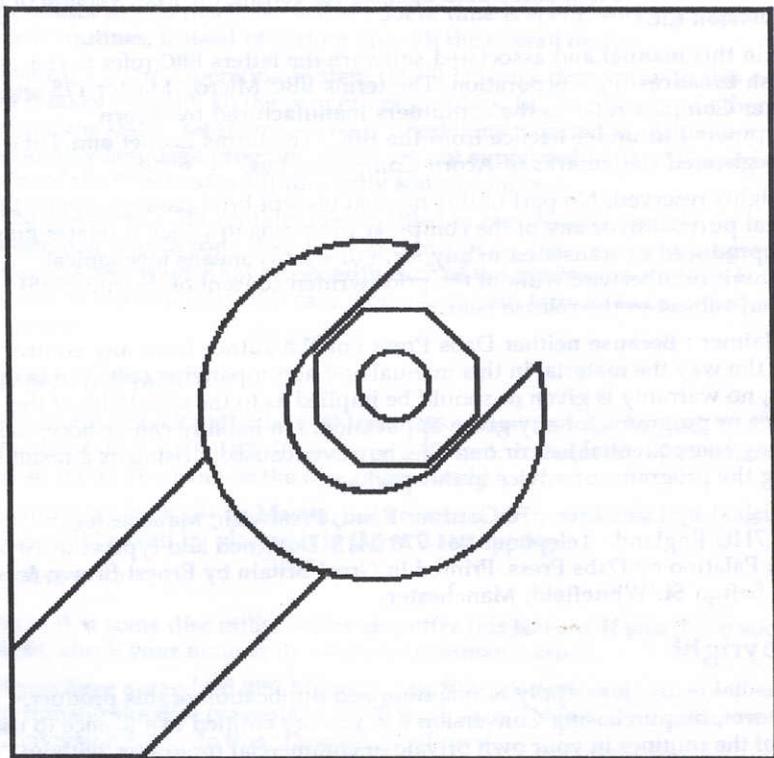


CONVERSION KIT



UTILITY SOFTWARE FOR BBC B/B+/MASTER SERIES

BRUCE SMITH

**DABS
PRESS**

Conversion Kit

© Bruce Smith 1987

First published June 1987

ISBN 1-87033-606-2

The author wishes to thank David Spencer for coding the ROM version of the Conversion Kit.

Within this manual and associated software the letters BBC refer to the British Broadcasting Corporation. The terms BBC Micro, Master 128 and Master Compact refer to the computers manufactured by Acorn Computers Ltd under licence from the BBC. The terms Econet and Tube are registered trademarks of Acorn Computers Ltd.

All rights reserved. No part of this manual (except brief passages quoted for critical purposes), or any of the computer programs to which it relates may be reproduced or translated in any form or by any means mechanical electronic or otherwise without the prior written consent of the copyright holder, subject to the release below.

Disclaimer : Because neither Dabs Press nor the author have any control over the way the material in this manual and accompanying software is used, no warranty is given or should be implied as to the suitability of the advice or programs for any given application. No liability can be accepted for any consequential loss or damage, however caused, arising as a result of using the programs or advice in this package.

Published by Dabs Press, 76 Gardner Road, Prestwich, Manchester M25 7HU England. Telephone 061-773 2413. Designed and typeset in 10 on 11 pt Palatino by Dabs Press. Printed in Great Britain by Ernest Brown & Son, Sefton St. Whitefield, Manchester.

Copyright

The usual restrictions apply to unauthorised duplication of this product. However, by purchasing Conversion Kit, you are entitled to a licence to use any of the routines in your own private or commercial programs, without further payment. If you are releasing a commercial product, please contact us for a formal licence which will be issued free of charge. This paragraph is not in itself a commercial licence. Our address is given above.

