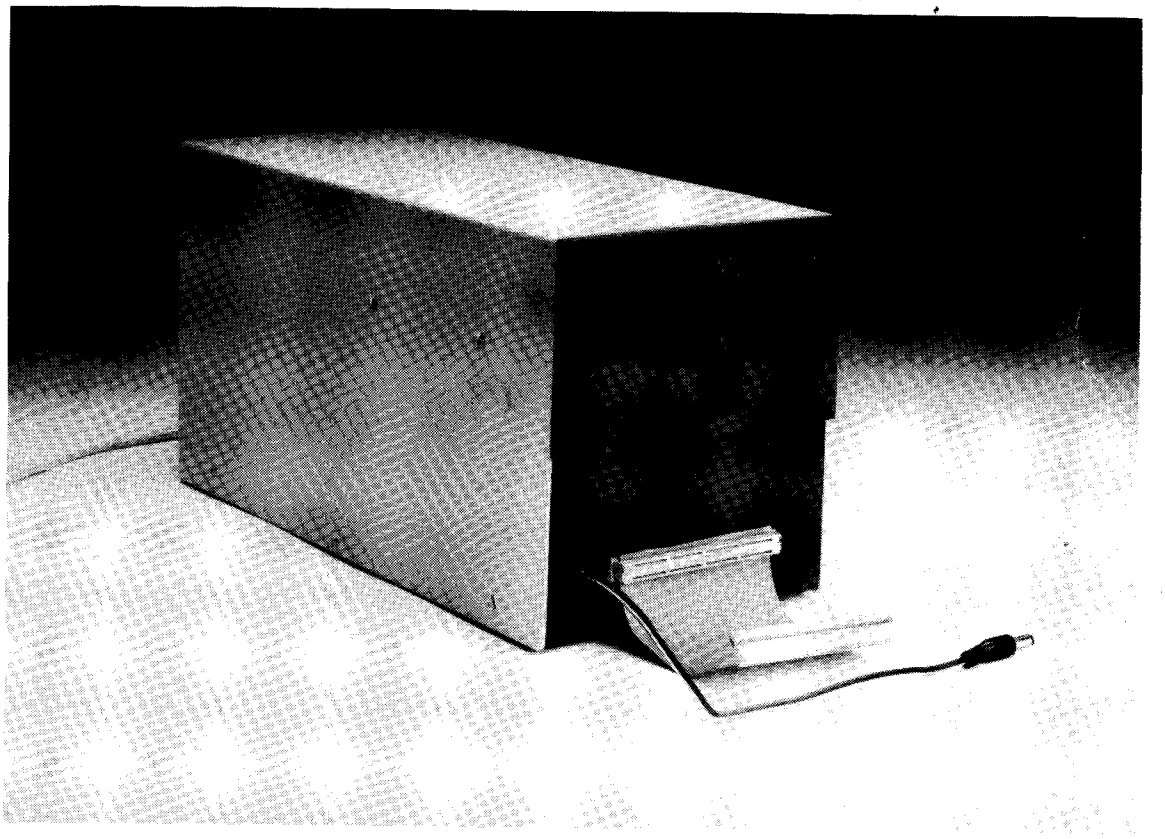
ATOM

DISC PACK



ACORN COMPUTER

Acorn Computers Limited
Fulbourn Road, Cherry Hinton, Cambridge CB1 4JN. Telephone 0223 245200

ATOM Disc Pack Manual

by

Acornsoft.

CONTENTS

*
*
1 The ATOM Disc Pack
*
*
1.1 Introduction

The ATOM Disc Pack is a free-standing unit providing upto 92K bytes of storage on standard 5.25 inch mini floppy discs in 40-track single-sided format. It contains the following:

An Olivetti OPE FD 501 disc drive unit
The Disc Operating System (DOS) in ROM
An Acorn floppy disc controller card
3K bytes of RAM
5V and 12V regulated power supplies

The unit plugs directly into the mains supply and is connected to the ATOM via a 64-way socket and an attached lead with plug. The socket accepts a ribbon cable which plugs into the bus extension socket on the ATOM, giving access to the extra memory inside the Disc Pack, and and allowing communication with the controller card for transfer of programs and data to and from the disc. The plug on the attached lead connects to the DC IN socket on the ATOM, providing 5V regulated power at up to 3A, and replacing the normal ATOM mains power supply.

1.2 ATOM Modifications

Before the Disc Pack can be used, some simple additions must be made inside the ATOM. Buffer IC's are required for the address and data buses, with a decoder to select the external data bus when an address outside the ATOM is used. The ATOM's 5V supply regulators are no longer used and are bypassed with links. ON NO ACCOUNT should the normal 8V unregulated mains supply be used with the ATOM once these links have been fitted. A label is provided warning that the unit needs a regulated 5V supply, and this should be stuck where it is clearly visible, near the DC IN socket.

The modifications require the bus buffer pack and connector, a fine-tipped soldering iron, solder, a screwdriver and some tinned copper wire. Alternatively, they can be handled by an Acorn dealer. The steps are as follows:

1 With the ATOM face down, remove the two visible screws and take off the bottom section of the case.

2 Locate the positions of IC's 2,3,4 & 5. These are clearly marked in silk-screened white near the left-hand end of the board, viewed from the keyboard side. Fit the IC's in the sockets provided as follows, taking care to put the end of the IC with the indent in the top next to the half-moon shape marked on the board:

IC2	81LS95
IC3	81LS95
IC4	DP8304 (equivalent to INS8208)
IC5	74LS30

If the bus buffer IC's have already been fitted - to drive an extension RAM board for example - only IC5 and the connector are required.

3 Locate the position of PL6. This is near the IC's just fitted and also clearly marked. Be clear which of the positions of PL5 and PL6 is to be used - PL6 is the one nearest the board edge. Remove the four self-tapping screws holding the board to the top section of the case, and lift out the board. Fit PL6 on the component side of the board (marked side 2) and carefully solder the pins on the other side (marked side 1). Check the area around the plug to make sure there are no solder splashes that could cause short circuits, and check the solder joints on the plug one by one.

4 Locate the positions of LK6 and LK7. These are adjacent to the regulators fitted to the heatsink near the middle of the board's back edge. Make the links from two short lengths of wire, and fit them in the marked positions on the component side. Solder the joints on the other side, check the area for solder splashes, and then check each joint.

5 Fit the board to the top section of the case, and replace the four self-tapping screws. Fit the bottom case section and replace its two screws.

1.3 Connections to the ATOM

The Disc pack should be connected to the ATOM with the ribbon cable supplied. One end fits into the socket on the front of the Disc Pack, and the other plugs into the socket marked ACORN BUS EXTENSION (PL6) on the back of the ATOM. The cable fits either way around, but the connectors themselves are polarized and will only fit one way. The plug on the end of the attached lead is fitted into the ATOM's DC IN socket. When the Disc Pack is connected to the mains and turned on with the switch at the back, the ATOM should produce its normal start-up message and prompt on the display.

