Real Time Clock

OWNER'S MANUAL

SOLIDISK TECHNOLOGY

THE STL REAL-TIME CLOCK

The STL REAL-TIME CLOCK and Calendar (RTC) is a sophisticated add-on for the BBC microcomputer. It is installed simply by plugging it into a vacant ROM socket, and by programming the options.

Release date: Mon, 14 July 1986. 10:00:00

Solidisk Technology Limited, 17 Sweyne Avenue
Southend-on-Sea
TELEPHONE (0702) 354 674

SECTION 1:

INSTALLATION

The RTC is best installed in the rightmost ROM socket on the BBC computer board. The battery can be left in an unused corner on the left. See the figure below.

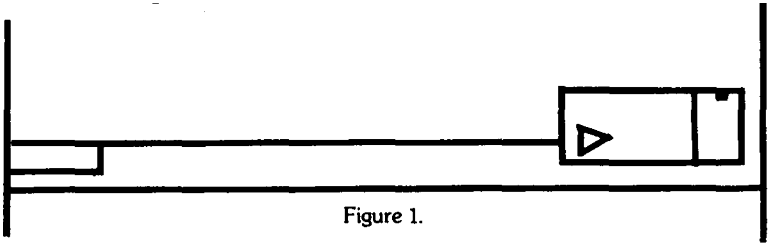

Figure 1.

If you have a Sideways ROM expansion board such as APTL's or Watford's you may install it in any other ROM socket. The normal version of the RTC is not suitable for any of the ROM sockets on the Solidisk Fourmeg 32 or 256 boards; there are three reasons for this: 1) the RTC is a slow device, 2) the RTC has a small capacity ROM and 3) it would need a modification before it would work in these sockets.

If there is enough clearance above the RTC board, you may resolder the battery directly to the board — removing the trailing battery lead.

MAINTENANCE:

The RTC does not require any special attention except the usual precaution when a battery (however well constructed it may be) is in use. Avoid intense heat and physical damage to the battery which may result in a leak of its chemicals.

The battery is of the NICAD rechargeable type. It is 'trickle charged' when the computer is switched on. After an initial 12 hours period charging time, the RTC should stand prolonged periods without power.

Don't switch on the computer just for charging the battery if you use your computer regularly. You may remove the RTC and re-install it at any time. Its battery will keep the information stored safe while not in use.

SECTION 2 TIME COMMANDS

THE *SETTIME COMMAND:

The *SETTIME command followed by the date or the time will reset the time kept by the RTC.

An example of *SETTIME:

```
*SETTIME 15:20:00 <RETURN>
```

Another example of *SETTIME:

```
*SETTIME Mon, 21 JUL 86 <RETURN>
```

This may produce the following display, if followed by *TIME:

```
Mon, 21 Jul 1986. 15:20:20
```

```
*SETTIME dow, dom mth yr.hh mm ss
```

The COMMA must be used after the day of the week, a PERIOD after the year. hh, mm and ss can be separated by space, comma, semicolon, period or colon.

The command may have only the first part (including the period after the year) or the second half (DO NOT start with a period).

dow = <day of the week>
dom = <date in the month>
mth = <three first letters of the month>
yr = <two digits of the year (eg 86)>
hh = <a number between 0-23 for the time in hours>
mm = <a number between 0-59 for the time in minutes>
ss = <a number between 0-59 for the time in seconds>

Valid 'days of the week' are: SUN, MON, TUE, WED, THU, FRI and SAT.

Valid 'months of the year' are: JAN, FEB, MAR, APR, MAY, *JUN,* JUL, AUG, SEP, OCT, NOV and DEC.

Valid years are 1900 to 1999 or 00 to 99, the 20th century is automatically assumed.

Number of days in the month and leap year are automatically taken into account.

THE *TIME (*TI.) COMMAND:

*TIME will produce (if set), the day, date and time. A typical display is like this:

```
Sun, 04 Jul 1986. 15:10:05
```

Day of the week, date, month, year, hours, minutes and seconds.