Introduction

Conversion Kit isn't like other software packages - it doesn't perform any useful function in itself, but is a collection of 6502 assembly language routines for programmers to use in their software. A difficulty often faced by programmers, especially novices, and those not primarily trained on this processor or this machine, is that a lot of time is spent working out fairly basic routines, instead of getting on with the overall design.

The BBC Micro Operating System (MOS) is better than many in this respect as it takes virtually all the pain out of graphics, file handling and other mundane tasks. Yet there are many operations required in virtually every assembly language program which are not supported by the MOS. It is the aim of the Conversion Kit to supply some of these.

The Conversion Kit routines are supplied in two distinct formats, as assembler source code that you can incorporate into your own programs and as a Sideways RAM image with each of the conversion routines being accessed with an OSWORD call. The use of both formats is described in this manual.

Disc formats

The Conversion Kit disc is in 40 track DFS format and can be read by any Acorn or third party DFS. If you have a 40 track drive (or 40/80 switchable set to 40 track) you can use the disc immediately.

If you have a BBC B+ or Master, but are using 80 track drives only, the disc may still be read by placing it in Drive 0 and typing:

```
*DRIVE 0 40
```

Note that some disc utility ROMs also offer this feature. If you have such a ROM, check your manual to see if this command exists.

If you have a standard BBC Micro and an 80 track drive only, you will need to transfer the disc contents to an 80 track disc. There is a program on the Conversion Kit to do this. Proceed as follows:

- 1 Format a new 80 track disc.
- 2 Insert the Conversion Kit disc, and type:
CHAIN "COPY40"
- 3 If you have dual drives, put the 80 track disc in the other drive.
- 4 The program will request details of the whereabouts of the source disc (40 track) and destination disc (80 track). Enter the drive numbers pressing RETURN as required. If you only have a single drive enter 0 for both source and destination drives.

- 5 With dual drives the programs will be transferred automatically - a process that will take minute or two. With single drives you will need to be on hand to swap discs when requested to do so.
- 6 When transfer is complete and the 80 track disc has been tested, the 40 track disc may be kept as a backup or if you prefer re-formatted to 80 tracks to replace your original 80 track disc.

Also, if you can read 40 and 80 track discs, you can make an 80 track disc by merely *COPYing the files from one disc to the other.

Files on the Conversion Kit disc

There are 30 files on the Conversion Kit disc. 24 of these are Conversion Kit source routines, explained below. The other six are:

!BOOT A !Boot file to CHAIN "MENU"
 CKITROM The ROM version of the Conversion Kit
 COPY40 A program to convert to 80 track discs—see above.
 MENU A menu program for the source routines
 ROMEXPL An example program using the ROM routines
 ROMSRCE The ROM version source code (using BASIC's assembler)

Using the Conversion Kit Source Routines

To use the Conversion Kit, insert the disc, and hold down SHIFT, press BREAK, release BREAK and then release SHIFT. A menu program (called MENU) will be loaded providing a brief description of the programs on the disc. The menu may be browsed by use of the up and down arrow keys, and a program loaded by pressing the RETURN key.

With a couple of exceptions, all programs contain a front end which is designed to demonstrate the program's operation. You will be asked to enter values in many cases that will be used to demonstrate the programs.

The machine code routines are all stored within a PROCedure, normally PROCass, and this can be appended to your own assembly language programs as and when required. This is done by loading the required program and deleting all lines other than the PROC itself. This can then be *SPOOLED to disc by typing the following BASIC commands:

```
RENUMBER 10000
*SPOOL S.assem
LIST
*SPOOL
```

RENUMBER will renumber lines within the PROC to avoid line number clashes when the program is re-loaded. Enter your own program and then *EXEC the *SPOOLED program thus:

Examine the original demonstration program for details on using the routine. In most cases the assembly language is fully documented.

The Programs

A brief description of the programs on the Conversion Kit is given below. In some cases, further documentation is given in the programs themselves. Look at the program listings to see where the input and output parameters are stored.

