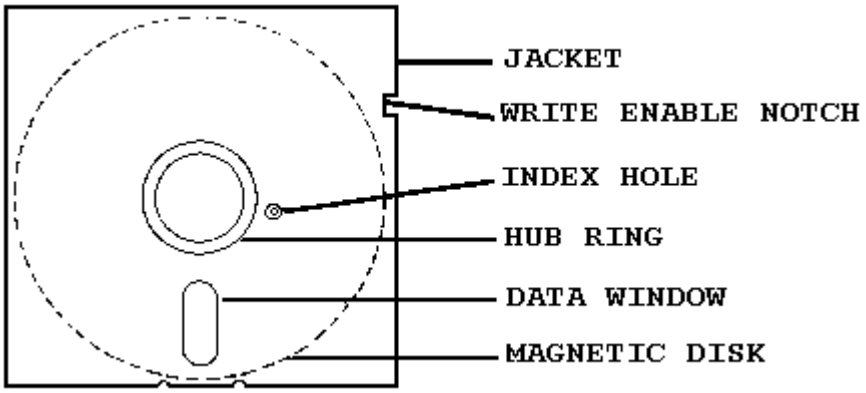


Fig B.1. A 5 1/4" floppy disk

Index Hole

This is a small hole in the jacket of the diskette, When optically aligned with a second hole in the diskette itself it will produce a short pulse. This is referred to as a soft sector index and is used by the computer to time a single revolution of the disk.

Data Window

This is an elongated hole in the jacket which allows the read/write head to make contact with the floppy disk. The window appears on both sides of the disk as 40- or 80-track drives can have double heads for recording on both sides of a disk.

Write Enable Notch

This small notch on one side of the floppy disk is used to enable the disk drive to write on a diskette, When this notch is covered with a write protect tab, the disk drive will be prevented from writing data to the diskette. Useful for preventing accidental erasure of programs and data.

Hub Ring

This exposed portion of the floppy disk is mechanically clamped between the disk drive's clutch and drive hub, which spins at 300 rpm. The clutch engages as the disk drive door is closed.

Format and Density

Before a brand new diskette can be used by the computer it has to be formatted. This is a process which segments each of the tracks into a number of sectors. A sector is like a box which is capable of holding 512 bytes of information. The size and number of sectors placed on a single track denotes the density under which the system will operate. These terms should not be confused with the number of tracks per disk or whether the disk is double or single sided. See Figure B.2. overleaf,

QL Floppy Disk Format - 5.25"

For ease of data transfer between manufacturer's floppy disk systems, it is necessary to have a common standard for disk formats. Clearly this only applies where the disks are physically compatible: physical conformation, dimensions, recording method, recording density, track spacing and positioning must all match on the source and destination machines. There is no requirement for the format for (e.g.) 5.25" and 8" disks to be the same, however, for convenience, this standard is proposed not only for 5.25" drives, but also for electrically compatible 3.5" and 3" drives. Similar formats may be derived for other standards. This standard has been based on the original Sinclair Research proposals, and compatibility between different manufacturers has already been established,

Floppy disks will be sectored in 512 byte sectors. 5.25" compatible disks will have 9 sectors per track (MFM 200 ms rotation), for a 40 track drive, single sided, this gives 180k bytes and for an 80 track drive, double sided, this gives 720k bytes capacity.

Tracks are numbered from 0, sectors on a track are numbered, by ones, from sector 1 immediately after the index mark.

The physical format is basically IBM System 34 (8" MFM) with four changes. There is no index mark recorded, the sector length flag is $02, the data record is 512 bytes long, and the write splice gap is increased,

For IBM standard format on MFM recording with 256 bytes sectors, the write splice gap at the end of a data record is 54 bytes. This is increased to 84 bytes allowing for a short term speed variation of + or - 4%, Using this, each sector is recorded in 658 bytes, this sets the gap between sector 9 and sector 1 to approximately 6250-5922 (328) bytes, allowing a long term speed variation of + or - 2.75%.

Regardless of the physical characteristicse all floppy disks will have the same directory structure. Track zero will hold the map of sector allocations. The first block of the map will be in sector 1 side 0 track 0.