1.4 Memory

Of the 3K bytes of RAM on the controller card, the DOS uses 2K from #2000 to #27FF, leaving 1K from #3C00 to #3FFF available to the user. The DOS ROM is situated from #E000 to #EFFF.

1.5 Introduction to Floppy Disks

The floppy disks used by the Atom Disk Pack are 5.25" in diameter and should be single-sided, single-density and soft sectored. They are plastic disks coated with metal oxide in the same way as magnetic tapes. Before new disks can be used on the Atom Disk Pack they must be prepared by "formatting" them. This process divides the disk up into sectors of 256 bytes each. (see Section 3 'Utility Programs' to fid out about formatting the disks.)

1.6 Use and Care of Discs

It is important that discs are handled and stored properly. Disc damage can impair or prevent data transfer and can result in loss of recorded information and even damage to the drive. The following points should be noted:

(a) Do not touch the mylar disc surface, especially with fingers or hard objects, and do not attempt to remove the disc itself from the

sleeve.

(b) Insert the disc carefully into the drive until the backstop is reached. Do not attempt to shut the drive door until the disc is fully inserted. The disc should be inserted with the label at the top right, the labelled side being the last one to enter the drive.

(c) Open the drive door and adjust the position of the disc if it rotates noisily.

(d) Avoid damage to the centre hole which locates the disc onto the drive hub

(e) Return the disc to its paper storage jacket with the head slot inside when not in use.

(f) Keep discs away from magnetic fields, including power supply transformers and cathode ray tubes.

(g) Do not bend or fold discs.

(g) Store and use discs at room temperature and away from direct sunlight.

(h) Write on the disc label before sticking it on the disc. Use a felt pen, NOT a biro, to write on labels that are already on discs.

The notch on the side of the disc sleeve can be covered with a self-adhesive tab to prevent the drive writing to the disc.

*
*
2 The Disc Operating System
*
*

2.1 Introduction

The Disc Operating System (DOS) handles the creation, deletion and cataloging of disc files on a single 5.25" disc drive. It is a 4K program resident in ROM, using 2K bytes of RAM for catalogue and file buffers. The DOS replaces all functions of the ATOM Cassette Operating System and provides additional features that take advantage of the speed and random access capabilities of disc storage.

2.2 Direct DOS Commands

Direct commands are those entered by the user, and are concerned with the transfer of program and data files as a whole. Each is preceeded by '*', which enters the DOS command line interpreter, and it can be abbreviated by entering enough characters to distinguish the command from others, followed by '.'.

Filenames of up to seven characters are allowed. Where a name or other string parameter contains no spaces or " quotes, it may be entered directly. Otherwise it must be contained by " quotes, with every " in the name typed as "".

Command — Abbreviation

***DOS** — ***D.**

Enters the Disc Operating Sytem. The COS is re-entered on BREAK.

***CAT** — ***.**

Displays the catalogue. The first line shows the title of the disc and the current qualifier. The filenames are displayed arranged by qualifier, with '#' indicating that the file is locked.
The catalogue is read off disc unless the disc is in the 'ready' condition (still spinning), in which case it is transferred from the catalogue buffer (at #2000).

***DIR** — ***D.**

Reads the catalogue from disc to the buffer but does not display it. This command is often used to wait for completion of a previous operation, since the prompt may re-appear while a transfer continues under interrupt.

***TITLE <title>** — ***T.**

Sets the title of the disc to the first 13 characters of <title>.

***SETx**

Sets the qualifier for all subsequent operations on files to x. Operations that use existing files will only search among those created under the qualifier currently set. The default qualifier is space. When returning to the default qualifier, be sure to enter the space after the SET command.

***USEx**

Temporarily sets the qualifier to x. The system returns to the previous qualifier after the next successful operation. Two successive USE commands are equivalent to a SET.

***LOCK <name>**

Locks the named file. Locked files cannot be deleted or updated.

***UNLOCK <name>** ***U.**

Unlocks the named file.

***INFO <name>** ***I.**

Displays information about the named file. The format is:

qualifier :	lock state	filename	load address	execution address	length in bytes	start sector

For example:

```
s: #BASIC    C000 C2B2 01000 002
```

The '#' indicates that the file is locked.

***MON** ***M.**

Turns on messages. Subsequent file operations print out the file's information in the format of *INFO.

***NOMON** ***N.**

Turns off messages.

***SAVE <name> aaaa bbbb cccc** ***S.**

Creates a file. The contents of memory from aaaa to bbbb-1 are saved as the named file. If a file with the same name already exists under the current qualifier, it is deleted. The adress cccc is an optional execution address and it defaults to aaaa if not supplied.

***LOAD <name> aaaa** ***L.**

Loads the named file. The file is loaded into memory starting at aaaa, or at the file's start address if this is missing.

***DELETE <name>** ***DE.**

Deletes the named file.

***GO aaaa**

Executes the machine-code subroutine at aaaa.

***RUN <name> <string>**

Load and run the named file, using the start and execution address. <string> is transferred to memory starting at #0140 with a terminating CR, where it can be accessed by the user's program.

***EXEC** ***E.**

Executes the named text file as if it had been entered at the keyboard.

2.3 Using the DOS from BASIC

The DOS redirects BASIC vectors so that commands, statements and functions that handle the transfer of programs and data to and from cassette via the COS perform their equivalent functions with disc. Unnamed files are not allowed, PTR and EXT are added to allow random access within files, and the keyboard and display can be used as input and output files respectively. Upto five files can be open simultaneously. Features such as qualifiers, messages etc. operate as for direct DOS commands. File handling from BASIC is discussed fully in "Atomic Theory and Practice", so only a brief summary of the facilities is given here.

LOAD "<name>" **L.**

Loads the named file into the current text space as a BASIC program. LOAD can only be used in direct mode.

SAVE "<name>" **SA.**

Saves the BASIC program in the current text space as the named file.

FIN "<name>" (function) **F.**

Opens the named file for input and returns its handle, or zero if the attempt is unsuccessful. File handle zero gives input from the keyboard.

FOUT "<name>" (function) **FO.**

Opens the named file for output and returns its handle, or zero if the attempt is unsuccessful. File handle zero gives output to the display.

PUT A,W

Outputs the four bytes of W to the file whose handle is A.

GET A (function) **G.**

Inputs four bytes from the file whose handle is A, and returns an integer.

BPUT A,B **B.**

Outputs the least-significant byte of B to the file whose handle is A.

BGET A (function) **B.**

Inputs a byte from the file whose handle is A, and returns the value.

SPUT A,S **SP.**

Outputs the string S to the file whose handle is A.

SGET A,S **S.**

Inputs a string to S from the file whose handle is A.

FPUT A,%F **(FP ROM)**

Outputs the 5 bytes of the floating-point variable %F to the file whose handle is A.

FGET A **(function, FP ROM)**

Inputs 5 bytes from the file whose handle is A, and returns a floating-point value.