If the time was not set (or the battery is flat), the time display will be: ` TIME UNSET'. See *SETTIME below.

THE *RTC COMMAND

*RTC <RETURN> will display (only in mode 7) continuously the time in the top right corner of the screen, just like a digital clock.

The display is like this:

```
15:11:30
```

*RTC ALL <RETURN> will display in all screen modes including shadow modes. Its use should be restricted to where direct split cursor screen editing is not required.

THE *NOCLOCK COMMAND:

This command cancels the effect of *RTC (display the clock), *ALARM and *BELL.

THE *ALARM COMMAND:

The *ALARM <HH:MM:SS> followed by the time in hours (0-23), minutes (0-59) and seconds (0-59) will set off a predetermined command or sequence of commands at the specified time. This command will be detailed further in section 3 "PROGRAMMING WITH THE REAL TIME CLOCK".

THE *BELL COMMAND:

The *BELL <HH:MM:SS> followed by the time, in hours (0-23), minutes (0-59) and seconds (0-59) will set the alarm time as specified. It will 'bleep' until a key is pressed.

An example of *BELL:

```
*BELL 07:30:00 <RETURN>
```

will wake you up (if you leave the computer switched on overnight) in the morning.

NOTES

*BELL can be issued after *RTC but *RTC or *ALARM or *NOCLOCK will all cancel BELL. These commands are based on regular UT2 interrupts. The latest command issued will use the interrupt to perform its function, thus cancelling each other.

Associated commands are *NOCLOCK, *ALARM, *RTC.

SECTION 3

THE CONFIGURE COMMANDS

What's configure?

You will find in the bottom right corner of the keyboard a set of holes into which you can fit 8 switches. These are called "Configuration switches" because their setting will determine the way the computer will behave when switched on. They set the default screen mode (from 0 to 7), the action of the <SHIFT BREAK> and also the stepping speed of your disc drives. On the MASTER, the battery powered Real Time Clock has some 50 bytes of RAM which extend the scope of the (still retained) configuration switches. One may expect the Solidisk Real Time Clock unit to perform a similar role as the unit built into the Master. It actually can do much more, as you will see later on.

The Real Time Clock RAM replaces the configuration switches — that saves you money and time, and also indicates clearly which switch does what.

It also stores a series of 'codes' which it sends to the computer everytime the computer is powered up or when the <CONTROL> key and the <BREAK> key are simultaneously pressed.

The list of 'codes' and conventions are of course closely related to the MASTER 128k as the latter is the inspiration for this unit.

For the technically minded, the action of the clock unit is as follows when the machine is switched on:

- issue *FX 255 to replace the configuration switches.
- issue *TV to suit the display unit.
- issue *FX 3, *FX 5, *FX 6 to suit the printer.
- issue *FX 7, *FX 5, *FX 156 to suit the RS 423 (MODEM).
- issue *FX 11, *FX 12 and *FX 202 to set up the keyboard.
- issue *FX 212 to set up the loud-speaker.
- set up the screen, shadow memory and VDU 14/15 to control the scroll option.
- unplug the ROMs if required.
- set up the system clock if required.
- initialise any filing system if required.
- enter any language if required.
- perform a list of user specified commands if required.

THE *CONFIGURE (*CON.) COMMAND:

This call is used either to list all the options or to configure the machine for start-up. If the clock has just been installed, and therefore the CMOS RAM is uninitialised, the configuration will be the default one. After a *CONFIGURE, however, this function will work and it is therefore necessary to ensure that the set-up parameters are sensible, otherwise the machine may lock up on subsequent hard breaks.

In order to list all the options available, then it is necessary only to type " *CONFIGURE", followed by RETURN.

THE OPTIONS ARE:

BOOT Reverse the <SHIFT><BREAK> action.
(Autostart with no shift).

