PMS

USER GUIDE

for the

BBC
SECOND
PROCESSOR

B2P-6502

User Guide

for the

B2P-6502 Second Processor

for the BBC Microcomputer System

Gordon Cameron BSc.

Derek Mathieson

Permanent Memory Systems

Permanent Memory Systems is the trading name of Ganarch Limited.

Please note that within this text, the terms TUBE and Hi-BASIC are registered trade names of Acorn Computers Ltd.

BBC Hi-BASIC is produced under license by PMS from Acorn Computers Ltd.

B2P Hi-WORDWISE PLUS is produced under license by PMS from Computer Concepts.

The B2P is guaranteed against defect due to manufacture for *a* period of 6 months from the date of purchase. Any misuse of, or modification made to the B2P during this time will invalidate this guarantee.

This User Guide was prepared using Hi-WORDWISE PLUS running on one of PMS's amazing B2P-6502 Second Processors. The B2P can hold the entire 33K (yes - more than 33000 bytes!!!) of this User Guide and preview it in MODE 0.

Printing was on *a* standard Epson RX80. using PMS Multi-Font NTQ Printer Driver.

CONTENTS

Second Processor Addressing

Using The B2P From Machine Code

The BOS ROM

Appendices

Last Words

CHAPTER ONE

INTRODUCTION

1.1 Checklist of Items

As well *as* this User Guide, you should have the follwing items in the box you have just opened - the B2P Second Processor and the BOS ROM. Should either of these items are damaged or missing contact PMS without delay.

1.2 What is a Second Processor?

The Second Processor is basically another computer, with its own RAM and microprocessor, which runs in parallel with your existing computer. The effect is the same *as* when two engines are used to pull a train, a larger load can be pulled at higher speed because the workload is shared. Similarly when a Second Processor is added to the BBC, the workload is shared between the two processors thus increasing both speed of operation and the amount of available memory.

With the PMS B2P 6502 Second Processor fitted, the BBC becomes the INPUT/OUTPUT Processor (I/O), handling the time-consuming and memory grabbing tasks such as controlling disks, keyboard and screen, while the B2P gets on with the business of running the current language eg BASIC. VIEW etc. Communication between the B2P and I/O Processor is achieved by using the TUBE. The passing of information between to two processors is done under complex protocols, which are implemented by the B2P Operating System (the BOS).

From now on, the term I/O Processor refers to the BBC. and the term Second Processor refers to the B2P.

INSTALLATION

The following section deals with the installation of the B2P and BOS
ROM. When using the Second Processor, the BOS ROM MUST be present in
the I/O Processor. If you are short of ROM sockets, and do not wish
to fit the BOS ROM, the B2P Operating System may be loaded into a
bank of sideways RAM, in which case consult APPENDIX I - "
Transferring the BOS into sideways RAM".

2.1 Installing the BOS ROM

The BOS ROM can be plugged into any sideways ROM socket, either on
the main BBC circuit board or on an extension ROM board.

To install the BOS ROM in the main BBC board proceed as follows:

(1) SWITCH OF THE POWER TO THE BBC

(2) Remove the FOUR *screws* (which may be marked FIX), on the
    underside of the BBC.

(3) Remove the TWO or THREE nuts and bolts securing the keyboard,
    and slide it back to reveal the ROM sockets.

(4) On the bottom right-hand side of the main circuit board there are
    five 28 pin sockets. The BBC's OS is in the left-most socket and
    two others will be occupied by BASIC and the DFS. The BOS ROM
    can be inserted into any empty 28 pin sockets. If all the
    sockets are filled with ROMs then you will have to either remove
    a ROM, or purchase an extension ROM board.

(5) The BOS ROM MUST be fitted with the notch towards the REAR of the
    BBC, ie the same *way as* the other ROMs present. If the pins do
    not line up with the socket, bend the pins of the ROM by
    pressing the legs against a flat surface eg a table top. Check
    that none of the legs have missed the socket and that none are
    bent under the ROM.