PTR A

Returns the value of, or assigns a value to, the pointer to the next byte for input or output in the file whose handle is A. PTR can be thought of as a special variable - in an expression it returns the current value of the pointer as an integer, and on the left-hand side of an assignment statement it updates the pointer with the result of evaluation of the right-hand side.

EXT A **(function)**

Returns the extent (current length) of the file whose handle is A.

SHUT A **SH.**

Closes the file whose handle is A. SHUT 0 closes all files.

2.4 DOS Responses and Error Messages

After a DOS command has completed successfully, the BASIC prompt returns. If for some reason the command is not understood or cannot be carried out, a message indicating the problem is printed first. The various responses and their meanings are as follows:

Command?

The command is not recognised.

Syntax?

The parameters are illegal or incorrectly presented.

Name?

The filename is longer than eight characters.

File?

The file is not found.

Disc prot

The disc is write-protected.

File prot

The file has been locked.

Disc full

There is insufficient room on the disc to create the new file.

Full

There are already 31 files in the catalogue.

If the fault is with the disc or drive rather than the command, one of

the following error messages is given:

Disk error 08
Clock error

During a disc read operation a clock bit was missing.

Disk error 0A
Late DMA

During a disc transfer the processor did not respond fast enough. Probably due to a faulty disc controller card.

Disk error 0C
ID filed CRC error

The cyclic redundancy check derived from the data read back differed from that stored on the disc.

Disk error 0E
Data CRC error

The cyclic redundancy check from the data read back differed from that stored on the disc during a disc read.

Disk error 10
Drive not ready.

During a transfer the disc stopped rotating. Often due to a badly inserted disc.

Disk error 14
Track 0 not found

The controller failed to find track 0. Often due to an unformatted disc.

Disk error 18
Sector not found

The controller failed to find the required sector. Often due to a corrupted or unformatted disc.

*
*
3 The Utility Programs
*
*
A disc with four utility programs is supplied with the ATOM Disc Pack. A utility program is used by typing '*', followed by its name, followed by RETURN.

FORMAT

Initialises a disc to the Acorn 40-track soft-sectored format. All new discs must be formatted before they can be used. On running, the program dislays a prompt message and waits for YES to be entered (without RETURN). Before replying the ultilities disc should be replaced by the disc to be formatted. The program initialises the entire disc, clearing the catalogue, and then verifies each sector.

COMPACT

Relocates the files on a disc to make wasted space left between them by deletion and re-saving available for use by new files. The program waits for the user's disc to be put in the drive, and begins when a key is pressed. It displays information on all files before and after relocation, and the number of free sectors when completed.

INFALL

Gives information on all files on the disc in the format fo *INFO. The program waits for a key to be pressed.

VERIFY

Lists sectors on the disc which contain errors. The program waits for a key to be pressed.

*
*
Appendix A
ATOM Disc Pack Parts List
*
*
Chassis Assembly

DISC DRIVE	Olivetti 5.25 inch 40 track single sided
CHASSIS	202,005,M1
COVER	202,005,M2
WIRING DIAGRAM	202,005,W
SW1	Mains switch, Arrow 1602 - 11E
MAINS CABLE	3 core, 3 m
CABLE RESTRAINER	13 mm mounting
FS1	Fuseholder 20 mm
FUSE	3A 20mm
T1	Mains transformer
C1	4700uF 25v Elect. capacitor
C2	4700uF 25v Elect. capacitor
C3	4700uF 25v Elect. capacitor
TAPPED PILLARS	25 mm M3, 2 off
RUBBER FEET	4 off
REG 1	78H05K (TO3)
REG 2	7812 (TO126)
RIBBON CABLE	34 way 250 mm
INTERNAL WIRE	24/0.2, A/R
IDC CONNECTOR	34 way, T & B Ansley 609 - 3400M
PCB CONNECTOR	34 way, T & B Ansley 609 - 3415M
POWER CONNECTOR	4 way, AMP 1 - 480424 - 0, pins 61117 - 1
SCREW	Pan head M3 6mm, 6 off
SCREW	Pan head M3 12 mm, 7 off
NUT	Full M3, 17 off
WASHER	Plain M3, 21 off
SCREW	Self tapping, panhead, No.6 12 mm, 4 off
USPIRE NUTS	No 6, 4 off
WASHER	Plain M4, 4 off
SOLDER TAG	M3
SLEEVING	For mains connections, A/R
BARRIER STRIP	6 way, RS 423 - 497
BOOT, INS	For fuseholder
CLIP	For capacitor RS 543 - 018, 3 off
C5	Capacitor, disc ceramic, 470nF
C4	Capacitor, disc ceramic, 100nF
THERMAL GREASE	For REG 1 and REG 2, A/R
PL1	Header, 3 way, Molex
BR1	Bridge rectifier, 6A, RS 262 - 078
BR2	Bridge rectifier, 2A, RS 261 - 592
HEADER INSERTS	Molex, 3 off
SPADE CONNECTORS	0.25 inch, 2 off
SCREW	Countersunk head M3 6 mm, 16 off
CABLE RESTRAINER	For Atom power cable

Printed Circuit Board Assembly

PCB	Acorn 202,005
IC1	7438
IC2	74LS00
IC3	7406
IC4	74LS04
IC5	4013
IC6	4013
IC7	74LS93
IC8	74LS93
IC9	74LS393
IC10	4020
IC11	74LS02
IC12	74LS139
IC13	8271 (Intel)
IC14	74LS42
IC15	2532 Atom DOS ROM
IC16	2114
IC17	2114
IC17	2114
IC18	2114
IC19	2114
IC20	2114
IC21	2114
IC22	74LS138
R1	3K3
R2	3K3
R3	3K3
R4	150
R5	150
R6	470
C1	10 nF Ceramic disc
C2 - C13	47 nF Ceramic disc
C14	47 uF 10 v Electrolytic
C15 - C23	47 nF Ceramic disc
	34 way, Molex terminals
PL1	3 way, Molex terminals
	64 way Euro-connector socket, straight pins
	64 way Euro-connector plug, bent pins
XTAL	4 MHz
	40 pin IC socket, 1 off
	24 pin IC socket, 1 off
	18 pin IC socket, 6 off
	16 pin IC socket, 4 off
	14 pin IC socket, 10 off

*
*
Appendix B
Circuit Descriptions
*
*

The Controller Card

The Acorn disc controller card uses an Intel 8271 integrated circuit to minimize both hardware and software overheads involved in using mini-floppy disc drives. The controller has a 64 way connector for the stancard Acorn bus and a 34 way ribbon cable plugs onto the board to connect the drive.

The 8271 is addressed in the memory map at page A in block 0 by IC22. The high power required by the mini-floppy interface is provided by 7406 and 7438 drivers. A 2MHz clock is required by the 8271 for master timing, and this is generated by division of a 4MHz clock produced by a crystal oscillator IC4.