CAPS Sets CAPS LOCK.

DATA (0-255) Sets the option of the RS423 word bits as in *FX156 **(see** the advanced User Guide for the bit patterns used), normal setting: 86.

DELAY (0-FF) Configures the auto-repeat wait time of the keyboard (0-255), normal setting: 50.

DIR Forces ADFS to load the CSD each time.

FDRIVE (0-3) Initialises the speed of the floppy disc controller (0-3).

FILE (0-8) Sets the filing system number of the filing system (0-8), where;

0 = the most prevailing filing system
1 = 1200 baud cassette
2 = 300 baud cassette
3 = ROM filing system
4 = Disc filing system (DFS)
5 = Econet filing system
6 = Telesoftware filing system
7 = IEEE filing system
8 = Advanced Disc filing system (ADFS)

IGNORE
(0-255) Issues *FX6, n to ignore a character in the print stream, normal setting: 10 or 255.

LANG n Forces the start-up Language to that in the specified socket, normal setting: BASIC.

LOUD Returns the volume of the bell setting to normal.

MODE n Forces mode to that specified (0-7 or 128-135). When the higher range is used, turns shadow RAM on where fitted.

NOBOOT Returns <SHIFT><BREAK> action to normal.

NOCAPS Enables lower case characters by default.

NODIR Enters FADFS rather than ADFS, does not load the Root Directory.

NOSCROLL Issues VDU 15, disabling scrolling.

NOTIME Leaves the BBC system clock unset, if TIME=0 on hard break or power up.

NOTUBE Disregards a second processor whether fitted or not.

PRINTER
(0-255) Sends printer output to the specified destination:

(0) printer sink
(1) parallel port
(2) serial port
(3) user printer routine
(4) network printer
(5-255) user printer routines
Default value =1

QUIET Forces a quieter volume to the bell setting (*FX212,224)

REPEAT
(0-255) Produces a faster (or slower) auto-repeat speed.

RX (0-8) Sets the RS423 receive baud rate as in *FX7 1-8.

SCROLL Enables scrolling by default (VDU15).

SHCAPS Sets SHIFT LOCK on.

TIME Loads the BBC system clock with the RTC time of the day. See BASIC TIME.

TUBE Enables the use of second processor.

TV m.n Issues a *TV command on hard break, normal setting is *TV 0,0. See *TV in user guide.

TX (0-8) Sets the RS423 transmit baud rate as in *FX8 1-8.

To change the configure options, type "*CONFIGURE", followed by the option it is required to change with a number as appropriate. The following example will set the bell to quiet, keyboard to lowercase, TV 255, 1, select ADFS as filing system, no root directory loading, printer to automatic LF after CR:

```
*CON. QUIET
*CON. NOCAPS
*CON. TV 255,1
*CON. FILE 8
*CON. NODIR
*CON. IGNORE 255
```

NOTES:

To cancel all actual `CONFIG' setting (or to return the machine to its default setting):

```
*FX 162, 12
```

Then press <CONTROL> BREAK.
Associated commands are *BOOT, *ALARM, *STATUS.

THE *STATUS COMMAND:

*STATUS displays the status of the above options for checking.
*SHOW is an alternative.

THE *UNPLUG COMMAND:

*UNPLUG, followed by a number (0-15), will disable the specified ROM the next time you press BREAK. . If the ROM to be unplugged does not have a Service entry or a Language entry (rom type 2 for example) then UNPLUG will have no effect. Unplugging the RTC ROM will also have no effect.

NOTES

Users of STL TOOLKIT/MANAGER ROMS will be able to restore temporarily the unplugged firmward with the *RWIPE command; it will be disabled again once the <BREAK> key is pressed.

THE *INSERT COMMAND:

*INSERT, followed by a number (0-15), will enable the specified ROM the next time you press <BREAK>.

THE *ROMS COMMAND:

*ROMS is a command which lists all the ROMS in the machine, together with their ROM type and their UNPLUG status.