- ASCHEX** converts the ASCII hex number in the accumulator into binary. For example, if the accumulator contained ASC"F" then the routine will return 15, ie, &0F.
- 4ASCHEX** converts four ASCII hex digits into a 16 bit binary value. For example, if the four ASCII hex digits were ASC"F" ASC"F" ASC"F" ASC"F" the 16 bit binary value &FFFF will be returned in locations &70 and &71.
- DEC2BN** this is similar to 4ASCHEX except that a 1 to 5 digit decimal string may be entered and this is converted into a 16 bit binary value. Thus entering: ASC"5" ASC"1" ASC"2" will return &200 at the location pointed to by the variable 'current'.
- BN2DEC** takes a 16 bit binary value and returns a signed decimal ASCII string. Binary values in the range &0 to &7FFF will return positive decimal strings, whereas values &8000 to &FFFF are treated as negative strings. &FFFF will return -1, &FFFE will return -2 etc.
- BTOD** takes a 16 bit binary value and prints a decimal ASCII string to the screen. Loading the buffer with &FFFF will return the ASCII characters "65535".
- REALINT** takes a 5 byte floating point binary number in BASIC's format and returns the equivalent 4 byte integer value.
- INTREAL** takes a 4 byte integer value and returns the equivalent 5 byte floating point value, in BASIC's format.
- BININ** converts an 8 bit ASCII string of 1s and 0s into binary. Entering 11110000 will return &F0.
- BINOUT** converts the value in the accumulator into an ASCII binary string which is printed. If the accumulator holds &81 then "10000001" will be printed.
- COMPARE** compares two 16 bit numbers. Returns:
&01 if first is greater
&03 if values are equal
&80 if second is greater

- COPYSUB** copies a substring from a main string, ie, enter CONVERSION as main string, enter start point of 3, enter number of characters to be copied, ie, 4 and the substring returned is VERS.
- INSERT** Inserts a substring into a main string. Enter main string of CONION, substring to be entered=VERS, start index=3 and new string is CONVERSION.
- CONCAT** concatenates a second string onto a first string. Note that to insert any spaces, they should appear at the end of the first string.
- BUBBLE** performs a machine code bubble sort. Data to be sorted should be stored within program DATA statements (or poked into array). Data is sorted into ascending order.
- SAVESR** saves current graphics mode screen as "SSAVED".
- LOADSR** loads screen called "SSAVED" into graphics memory.
- SPTI** prints the required number of strings including control codes as so desired.
- USTACK** implements a User Stack similar to that on the 6809 processor. The registers pushed are the program counter, the status register, the accumulator, X and Y registers, plus three other user defined 16 bit registers. These pseudo registers are:
 PR1=&80,&81 PR2=&82,&83 PR3=&84,&85
 The byte after the user stack subroutine call defines which registers are to be pushed as follows:
 bit0=PR1, bit1=PR2, bit2=PR3, bit3=Y, bit4=X, bit5=Acc, bit6=Status, bit7=PC
 To push all registers the byte would be 11111111=&FF. To push accumulator, index and status registers the byte would be 00011110=&1E.
- ADDSUB** Multibyte addition and subtraction routines.
- MULTDIV** Multibyte multiplication and division routines.
- CLOCK** displays an interrupt driven clock at the top of the Mode 7 screen. This display will not work with other modes, or if Shadow RAM is selected.
- CRUNCH** compacts a Basic program.
- GLOB** a global search and replace facility, ie, to change variable names in a Basic program. Use f0 to start. When run you are asked for variable name, and the name it is to be changed to. The number of occurrences are then displayed.

MEMOVE an intelligent memory move program. RUN and enter as follows:
start address:&3000 Destination:&5800:Length:&2000
to see screen memory moved (with Shadow RAM disabled if installed).