The data and clock signals are mixed during recording and thus data read back from the disc has to be processed to recover the information. The 8271 does this itself using the digitally timed retriggerable monostable IC7 and IC8, which generates a data window signal. IC5 and IC6 are two digital monostables used to indicate drive ready status based on the index pulse repitition rate. Transfer between the floppy disc controller and processor is synchronised with the NMI (non-maskable interrupt). The controller automatically unloads the head after 10 revolutions have occured without further access requests from the processor.

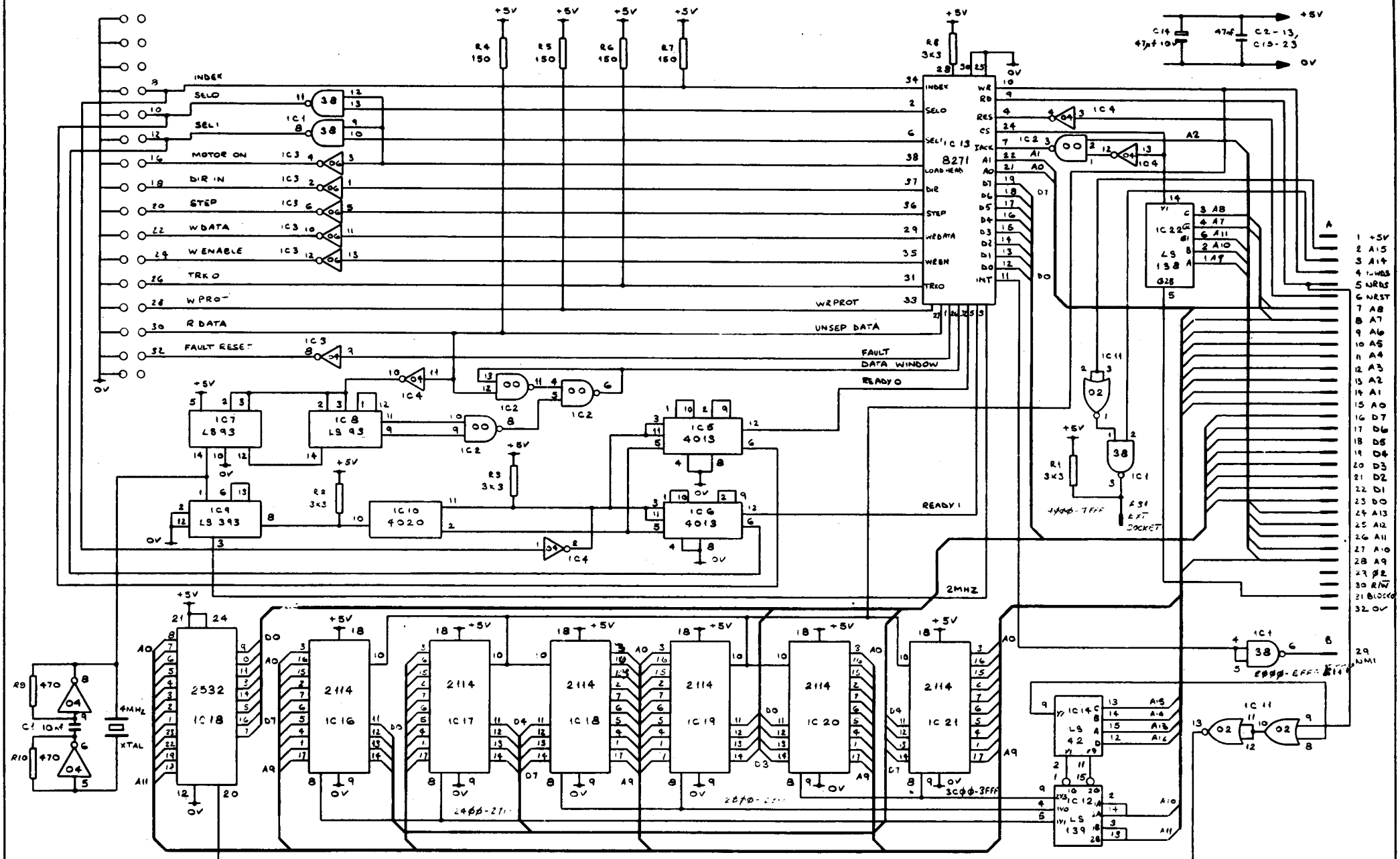
The Disc Drive

The drive unit is an Olivetti OPE type FD 501. It is a 40 track single sided unit and using single density recording provides a formatted storage capacity of 92.16 Kbytes. It has a printed circuit board attached to it containing the read/write electronics and the control signal processing circuitry, and industry standard interface connectors for power and logic signals.

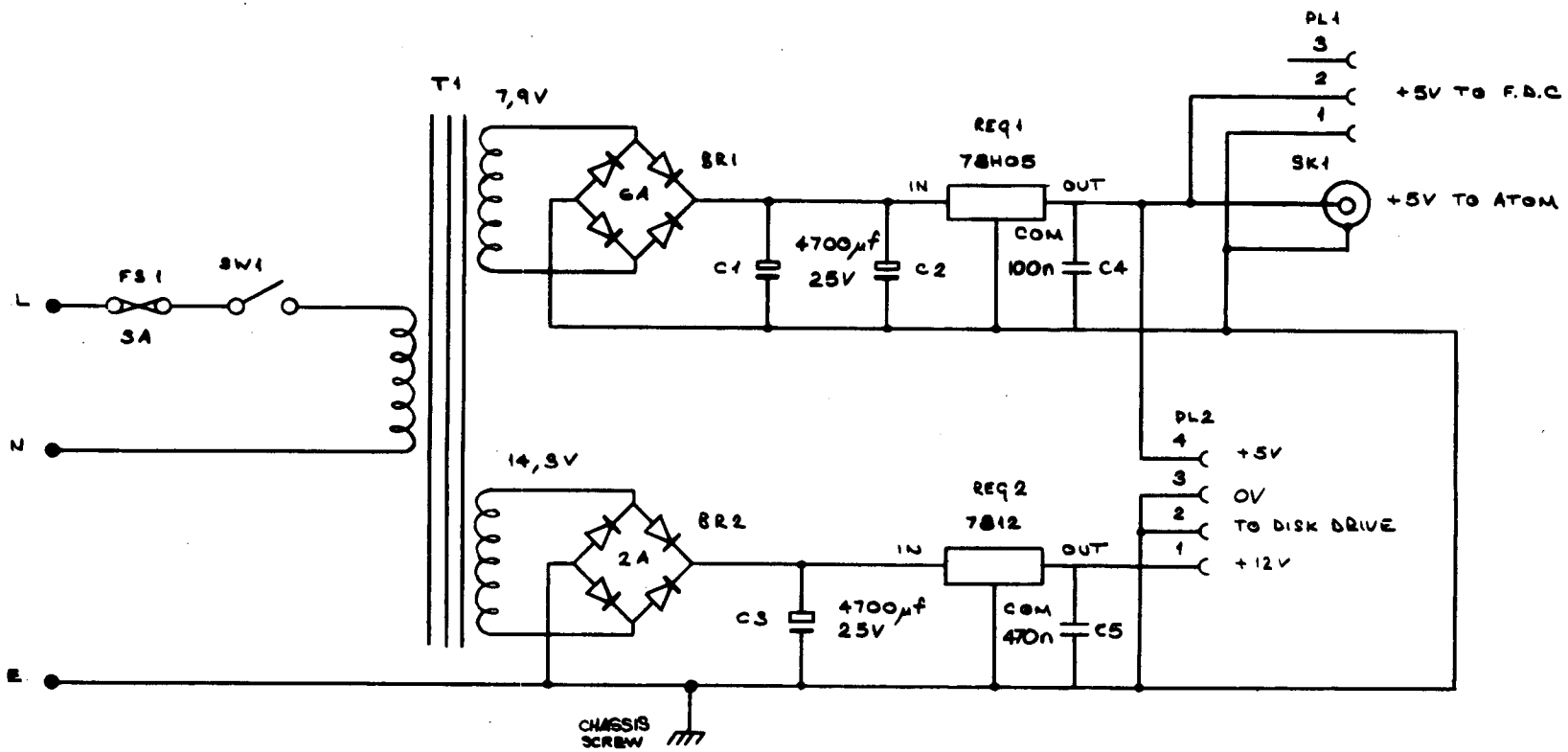
Precise head positioning is assured by using a stepper motor driving a spiral cam directly. The head carriage is guided by a steel peg which rests in a track on the spiral cam. A D.C. servo controlled motor is used to rotate the disc spindle at 300 rpm.