(6) Replace the keyboard, and the screws holding the lid, and SWITCH
    ON the BBC.

(7) If the BBC does not start up as normal SWITCH OFF immediately,
    and check that the ROM is correctly installed.

(8) Type *HELP. The title string of the BOS ROM should appear, if not check
    that the ROM is correctly installed.

2.2 Plugging in the B2P Second Processor.

Power to the BBC must be OFF before plugging in the B2P.

The B2P has two pieces of ribbon cable attached. The broader cable is terminated by a 40-way header which should be plugged into the TUBE socket on the BBC. The TUBE socket is the right-most socket under the BBC. Insert the header with the BUMP UPWARDS, and push home firmly.

The 2-way cable from the B2P is connected to pins 11 and 12 of the 1MHz BUS. The 1MHz bus socket is the second socket from the right under the BBC, immediately to the left of the TUBE. If the 1MHz bus is already being used for another device, eg TELETEXT adapter, consult APPENDIX II - "NPFD signal from BBC board". Pins 11 and 12 of the 1MHz bus are the SIXTH vertical pair of pins along the 34 way connector, counting from the RIGHT-HAND side of the socket. With the wire coming out of the RIGHT-HAND side of the 2 pin plug, insert the plug onto pins 11 and 12. and push in firmly.

FIGURE 1
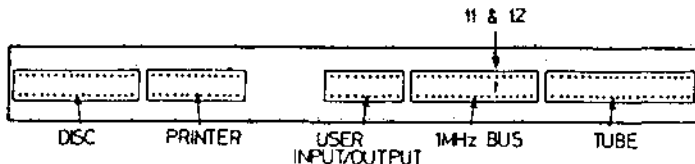
NOTE: Although two wires are connected to the 1MHz bus only one, pin 12, is actually used by the B2P.

Apply power to the BBC.

If the BBC does not power up as normal, SWITCH OFF and check that the TUBE connector is pushed home completely, and that the 2-pin plug is in the correct way and on pins 11 and 12 of the 1MHz bus. If success eludes you contact PMS.

Your B2P Second Processor is now installed.

The B2P can be removed at any time, with the POWER OFF. If the B2P is not active then it may be left permanently plugged into the BBC without interfering with any other software or hardware.

CHAPTER THREE

STARTING UP THE B2P,

3.1 Activating the Second Processor

To activate the B2P type:

*B2PON <RETURN>

The screen will clear and the title string will appear:

PMS B2P-6502 Second Processor

There will be *a* short pause while the current language (normally BASIC) is copied over the TUBE, and then the prompt for that language will appear.

The Second Processor is now active and will remain active until switched off.

If the B2P does not start up, ie the above message does not appear, check that the BOS ROM is present, and that the B2P is correctly fitted. SWITCH OFF the BBC and try again. DO NOT try starting up the Second Processor without first switching off the BBC as the B2P must be reset by a switch on. (see APPENDIX III "Fitting a RESET button")

3.2 Switching Off the Second Processor.

Unlike the ACORN 6502 Second Processor the B2P can be switched off by software. this is achieved by the command:

*B2POFF <RETURN>

The effect of this command is to make the B2P go dormant. The BBC now behaves *as* if the Second Processor is not attached. If the B2P has been switched off by a *B2POFF command, it can be re-activated by issuing the command *B2PON. Data present in the B2P is maintained during the time that it is dormant.

3.3 Effects of BREAK

When the B2P is active, the BREAK key has the following effects:

BREAK alone- B2P remains in the current language.

CTRL-BREAK - causes the default language to be copied across the TUBE. If the current language is not the default language, then the language present in the B2P will be overwritten by the default language. The default language is the one which appears at switch on.

B-BREAK - This has the effect of returning you to the I/O Processor, while making the Second Processor go dormant. This allows advanced programmers to develop software on both sides of the TUBE simultaneously. Either BREAK or CTRL-BREAK will return you to the Second Processor as above. Data in the Second Processor is retained while it is dormant, not withstanding the effects of the BREAK.