Capacity

By knowing the number of tracks on a disk and how many sectors per track there are, the storage capacity of a disk can be calculated. For example, in the Cumana filing system there are 9 sectors per

track with 512 bytes per sector. This is a storage space of 4,608 bytes per track. Thus

40-track single sided drive holds 184,320 bytes,
40-track double sided drive holds 368,640 bytes,
80-track single sided drive holds 368,640 bytes and
80-track double sided drive holds 737,280 bytes.

The disk drive capacity is usually quoted in rounded form:
80 Kbytes, 360 Kbytes and 720 Kbytes respectively.

Directory

The directory of a floppy disk is an area of diskette that contains all the necessary information needed for the computer to locate, and thus load, any data contained on the diskette.

When a program is saved to the disk, the disk filing system automatically makes a directory entry. The entry contains the length of the file, the file name and the number of the sector which describes the position of the file.

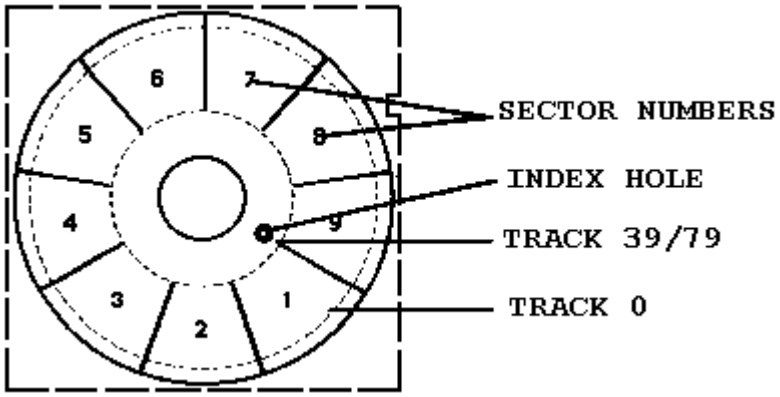


Fig B.2 Showing the 9 sector double density format (the sectors are laid out during the format process).

Appendix C

Diskette Handling Precautions

To prevent damage to diskettes follow instructions printed on the floppy disk sleeve. In particular:

* Do not place diskettes within a magnetic field.
  Magnetic fields are generated by loudspeakers, transformers, electric motors, television sets, monitors and many other electric devices.

  For instance, leaving a diskette on top of a microcomputer is inadvisable. The Micro usually contains a power supply and a loud-speaker.

* Place diskettes in their protective sleeves when not in use.
  An unprotected diskette can pick up small particles of dust that can cause damage to the read/write head.

* Keep diskettes in a rigid plastic diskette box or library case.

* Do not bend the diskettes.

* Keep diskettes at room temperature.

* Do not touch the magnetic recording surface.

* Do not place heavy objects on top of diskettes and only write lightly on the label, with a felt-tipped pen.

An example of these instructions printed on the floppy disk sleeve is shown in Fig. C.1. over page.