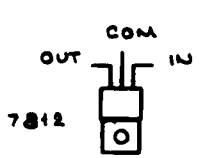
The Power Supply

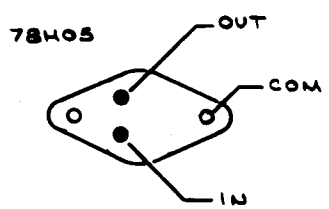
The disc drive requires regulated supplies of 5v and 12v and these are supplied by the disc pack internal power unit. Spare capacity is available on the 5v supply, the current availeble being 3A, so a fully expanded Atom may be powered.



DATE	21-9-81	13-1-82	© COPYRIGHT 1981 ACORN COMPUTERS LTD 4 MARKET HILL CAMBRIDGE TEL: 53723	DRN FWDS	TITLE CIRCUIT DIAGRAM FOR ATOM DISC PACK F.D.C. CARD	REF 106,000/c
ISSUE	1	2		DATE 21-9-81		ACORN COMPUTERS LTD



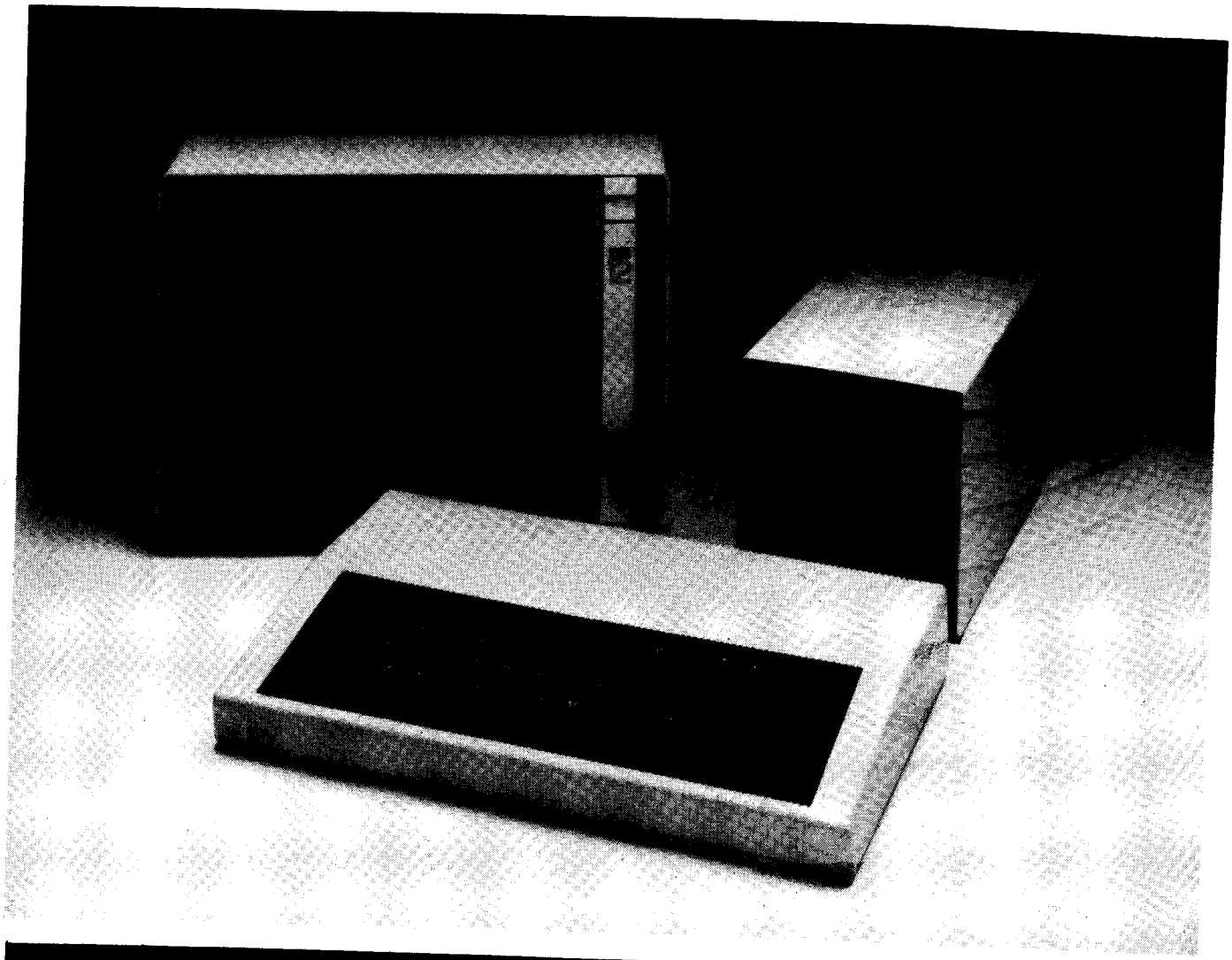


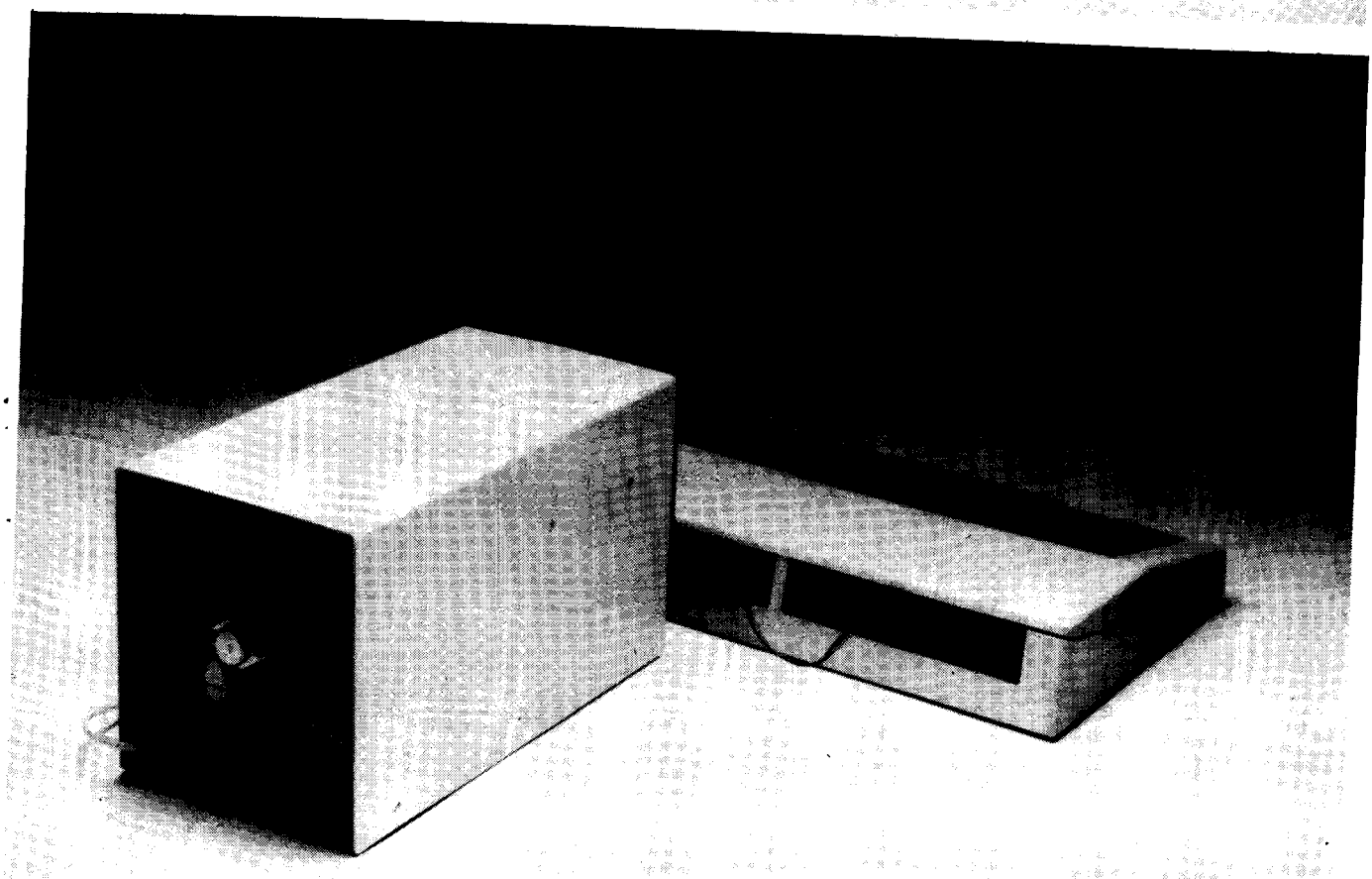


ALL MAINS JOINTS TO BE SLEEVED

INTERNAL WIRING 24/0,2.

DATE	30-9-81			© COPYRIGHT 1981 ACORN COMPUTERS LTD.	DRN GWDS	TITLE WIRING DIAGRAM FOR ATOM DISC PACK.	REF 201,011
ISSUE	1				DATE 18-9-81		ACORN COMPUTERS LTD