NOTE: When a B-BREAK is executed, memory from &400 to &7FF is overwritten, in BOTH processors. Therefore in BASIC, integer variables will NOT be maintained over B-BREAK.

3.4 *HELP and Processor Status.

Entering:

*HELP \<RETURN\>

will display the current status of the Second Processor. The status will change as a result of the commands *B2PON or *B2POFF being executed, or a B-BREAK being performed.

CHAPTER FOUR

SYSTEM MEMORY

4.1 The I/O Processor Memory Map.

Figure 2 shows the memory map of the I/O Processor (BBC) when using the B2P.

FIGURE 2

OS ROM	&FFFF
	&FF00
Memory mapped I/O	&FC00
OS ROM	&C000
Selectable language and filing system ROMs	&8000
RAM used for screen display	
	Movable boundary
	OSHWM + &5FF
Reserved for OS use and utilities eg DFS	0000
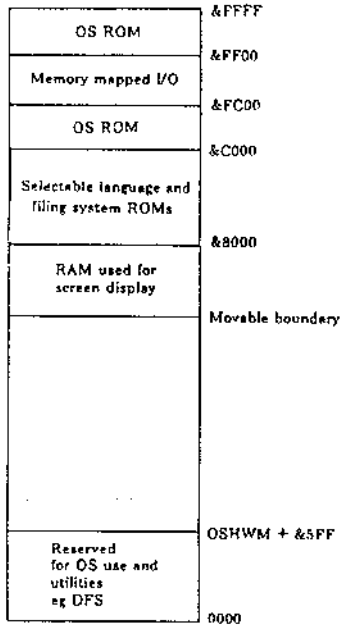
NOTES:

1. OSHWM - the Operating System High Water Mark is raised by &600 to allow characters &20 to &FF to be redefined. This is equivalent to issuing a *FX20,6 (refer to chapter 42 BBC Microcomputer System User Guide). When the Second Processor is activated the character set is exploded automatically.

2. The memory between OSHWM and bottom of display RAM is available for user applications.

3. The TBUFF printer buffer utility uses the memory between OSHWM and the bottom of display RAM, if activated.

4.2 The B2P Second Processor memory map.

Figure 3 shows the memory map of the B2P when activated.

FIGURE 3


```
                              &FFFF
        Second Processor OS

                              &F800
        User memory
        (14K)
                              &C000
        BASIC language
        equivalent to
        BASIC ROM in
        BBC Microcomputer
HIMEM   BASIC stack           &8000
        and
        dynamic
        variable
        storage
LOMEM   User                  Movable boundary
        memory
PAGE                          &0800
        Language
        variables
                              &0400
        System
        variables and
        stack
                              0000
```
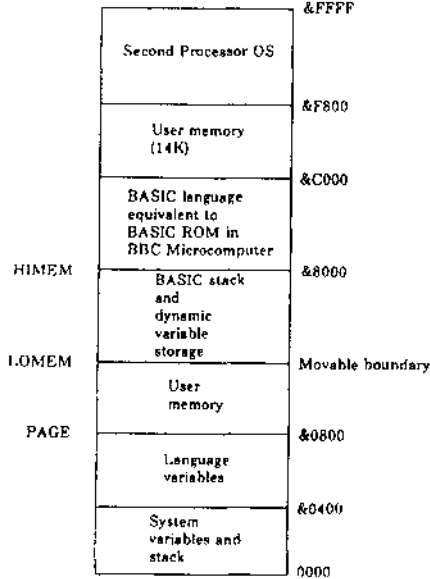
NOTES:

1. All the memory in the B2P is RAM.
2. The diagram above shows the memory map when using a standard language such as BASIC or VIEW. Since no memory is used by the screen display or filing systems, there is a considerable increase in the amount of memory available for the user.
3. There is a 14K area of memory between &C000 and &F800 which is normally unused. This area of memory may be used for machine code routines or for BASIC variable storage (see note in chapter 5).