Using the Sideways RAM Image

Many of the conversion routines are also included in a ROM image called CKITROM on the disc. In order to use the Conversion Kit in this form it is necessary to have at least 8k of Sideways RAM (SRAM) fitted to your computer. This is available as standard on the B+128 and Master Series machines, and can be added as an upgrade to a Model B or B+.

Before use it will be necessary to load the ROM image from disc into the SRAM. If you have a B+ or Master computer this can be done using the following command:

```
*SRLOAD CKITROM 8000 W
```

This will load the image into the lowest RAM bank, see your User Guide for more details. If your micro is fitted with third party SRAM then the loading technique depends on the type of RAM and details can be found in the RAM user guide. Once the ROM image has been loaded you should press CTRL-BREAK to initialise the ROM.

All the routines in the ROM version are called by means of a single OSWORD call. The first byte of the parameter block determines the function while the remainder of the parameter block is used to pass the arguments to and from the routine.

The OSWORD call used by Conversion Kit is 96 (&60) so the A register should be loaded with this, and the X and Y registers should be set to point to the parameter block (X-lo, Y-hi) before calling OSWORD. From BASIC these registers can be set up by putting their values in the variables A%, X% and Y% respectively before issuing:

```
CALL &FFF1
```

to execute the routine. Obviously it is necessary to define the parameter block before calling the routine. A typical BASIC program could call a routine in the ROM quite simply as this short programs demonstrates.

```
10 DIM block% 10 : REM Allocate space for parameter block
20 A% = &60
30 X% = block% MOD &100 : REM Point to parameter block
40 Y% = block% DIV &100
50 ?block% =opcode : REM Specify which routine
60 REM {Set up rest of parameter block depending on routine}
70 CALL &FFF1
```

A description of each of the routines supported by the Conversion Kit ROM image, along with their opcodes, the format of the parameter block on entry, and the parameters returned is given below. While all the routines supported perform the same function as their equals in the stand alone routines because of the use of OSWORD the method of using some of the routines is slightly changed.

The Conversion Kit routines use zero page memory between &A8 and &AF as general purpose scratch space, and the bottom of the stack area (&100 upwards) as buffer space. This choice will not normally interfere with your own program's workspace.

If OSWORD &60 is unsuitable ie, if it clashes with another firmware product, it is a simple matter to change it, by altering the variable 'ourword' in the file ROMSRCE.

A list of the available routines together with their opcodes can be found by typing

*HELP CONVERSIONS

(or *H.C.) when the ROM image has been loaded.

A program is provided on the disc, called ROMEXPL which demonstrates each one of the ROM version calls in turn.

Sideways ROM/RAM version compatibility

All the routines in the ROM version of the Conversion Kit with the exception of BUBBLE and MEMOVE are Tube compatible. If a Second or Co-processor is active the routines will automatically access the Second processor memory. MEMOVE and BUBBLE will always use I/O processor memory regardless of the presence of a Second processor.

Conversion Kit OSWORD Calls

All 16 and 32-bit quantities are in the usual lo-hi format.

ASCHEX Convert single ASCII character to binary

Entry: XY = 0
XY+1 = ASCII character to convert (ASC"0"-ASC"F")

Exit: XY+1 = Hex value of character
(&FF if illegal character)

ASC4HEX Convert ASCII string to 2 byte binary

Entry: XY = 1
XY+1,2 = Pointer to ASCII string

Exit: XY+1,2 = Pointer to character after string read
XY+3,4 = 16 bit value of hex string
XY+5 = Error flag

&00 - No error
 &FF - An error occurred

DEC2BN Convert ASCII decimal string to two byte binary
 Entry: XY = 2
 XY+1 = Length of string to convert
 XY+2,3 = Pointer to string in memory
 Exit: XY+1,2 = Two byte result
 XY+3 = Error flag:
 &00 - No error
 &FF - An error occurred

BN2DEC Convert two byte value to ASCII decimal
 Entry: XY = 3
 XY+1,2 = Pointer to memory for result
 XY+3,4 = Value to convert
 Exit: Parameter block unchanged
 Pointer contains string length, followed by ASCII bytes