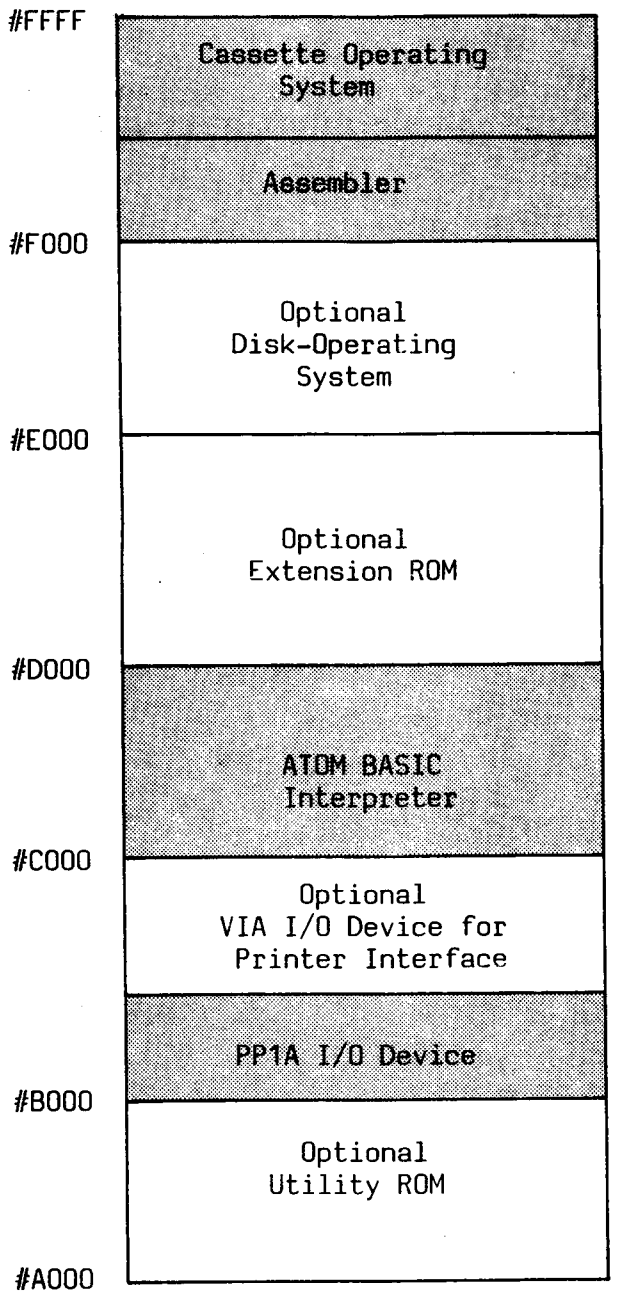
*
*

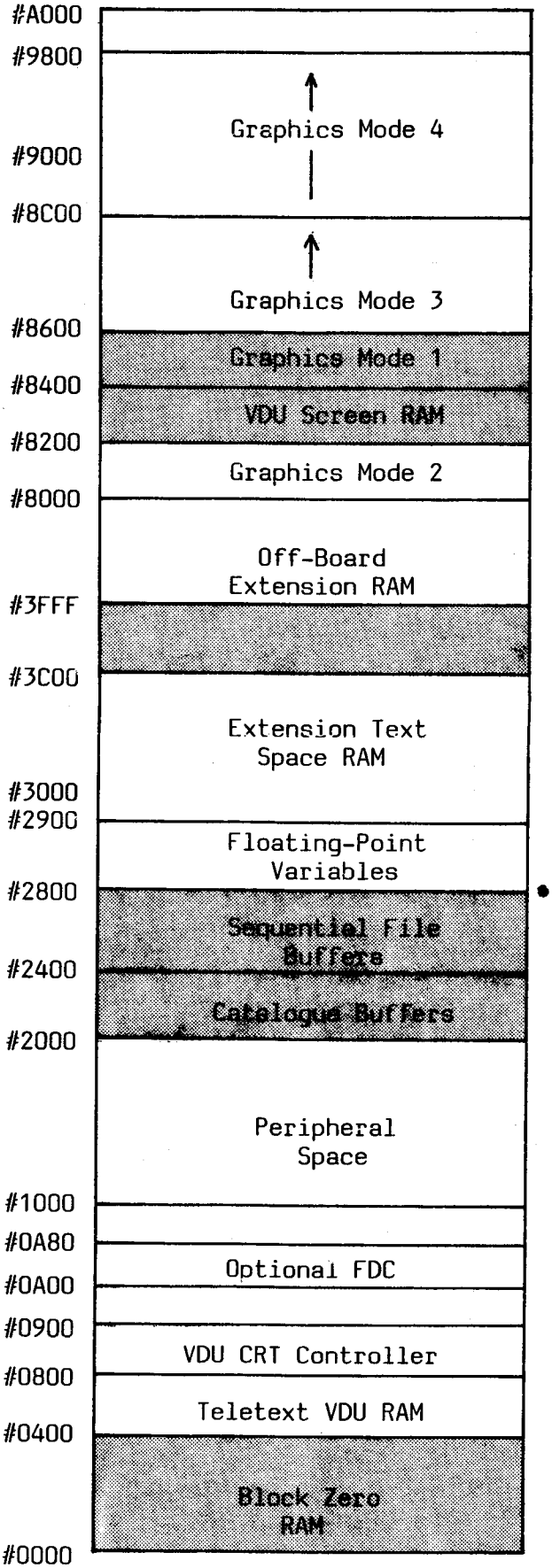
Appendix C

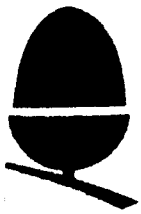
Memory Map

*
*

The following diagram shows how the ATOM'S address space is allocated. Shaded sections are present in the minimal system ATOM, with the disk pack additions.


#FFFF
Cassette Operating System
Assembler
#F000
Optional Disk-Operating System
#E000
Optional Extension ROM
#D000
ATOM BASIC Interpreter
#C000
Optional VIA I/O Device for Printer Interface
PP1A I/O Device
#B000
Optional Utility ROM
#A000


#A000
#9800
Graphics Mode 4
#9000
#8C00
Graphics Mode 3
#8600
Graphics Mode 1
#8400
VDU Screen RAM
#8200
Graphics Mode 2
#8000
Off-Board
Extension RAM
#3FFF
#3C00
Extension Text
Space RAM
#3000
#2900
Floating-Point
Variables
#2800
Sequential File
Buffers
#2400
Catalogue Buffers
#2000
Peripheral
Space
#1000
#0A80
Optional FDC
#0A00
#0900
VDU CRT Controller
#0800
Teletext VDU RAM
#0400
Block Zero
RAM
#0000



Atom Disk Pack Spool, Exec and Verify

The Atom disk pack provides several utilities to the user which were not documented in the initial release of the Atom disk pack manual. These utilities include the ability to keep a record of all information printed to the screen on the disk, and to store on the disk a set of commands which can then be obeyed as if they were typed in at the keyboard. Also mentioned was a verify command which is documented here for those who would find this a useful utility. The last undocumented feature is the software to drive the Acorn 40 character VDU card from the Atom. I hope you find these facilities a useful expansion to the many commands already documented for the disk pack.

*SPOOL "name"

This command opens a file of the given name and all characters subsequently printed to the VDU system are also stored in this file as ASCII codes. The spooling of characters to a file is stopped by using the

```
*SHUT
```

command.

A few examples will probably make this a little clearer.

Characters can appear on the screen in several ways, such as:

As a result of a print statement in a Basic program.