*ALARM:

*ALARM HH:MM:SS will set the alarm to the specified time of the day (eg once per day). *ALARM with no specified time resets the alarm for the last setting. *ALARM uses the second user Timer (UT2) to check the status of the alarm every 50 milliseconds. When the alarm time is reached, the RAM boot process will begin execution.

*ALARM will not disturb the display of the clock. The alarm is turned off by *NOCLOCK.

Associated command: *BOOT, *NOCLOCK, *RTC.

*BOOT:

*BOOT <command string> will:

1) execute a star command on power up.
2) execute a string of commands on power up.
3) perform the same operation on time if the alarm is set.

*BOOT may be used in two different ways, for example:

```
*BOOT WORDWISE <RETURN>
```

will select WORDWISE on power up, rather than the 'CONFIG. LANG rom number' command. This has two advantages: easier to remember and easier to remove/replace.

This way of using *BOOT is call 'star format' as the machine will accept the command as a star command.

```
*BOOT CHAIN "S/D"||M
```

will CHAIN the program (assuming that BASIC is the default or configured language). The machine will program Key f0 as specified by the command string, then insert it into the keyboard buffer. We recommend this form, called the 'KEY form'.

The KEY will not be executed until all the Communication, Screen, Printer, File and Language are configured.

Key form also allows more flexibility for example:

```
*BOOT *PRESTEL 01 234 567 ||M
*BOOT *BE ||M L PROG ||M TOP ||M etc.. .
```

You type in *BOOT <command list> exactly as you would do with *KEY 10 etc...

Both forms work with *ALARM.

For example, to dial PRESTEL at 8.30 PM (assuming that you have Solidisk Linemaster ROM):

```
*BOOT PRESTEL 128                    (star form)
*ALARM 20:30:00
```

or alternatively (safer):

```
*BOOT *PRESTEL 128 IIM               (key form)
*ALARM 20:30:00
```

SECTION 4

PROGRAMMING WITH THE REAL TIME CLOCK

The Real Time Clock (RTC) appears to the programmer as 50 RAM locations and 14 Registers. Reading is by *FX 161, writing by *FX162. These FX's are official.

RAM location addresses are from 0 to 49 inclusive. Register addresses are from 128 to 141 inclusive.

PROGRAMMING THE RAM LOCATIONS:

The RAM locations can be read using *FX 161, X where X is the address. The value read is returned in Y.

ENTRY CONDITIONS:

A = 161   X = CMOS RAM address (0-49)   Y = don't care

EXIT CONDITIONS:

A and X preserved, Y value of (location X), 0-255.
Flags undetermined. A copy of Yis also deposited in the osYval, &F1.

From BASIC, use the USR (function to read:

```
10 X% = 30:A% = 161
20 ramval (30) = (USR(&FFF4) AND &FF00) DIV 256
```

WRITING TO RAM LOCATIONS

Use *FX 162 X Y to put contents of the Y register (Y) to CMOS RAM location X. For example, to write 65 to location 40, you would use: *FX162, 40,65. The first number after the FX call (162) will be passed to CPU register X, the second number to Y.

ENTRY CONDITIONS:

A =162 X = ADDRESS (0-49) Y = VALUE TO BE WRITTEN

EXIT CONDITIONS:

A, X and Y preserved. Flags undetermined.

From BASIC, use CALL, USR (or OSCLI:

```
   10 A% = 162: X%= 30:Y% = ASC"S"
   20 CALL &FFF4
  30
```

WARNING:

The Solidisk Clock & Desk ROM uses a variable amount of RAM locations to store the machine CONFIGURATION information. It is recommended that you read the following paragraph carefully before attempting to program the RTC. The Higher address you use (while less than 50), the less the risk of corruption to configuration data.