BTOD Print out two byte value in decimal
 Entry: XY = 4
 XY+1,2 = Value to print
 Exit: Parameter block unchanged
 Decimal string is printed

BININ Read single byte in binary from keyboard
 Entry: XY = 5
 Type eight binary digits (others keys ignored)
 Exit: XY + 1 contains value entered

BINOUT Print byte a 8 binary digits
 Entry: XY = 6
 XY + 1 = Byte to print
 Exit: Parameter block is unchanged
 Binary string is printed

INTREAL Convert integer to real
 Entry: XY = 7
 XY+1,4 = 32 bit signed integer (lo-hi)
 Exit: XY+1,5 = 40 bit floating point result (format as BASIC)

REALINT Convert real to integer
 Entry: XY = 8
 XY+1,5 = 40 bit floating point value

Exit: XY+1,4 = 32 bit integer result
XY+5 = Error flag:
&00 - No error
&FF - Overflow occurred

Multiple precision integer arithmetic calls. These calls will perform integer arithmetic to any precision, but of course, BASIC, and the INTREAL call cannot deal with any precision greater than four bytes.

ADD Add two integer numbers

Entry: XY = 9
XY+1 = Length in bytes of each argument
XY+2,3 = Pointer to first argument (and to result)
XY+4,5 = Pointer to second argument

Exit: First argument is replaced by result
Parameter block unchanged

SUB Subtract the second integer number from the first

Entry: XY = 10
Other parameters as ADD

Exit: As ADD

MULT Multiply two integer numbers

Entry: XY = 11
Other parameters as ADD

Exit: As ADD

DIVIDE Divide first integer number by second

Entry: XY = 12
Other parameters as ADD

Exit: As ADD

COMPARE Compare two 16 bit values

Entry: XY=13
XY+1,2 = First 16 bit integer value
XY+3,4 = Second 16 bit integer value

Exit: XY+1 = Result code as follows:
&01 - First is less than second
&03 - Arguments are equal
&80 - First is greater than second

COPYSUB Copies a substring from a main string

Entry: XY = 14
XY+1 = Length of main string
XY+2 = Length of string to copy
XY+3 = Position to copy from (First char is 0)

XY+4,5 = Pointer to main string
 XY+6,7 = Pointer to store substring

Exit: XY+1 = Error flag:
 &00 - No error
 &FF - Unable to copy whole string
 XY+2 = Length of actual string copied

INSERT Insert a substring into a main string

Entry: XY = 15
 XY+1 = Length of main string
 XY+2 = Length of substring
 XY+3 = Position to insert *before* (first char is 0)
 XY+4,5 = Pointer to main string
 XY+6,7 = Pointer to substring

Exit: XY+1 = Length of new string
 XY+2 = Error flag:
 &00 - No error
 &FF - String couldn't be inserted

CONCAT Concatenate two strings

Entry: XY = 16
 XY+1 = Length of first string
 XY+2 = Length of second string
 XY+3,4 = Pointer to first string (and result)
 XY+5,6 = Pointer to second string

Exit: XY+1 = New length
 XY+2 = Error flag:
 &00 - No error
 &FF - Couldn't add strings

SPTI Print a sequence of strings

Entry: XY = 17
 XY+1 = Number of strings to print
 XY+2,3 = Pointer to strings, terminated by &0D

Exit: Parameter block is unchanged
 Strings are printed

SAVESR Save a graphics screen

Entry: XY = 18
 No other entry parameters

Exit: Screen is saved under the filename SAVED. If in Mode 3 or 6 a warning message is printed instead. The message is not an error and cannot be trapped. If in Mode 7 the 1k text screen is saved. Normal filing system rules apply, and filing system errors, eg, 'Disc full', may occur.