When a character is typed at the keyboard.
Or as a result of some command in Basic or the operating system (eg., LIST or *CAT).

All these characters will end up in the spool file if one is opened. One important point to note is that the carriage return and line feed codes sent to the screen at the end of each line of a Basic listing will also end up on the spool file. For example the Basic program:

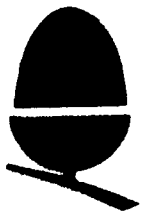
```
10   *SPOOL "Example"
20   PRINT "1234567890"
30   *SHUT
40   END
```

will create a file containing the following bytes in Hex:

```
31 32 33 34 35 36 37 38 39 30.
```

where as the sequence:

```
>*SPOOL "Program"
>LIST
  10  * SPOOL "Example"
  20  PRINT "1234567890"
  30  * SHUT
  40  END
>* SHUT
```



will create a file with the ASCII codes for the Basic prompt and the characters LIST followed by the bytes 0A and 0D which are the codes for line-feed and carriage return. Next in the file will be the codes for the next line which consists of some spaces followed by characters 1 and 0, a space and so on until the line-feed and carriage return at the end of this line. This continues including the codes, for SHU and T and a final line-feed before the spool file is closed.

The contents of a spool file can be examined or printed to the printer by a program such as:

```
 5  P.$2
10  A = FIN "Program"
20  DO
30  P. $BGET A
40  UNTIL PTRA  = EXTA
50  P.$3
60  END
```

Hence a record can be made of the output from the computer for subsequent printing, or examination by another program. The spool file can also be a very useful way of storing programs or common subroutines and this will be explained after the description of EXEC.

*EXEC "name"

This command is a complimentary utility to spool which gets bytes from a serial file on disk and enters them to the computer as if they had been typed at the keyboard. Hence doing a Exec on the file "Example" created as in the description of spool would go as follows:

```
*EXEC  "Example"
1234567890
```

and because there was no carriage return stored in the spool file, you would have to hit return to get the prompt back.