THE CONFIGURATION DATA

FUNCTION	**BYTE**	**D7**	**D6**	**D5**	**D4**	**D3**	**D2**	**Dl**	**D0**
System clock TIME setting	0	X							
ADFS or FADFS (NODIR)	0		X						
Floppy drive stepping	0		X ...	....	X				
Boot from disc option	0					X			
Screen mode (0-7)	0					**X**	......		X
Language ROM	1X	....	....	....	X				
Filing system (0-9)	1				**X**	......	......		X
Printer ignore (*FX6,X)	2X	....	....	....	....	....	....	......	 X
RS 423 Transmit rate (0-7)	3X	....	....	....	X				
RS 423 Receive rate (0-7)	3				**X**.........	......	......		X
Keyboard input delay (*FX11)	4X	....	....	....	....	....	....	......	 X
Keyboard repeat rate (*FX12)	5X	....	....	....	....	....	....	......	 X
TUBE option (1= ON)	6	X							
Shadow screen mode	6		X						
*TV, Y argument (0 or 1)	6			X					
RAM boot option (1= ON)	6				X				
Scroll (VDU 15)	6					X			
SHIFT keyboard	6						X		
CAPS keyboard	6							X	
QUIET (*FX 212,224)	6								X
*TV X argument (in *TV X Y)	7X	....	....	....	....	....	....	......	 X
Unplugged status of ROM 15	8 Xetc one bit per ROM								
ACIA control word (FX 156)	10X	....	....	....	....	....	....	......	 X
*FX 5 (printer status)	11X	....	....	....	....	....	....	......	 X
Checksum	12X....... .... ASC"S" XOR items 0-11 X								
RAM boot (*BOOT) string	13.......... .... variable								

It is not recommended to write to any of the first 13 locations (0-12 inclusive) or you will destroy the last setting of the machine unless you recalculate correctly the checksum byte. If the checksum is wrong, issue *SHOW (or *STATUS). The machine will automatically reconfigured in its default state, displaying the message:

`Machine unconfigured'

The B00T command string may be overwritten with no ill effect as its syntax is checked before its execution. Use *SHOW or *STATUS to display the command string (if any) and estimate its length.

THE 14 REGISTERS:

The Real Time Clock uses 14 registers to produce the Clock. They are summarily described below (comprehensive data sheets may be obtained from RCA, Motorola, or Hitachi):

REGISTER ADDRESS	NAME	RANGE	DESCRIPTION
128	TSEC	0-59	Time in seconds
129	ASEC	0-59	Alarm time in seconds
130	TMIN	0-59	Time in minutes
131	AMIN	0-59	Alarm time in minutes
132	THOUR	0-23	Time in hours
133	AHOUR	0-23	Alarm time in hours
134	TDAY	1-7	Day of the week (Sunday = 1)
135	TDATE	1-31	Day of the month
136	TMONTH	1-12	Month of the year
137	TYEAR	0-99	'86', the year.
138	XTAL	32	32,768kHz crystal
139	OPT	11	DON'T TOUCH, default options
140	STATUS	—	Clock status, see below
141	VRT		Valid RAM and Time, see below

You may write into ASEC, AMIN, AHOUR freely although it is recommended to use *ALARM HH:MM:SS instead. Never write directly into Txxx registers but use the TIME$ variable in BASIC 4, osword 15 or *SETTIME <date> <time> instead. This is because the VRT bit must be set after the Txxx registers are written to,otherwise you will get 'Time unset' error.

You may read the clock status (*FX 161, 140) any time. The only bit of interest is bit 5, the 'AF' (alarm) flag. If this bit is set, the alarm time has been reached. Issue *FX 161,140 as soon as you write into the Axxx registers to flush pending alarm. Reading the status register will clear the alarm status.

The last register, VRT, can be read with *FX 161,141. If bit 7 (of the Y register) is set, the contents of RAM are maintained correctly through switching off the computer or undo the RTC unit. The VRT bit is automatically cleared if the battery is flat. AVOID TOUCHING the CRYSTAL while removing the clock unit as it may turn off the VRT bit.