- LOADSR** Load a graphics screen
- Entry: XY = 19
No other entry parameters
- Exit: The file SSAVED is loaded to the screen. If in Mode 3 or 6 a warning message is printed instead. The message is not an error and cannot be trapped. If the file is not present the filing system will generate a normal 'Not found' error.
- MEMOVE** Move a block of memory
- Entry: XY = 20
XY+1,2 = Start address of source memory
XY+3,4 = Start address of destination memory
XY+5,6 = Length of block to move
- Exit: Parameter block is unchanged
Specified block of memory is moved
- BUBBLE** Bubble sort a set of *bytes*
- Entry: XY = 21
XY+1 = Number of bytes to sort
XY+2,3 = Pointer to start of data
- Exit: Parameter block is unchanged
Bytes are sorted into ascending order

The Conversion Kit ROM occupies 4k of ROM space, and thus could easily be incorporated into a larger ROM program. The file ROMSRCE contains a standard ROM header, but no provision has been made for *^** commands and the like. Note that the program as supplied uses &C000 as a memory buffer, for a 6502 second processor. If you are using a non-Tube machine, you will have to alter this. There isn't sufficient memory on a standard Model B to assemble the program. On a BBC B+ or Master 128, you will need to be in a shadow mode. If you are writing a ROM-type program for the first time, you will probably need a text showing how ROMs work in more detail. My own books *The BBC Micro ROM Book*, (Collins £9.95) and *The Advanced Sideways RAM User Guide* (Victory Publishing £9.95) address this topic, as does *The Advanced User Guide for the BBC Microcomputer* (Bray, Dickens and Holmes, Cambridge Computer Co. £14.95), and our own *Master Operating System: A Dabhand Guide* (see below).

About Dabs Press

Dabs Press is a computer book and software publishing house, publishing utility and business material for the BBC Micro. We hope you enjoy using Fingerprint and that it helps you to write better programs. We would like to hear from you, whether you wish to compliment or castigate us. Only by user reports can we completely eradicate bugs. Details of our other products are given below, together with ordering details—or you might like to contact us for an up-to-date list which contains more detail than we can fit in here. If you have an idea for a computer book or software package, then we would love to hear from you. We have a leaflet giving hints on how to present your submission.

Other Dabs Press products

The products marked * are in preparation at the time of writing and are subject to content change and price alteration without notice. They are due to appear between September 1987 and March 1988.

VIEW: A Dabhand Guide by Bruce Smith

This 248 page book is a complete tutorial and reference guide to the Acornsoft VIEW wordprocessor, covering everything from introductory wordprocessing to advanced techniques. The accompanying disc contains several utility programs which are printed in the book, and other programs of interest to VIEW users, including a Printer Driver Generator. The book also covers ViewSpell and ViewIndex. Price £12.95 (book), £7.95 (disc), £17.95 (book and disc if ordered direct).

Master Operating System: A Dabhand Guide by David Atherton

At 272 pages, this, the *only* advanced User Guide for Master, Compact and BBC B+ owners contains detailed information on all system calls and '*' commands as well as much previously unpublished information on the Tube, CMOS RAM/EEPROM and other new features. The accompanying disc contains the many utility programs in the book, with several extra programs, including a utility ROM image for programming as an EPROM or loading into Sideways RAM. Price £12.95 (book), £7.95 (disc), £17.95 (book and disc if ordered direct).

ViewSheet and Viewstore : A Dabhand Guide by Graham Bell *

This book is a complete tutorial and reference guide for the Acornsoft ViewSheet spreadsheet and the ViewStore database manager. Every aspect

of setting up and using a database or spreadsheet is described in detail, and numerous examples are provided to guide you. The accompanying disc contains a number of utilities including programs that join two databases together and help transfer spreadsheets into a word processor. Price £12.95 (book), £7.95 (disc), £17.95 (book and disc if ordered direct).