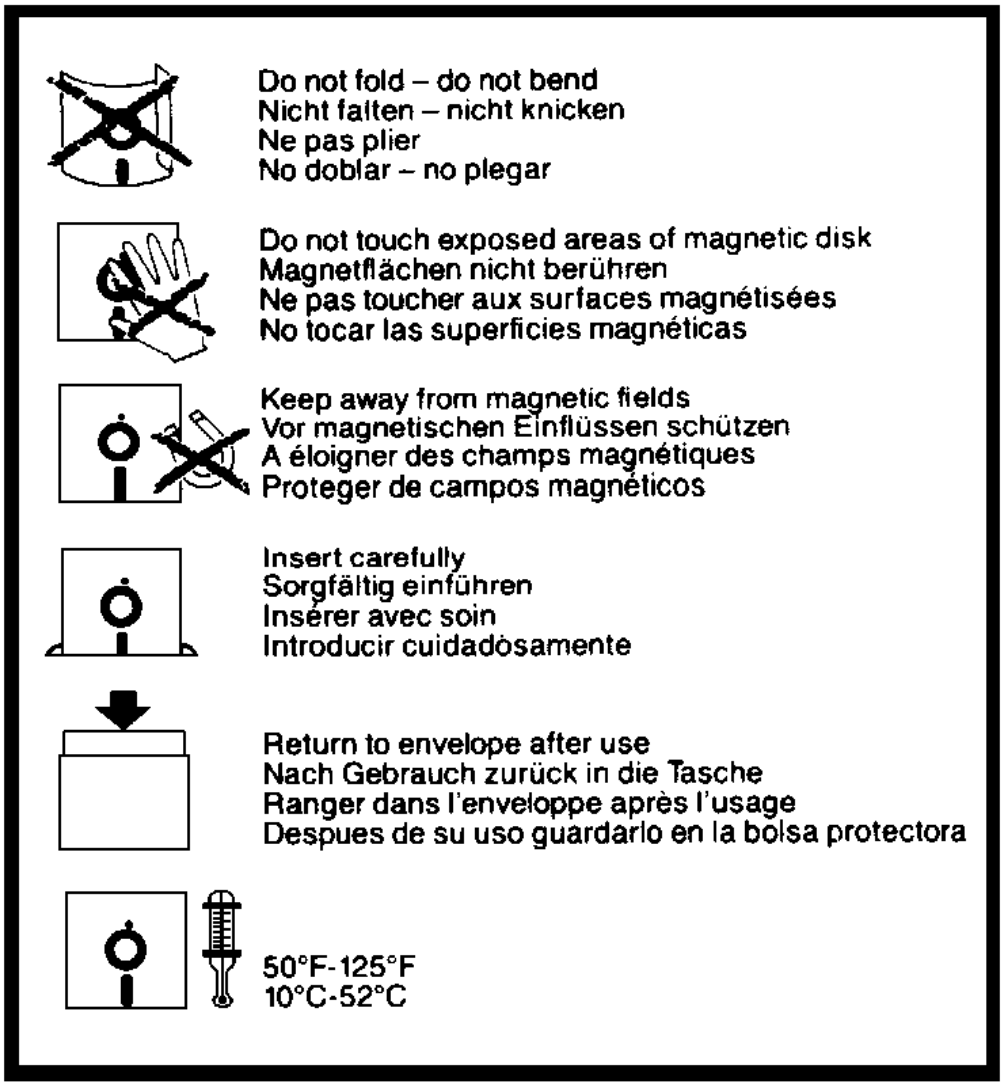


Fig. C.1 Some 5 ¼" floppy disk sleeve warnings

Appendix D

Maintenance Warning

There are no user serviceable parts inside a CUMANA disk drive or interface. Should your equipment require servicing refer it to qualified service personnel, or alternatively return it to CUMANA direct with details of the fault you experience.

Risk of Electric Shock

Should you attempt to remove the lid on a CUMANA disk drive be sure to isolate the supply first by switching off at the mains and removing the plug from the mains outlet.

Appendix E

Connecting Extra Drives

If you bought a single Cumana disk drive, either 5 1/4, 3 1/2 inch or 3 inch with an internal power supply fitted, with either the UK or European plug, you can easily upgrade to a dual drive system. On the cable that connects your single* drive to the host micro computer there is an additional 34 way edge connector. *This connector is used to daisy chain another single disk drive of the same type. i.e, if you have a 5 1/4 inch drive then the connector will only suit another 5 1/4 inch drive. The same applies to 3 1/2 inch.

When ordering a second drive for your system be sure to specify it as a second drive by post-fixing your order with 'E'. This means a disk excluding the user guide, connecting cable*, and system utility disk. The disk drive will also come factory preset to drive 2.

Connection

To connect the second drive to your existing system is a simple task, First be sure to isolate all units from the mains supply before proceeding. Remove the cover on the second drive by undoing the screws on either side of the lid and locate the cable entry point at the rear of the unit. Here you will find a 34 way cable strain relief fixed by two screws, remove the screws and the strain relief. Next locate the second connector on your existing 34 way drive cable and plug it into the rear of the drive mechanism adjacent to the cable entry point. The red or blue indicator stripe on the cable should go to the right hand side of the drive connector. Once this is connected, replace the cable strain relief, clamping the cable firmly in place. Lastly replace the cover and you are ready to use your dual drive system.

If you have any doubts about the fitting of a second drive, refer the job to qualified personnel or alternatively refer it to your dealer who will arrange to have it done for you.