The RTC software ROM is capable of simple process control. The full capability of the RTC can only be exploited by directly programming the HD 146818. A simple interrupt processing is provided through *RTC (display the time) and *ALARM (set off a predetermined process at the indicated time), which can be repetitively re-asserted. If this is not fast enough, you will have to write your own software.

THE BASIC TIME VARIABLE:

This variable will be assigned the value of the BBC system clock. When it is used TIME is given in 100th sec. The option *CON. TIME will assign the TIME of the DAY to the system clock, thereby making it directly available to BASIC 1 and 2. The following program will illustrate another way of producing *TIME from BASIC 1, assuming that the TIME config option is set:

```
10 T = TIME
20 PRINT T DIV 360000;":";:T=T MOD 360000: REM hours
20 PRINT T DIV 6000;":";:T=T MOD 6000 : REM minutes
30 PRINT T DIV 100: RAM seconds.
```

The system clock is also available directly to machine code programs through osword 1.

THE BASIC 4 TIME$:

If you use BASIC 4, then the variable TIME$ (TIME string) is automatically assigned the 'Real Time' as kept by the RTC. Vice versa, assigning a new value to TIME$ will set the RTC accordingly.

The following program simulates the *RTC command:

```
10 REM display the time
20 PRINTTAB(0,29) RIGHT$(TIME$,16,8)
30 X=INKEY(100) : REM wait 1 second
40 GOTO 20
50
```

The following program simulates the *SETTIME command:

```
10 INPUT "Enter the time (HH:MM:SS) "Time$
20
```

"TIME$" may be incorporated directly in a program e.g.;

```
10 A% = OPENOUT "MYFILE"
20 PRINT #A%, TIME$
```

etc.

This will provide a quick and easy method of datestamping data files. Programs may also be as easily tagged thus:

```
10 REM **********************•******
20 $(PAGE+7)=T1ME$:?(PAGE+7+LEN TIME$)=32
30
```

Then run it.

This will produce:

```
10 REM *Thu, 08 May 1986. 10:35:27 ****
```

or, in assembler:

```
100 EQUS TIME$
```

Making the wondering of which program was the later version a thing of the past. However, if BASIC 4 is not available, there is a simple fix.

```
10 TIME$= FNTtime
2000 DEFFNtime
2010 A%= 14:X% = 0: V%=9
2020 CALL &FFF1
2030 = $&900
```

This function will produce the time$ just as in BASIC 4.

OSWORD 14 and OSWORD 15

Two new OSWORLD commands have been made available for the reading and writing of the TIME function. These are OSWORD with the accumulator set to 14 for reading and 15 for writing. Simply, the Y and X registers point to a control block in memory to receive the clock data for read or containing the parameters for write.

In read mode, the text string is simply deposited in memory as pointed to by the osword parameter block: in write mode, however, the first byte contains the length of the string and the following bytes are the string itself.

No string terminator is needed in the latter case, although a carriage return is supplied by osword 14.

Example of osword 15:

```
10 DIM clock (30)
20 T$="MON, 1 JAN 89.09:00:00"
30 ?clock=LEN T$
40 $(clock+1)=T$
50 X%=clock MOD 256:Y%=clock DIV 256
60 A%=15:CALL &FFF1
70
```

SECTION 5

THE DESK COMMANDS

THE *DALARM COMMAND:

*DALARM — abbreviation for D(esk) Alarm, provides a simple way to get the computer to keep your appointments whilst retaining the full use of your computer.

The command once issued, will only act when the preset ALARM time has been reached.

It then momentarily interrupts the current program to display a pre-recorded message, produce a ringing bell and wait until you press a key. Control is then returned to the current program, until the next APPOINTMENT.

All you have to do is key in the time and the message into a file called "$. APPIONTS", and then issue *DALARM.