Fingerprint by David Spencer

Fingerprint is a single-step machine code tracing program. It allows you to step through machine code written by yourself, part of a commercial program, or even the operating system, examining and executing each instruction, and displaying the contents of registers and flags at every stage. Unlike many other monitors Fingerprint will even trace code situated in Sideways RAM— watch and learn how BASIC works for instance! Also included is a complete 20/40/80 column memory editor, accessing all paged memory on all machines, a powerful disassembler, and a ROM lister. Price £9.95.

Floating Point Assembler : A Dabhand Guide by David Spencer *

The first ever programming tutorial on floating point arithmetic in machine code. Learn the impossible—step by step details on floating point addition, subtraction, multiplication, division, sines, cosines – in fact everything. Lots of examples including multiple and double precision arithmetic. Another Dabs first! Price £14.95 (book), £7.95 (disc), £19.95 (book and disc if ordered direct). Versions for other microprocessors planned.

Bumper Assembler Bundle by Bruce Smith

If programming in machine code is just wishful thinking, here is the chance for you to explore this fascinating subject with our Bumper Assembler Bundle. We are able to bring to you a five-part package of assembly language materials (two books, two discs, and booklet) at less than a third of their normal price - all from the prolific keyboard of BBC expert, and former Acorn User Technical Editor Bruce Smith.

You get *BBC Micro Assembly Language*, an excellent 200-page introduction and tutorial guide to 6502 assembly language programming. *BBC Micro Assembler Workshop* containing a feast of utilities and conversion programs for assembly language programmers, two discs, containing a total of 90 programs from the two books, and a booklet detailing all the new chip commands and innovations supplied with the Master 128 and Compact.

All together this bundle would cost you over £30. We can offer it to you at an exclusive price of just £9.95 including p&p and VAT. An 3.5in ADFS version is also available at the same price.

Master 512 : A Dabhand Guide by Gordon Taylor *

At last a book for users of the Master 512 co-processor. Gordon Taylor of A&B Computing explains how to use your co-processor as an IBM compatible clone. Full details on using the software provided with the 512 and how to ensure that huge base of PC software will work for you to best effect. Topics covered include the 80186 processor, compatibility, DOS Plus, GEM applications, Model B adaptors, and the Solidisk PC Plus. Price: £14.95. A disc is planned to accompany this title. Details will be announced at a later date.

VIEW Family Applications: A Dabhand Guide by Bruce Smith and Graham Bell *

Bruce Smith and Graham Bell team up to provide a whole range of expert utilities for users of any VIEW products. Includes automatic file backup, front ends to the VIEW family, Sideways RAM software, how to write a ViewStore utility, advanced printer and screen drivers plus many more items. This is a software pack with detailed manual. Price to be announced.

Master Emulator ROM by David Spencer *

This new ROM software is especially for Model B and B+ owners, and provides you with most of the features of the Master 128, such as the new ** commands, the extended filing system operations including the temporary filing system, the *CONFIGURE system (using battery-backed Sideways RAM and/or a disc file), and if you have the hardware, Sideways or Shadow RAM. The only Master Operating System software not covered in this ROM is the extended graphics software. With this ROM you'll be able to use the new features documented in the Master Operating System, and for third-party shadow and Sideways RAM users, run programs which are normally only intended for Masters. Price : to be announced.

Ordering

All prices quoted are inclusive of VAT where applicable. Books and export orders are zero-rated for VAT. We accept orders accompanied by cheque, postal order, or official orders (PLCs, public sector and education only). We also accept Access/Mastercard by post and telephone.

When ordering, please advise which BBC Micro you have, whether you want 3.5in or 5.25in discs, and whether you need a VAT receipt. All orders are post free in the UK. Customers abroad should add £3 for surface mail, and £7 for airmail.

Send your order to Dabs Press, 76 Gardner Road, Prestwich Manchester M25 7HU. Tel: 061-773 2413. If you just want product information, telephone us and we'll send you the up-to-date list.