A sequence of commonly used commands can be stored as a file to be Execed. This can be very useful if you have a program which has a complicated set of start up instructions, for example, a data base program may need the following sequence to start it.

```
*LOAD "Data" 8200
?18 = #28
LOAD "Program"
RUN
```

Obviously to type this in every time would be a little tedious, so an Exec file could be created using the following program. (See overleaf)

```
10  A = FIN "Start"
20  SPUT A, "*LOAD ""Data"" 8200"
30  SPUT A, "?18 = #28"
40  SPUT A, "LOAD ""Program"""
50  SPUT A, "RUN"
60  SHUT A
70  END
```

After this has been done, the program can be started by typing:

```
*EXEC START
```

Another use of the Exec file is to merge Basic programs. This is particularly useful in loading commonly used rountines for programs. A program similar to the one above could be used to create a file containing a few lines of Basic including line numbers and when the file was Execed, the lines would be added to any Basic programs already in the current text space. if line numbers are duplicated, obviously the lines in the Exec file will overwrite the lines in the program, so care must be taken to ensure that you do not lose lines you want.

An alternative method of creating an Exec file for a routine is to spool a copy of the required routine, in the following way.

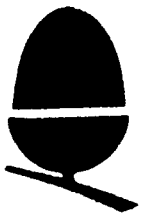
If we have a program with a routine between lines 2000 and 3000 that we want to use in other programs we can spool this section:

```
>* SPOOL "Section"
>LIST 2000, 3000
 2000 - - -
 - - - - -
 - - - - -
 2900 - - -
>* SHUT
```

The file created in this way will contain the >LIST and >* SHUT lines which will cause errors when the file is Execed in, but this will not be a problem. The file however, also contains line-feed characters which must be removed. The following program will create a new file called ROUTINE with the line-feeds removed.

```
10  A = FIN "Section"
20  B = FOUT "Routine"
30  E = 0
40  DO
50  D = BGET A
60  IF D = 13; IF E = 10; PTRB = PTRB-1
70  BPUT B,D
80  E = D
90  UNTIL PTRA >= EXTA
100 SHUT A
110 SHUT B
120 END
```

It is then possible to use the EXEC command to add this routine to your programs. Several such files could be created for different routines, with different line numbers, and this could constitute a useful library of functions.



* VDU n

Where n is Ø or 1.
This command can be used to redirect all screen output to a set of routines in the disk pack which drives the Acorn 40 character Prestel character set VDU. This card from Acorn can be connected inside this Atom with a suitable power supply and will provide a 40 character by 25 line screen display on a colour monitor, further details can be found in the Acorn Computer Systems brochure. Output will be sent to the 40 character VDU following *VDU 1 and to the normal Acorn screen following *VDU Ø.

Disk Verify for Acorn Atom

Verify

The verify utility listed here is written in Atom Basic to run on the Atom in conjunction with the Atom disk pack. The utility will attempt to verify all the sectors on a disk inserted in the drive indicating which files any corrupted sectors are in.

The disk to be verified should be inserted in the drive before the program is run.

The program will first attempt to load the catalogue of the disk. This is stored on the disk in two sectors on track zero nearest the edge of the disk. If there are any errors in either of these two sectors at this point, the rest of the verify program cannot operate and the program will report the fact and stop. Assuming the catalogue has loaded correctly, the program will then report.

NUMBER OF ENTRIES N

SECTORS 400

Where N is the number of file entries in the catalogue. The number of sectors should be 400. A disk formated for eighty tracks will have 800 sectors and will not work in the standard Atom disk pack, though it is still possible to read the catalogue off an eighty track disk as this is stored on the first track of the disk.

The program will then, for each program saved on the disk, print out the following details:

The file name

The start address in memory where the file would be located to

The execution address of the file

The length of the file in bytes

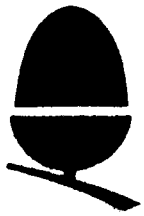
The location of the first sector of the file on the disk

The number of sectors on the disk occupied by the file

A row of dots should be printed below each file name. each dot represents a sector of the file that has been verified. Any stars printed indicate sectors that can not be loaded or verified. Any areas of the disk not allocated to files will be reported as blank and verified in the same manner, in particular you will normally find a large blank area after the last file on the disk.

If you do not wish to verify all of the disk the escape key may be used, however as the screen is disabled at certain points in the program, control f should be used after pressing the escape key in order to re-enable the screen. Pressing the return key will then cause an error message to occur before the prompt appears. The correct sequence to escape is therefore escape, control f, return, the error message can be ignored. The result on the screen is a clear visual indication of the size of files and blank areas on the disk, and those areas of the disk that contain corrupted sectors.

The corrupt sectors may be caused by exposure of the disk to heat, magnetic fields, dust, smoke, or by power failure or surges while the disk is inserted in the drive or file transfers are taking place. In any case the information should be copied off any disk on which errors start to occur and the disk reformated to prevent the risk of further loss of data from a suspect disk.



```
ERROR 94
>L.
     1REM*****************************
     2REM (C) ACORN COMPUTERS (1982)
     3REM*****************************
     5REM NO RECOMENDATION IS GIVEN
     6REM OR IMPLIED AS TO THE USE OF
     7REM ANY OF THE ROUTINES USED
     8REM IN THIS PROGRAM FOR ANY
     9REM OTHER  PURPOSE
    10$TOP="P.'""ERROR IN CATALOGUE SECTOR 1 OR 2"";END"
    11$TOP="P.'""ERROR IN CATALOGUE SECTOR 1 OR 2"";END"
    12?16=TOP;?17=TOP/256;*CAT
    13@=5
    15M=&FFFF;U=&9C
    20L=&2000;H=&2100
    30N=H?5
    32P.'"NUMBER OF ENTRIES"N/8
    35P.'
    45P.'
    50S=(H?6&15)*256+H?7
    55P."SECTORS" S'
    56P.'"NAME    STRT EXEC LENGTH SSEC SEC"
    60E=N
    65F=2
    70d
    75 B=H!E&M;Q=H!(E+2)&M;W=H!(E+4)&M+(H?(E+6)/16)*M
    80T=(H?(E+6)&15)*256+H?(E+7)
    85R=W/256;IFW&255<>0;R=R+1
    90IFF<T;P."BLANK                    "F T-F;A=F;C=T-1;X=100;G.r
   100IFE=0;END
   110FOR O=0TO6
   120P.$(O?(L+E));N.
   125P. &B,&Q,&W,T,R
   126 A=T ;C=T+R-1
   127IF R=0;G.190
   130X=190;G.r
   190 E=E-8
   200 IFE>=0;G.d
   300END
  1130rF=A
  1135LINK&E75B
  1140 !U=&A000
  1150 U?4=0;U?5=1;U?6=F/256;U?7=F
  1153 P.$21
  1155$TOP="G.s"
  1156Y=!16;?16=TOP;?17=TOP/256
  1160LINK &E4A3
  1165LINK &E226
  1167?16=Y;?17=Y/256;P.$6"."
  1180F=F+1;IF F<=C;G.1140
  1185P.''
  1200G.X
  2175s?16=Y;?17=Y/256;P.$6"*"
  2200G.1180
  2210REM
  2220REM
>
```