The command *DALARM <RETURN> will:

1) check that the $.APPOINTS file is present, if not, print the error message "$.APPOINTS not found".
2) set *DALARM <time of the next appointment>

The command *DALARM HH:MM:SS <RETURN> will simply set the ALARM for the specified time.

If you include *DALARM in a *B00T command as follows: *B00T

*DALARM ||M

This will automaticaly set up the D(esk) Alarm for you while the computer is in use.

Example of *DALARM:

The file "$.APPOINTS" consists of a list of entries in the following form, which you can make up by entering *BUILD $.APPOINTS (View or WORDWISE can equally be used):

Key this in:

Entry 1:	**Example:**
Any text...	Things to do Fri, 13/07/86
@ <time of the first event >	@09:00:00
Message..	Ring J. SMITH, LONDON (01) 123 456.
Entry 2:	
@ <time of the next event >	@09:30:00
Message	Ring Solidisk Find out any new add-ons at the next show.

Entry **3:**

```
@ <time of the next event >          @15:00:00
Message ................................................ etc...
etc...
                              <ESCAPE<
```

Now enter:

```
*SETTIME 08:59:00
*RTC
*DALARM
```

You should see the clock displayed and the message:

'The next appointment will be at 09:00:00'

At precisely 09:00:00, you should hear the bell ringing and see:

```
APPOINTMENT:
Fri, 13 Jul 1986.09:00:00
>>> URGENT <<<
Ring J. SMITH
LONDON (01) 123 456
Press any key
```

Press any key.

If you leave it until 9:30:00, the next message will come up etc...

THE *DESK COMMAND:

The *DESK command offers a nice way to start a program. Using a Mouse, it requires the AMX Super ART ROM in the machine to drive the mouse or the tracker ball. Solidisk will release soon a Mouse —Tracker ball driver ROM for the BBC computer. * DESK (or *ZDESK if the AMX ROM is installed in any higher priority socket than the Clock & Desk ROM) starts the DESK. You may enter the DESK automatically when the computer is switched on by including it in the *BOOT command:

```
*B00T *DESK ||M
```

Four main choices are:

1) DISC MENU
2) CALCULATOR
3) APPOINTMENTS
4) TIME

To make a selection, POSITION the pointer over the display, which will be automatically reversed, PRESS the 'EXECUTE' button (or f0 key if you are using the Keyboard).

To CANCEL a selection, PRESS the 'CANCEL' button (or key f2 if you use the Keyboard). Pressing CANCEL several times will take you out of DESK and back to the default language.

DISC MENU:

Move the pointer to FILE and select MENU.

This will display the disc directory, in DFS and ADFS. Moving the pointer to any filename will bring down a SUB MENU, with options to SHOW, RUN and DELETE. Move the pointer to any SUB OPTION to execute the command.

Moving the pointer to a Directory Name will select it as the CSD and renew the DISPLAY.

Move the pointer to 'NEW DISK' to start again after a disc change or go back to the R00T directory.

Move the pointer to any 'DRIVE' icon on the bottom of the display to change the current DRIVE.

CALCULATOR:

Move the pointer to OPTION, then select CALCULATOR. Enter numbers by selecting keys on the calculator keypad. Select "=" to display result of the operation. CALCULATOR will work with both BASIC 1 and BASIC 2, and with all maths functions.

APPOINTMENTS:

Move the pointer to 'OPTION' then select APPOINT. This will allow you to edit the $.APPOINTS file with ease.

APPOINT will display the current contents of the '$.APPOINTS' file. Move the pointer up or down to change the display. Enter new entries if required from the keyboard.

TIME:

Move the pointer to the CLOCK box. POSITION the pointer over ARROW LEFT or ARROW RIGHT in front of the HOURS MINUTES or SECONDS. Press the 'EXECUTE' button to increment or decrement the time. Select 'SET' to write the time to the Real Time Clock.

BBC Computer in this manual refers to the computer produced by Acorn Computer Limited for the British Broadcasting Corporation.