4.3 The Effects of *B2PON

When the Second Processor is activated by the command *B2PON. the following events take place:

1. The I/O TUBE software is copied from the BOS ROM and is installed at &400-&7FF in the I/O Processor.
2. The TUBE OS is copied across to &F800-&FFFF in the Second Processor.
3. The character set is then exploded to allow all 224 printable characters to be redefined
4. The current language in the BBC is then copied across the TUBE to its TUBE relocation address in the B2P. For non-'Hi' versions of languages this address is &8000.

BASIC ON THE SECOND PROCESSOR.

5.1 Entering BASIC

If BASIC is the default language on the BBC, it will be automatically copied over the TUBE when the B2P is activated by *B2PON,or after a CTRL-BREAK with the B2P active.

If you have another language in the B2P and you wish to enter BASIC, type *BASIC. Entering BASIC this way will overwrite the previous language in the B2P.

5.2 Using Standard BASIC.

Using BASIC on the B2P is exactly the same as using it on the standard BBC, except that the amount of free memory is the same no matter which screen MODE is in use.

PAGE is fixed at &800 and HIMEM is always at &8000 (for standard BASIC) regardless of the MODE or filing system.
When in standard BASIC, a 14K area of memory, between &C000 and &F7FF, is unused (see figure 3). This may be used for variable storage -by setting LOMEM to &C000, and HIMEM to &F800. BASIC variables are then stored in this area, rather than taking up BASIC program space.

5.3 Using Hi-BASIC

Hi-BASIC is a version of BASIC 2 which has been relocated to make use of the 'spare' 14K of memory normally free when in a standard language. Hi-BASIC raises HIMEM to &8800 giving 44K free for BASIC programs and data. The syntax and commands of Hi-BASIC are exactly the same as BASIC in the standard BBC. (see APPENDIX IV "Entering HiBASIC").

NOTE: Hi-BASIC is available directly from PMS, on ROM or disk.

5.4 Notes on running BASIC programs on the 82P.

Programs developed on the standard BBC will RUN unaltered in the B2P, as long *as* they comply with the conditions set out in the following section.

Programs which directly access screen RAM, system variables or undocumented Operating System calls will NOT function correctly when run on the B2P. Direct access to hardware, eg 6522, 8271, 6845, A/D, RS423 chips or devices on the User Port or 1MHz bus is not possible by programs in the B2P and will cause unpredictable results if tried!

NOTE: Official OS routines are provided to access all these devices and must be used to maintain Second Processor compatibility.

If BASIC programs contain machine code routines, care must be taken to ensure that the code *is* located in areas of memory which are not used on the Second Processor. For example, on a BBC, a common place to put code is &A00-&AFF, (cassette input buffer). This area of memory on the Second Processor is now used for BASIC program storage, and M/C routines placed there will overwrite programs present in this memory.

LANGUAGES

6.1 Languages Resident in ROM.
 ROM based languages, other than BASIC, can be used on the Second
Processor by typing in the appropriate star command to enter the
language. This will cause the language to be copied across the TUBE
into the B2P. Once in the B2P, they will then operate as on a
standard BBC but again with the advantage of increased memory.

eg to enter VIEW type:

> **♦WORD (RETURN)**

6.2 Using Languages Prom Disk.
When using the B2P, it is possible to load SOME ROM-based languages
directly from disk. The software should first be transferred out of
the ROM onto the disk.

With the language ROM present in the I/O Processor, activate the B2P,
enter the appropriate language as in 6.1 and then, from the language,
type

> **•SAVE Lang 8000+4000 <RETURN>**

for a normal language

or

> ***SAVE lang B800+4000 <RETURN>**

for most 'Hi' languages.

This will save a file called 'lang' on the disk. The ROM may then be
removed from the BBC, stored away, and whenever the language is
required, typing:

> ***lang <RETURN>**

will load the language into the B2P and enter it.

This allows you to have a library of languages on disk, and call them
up whenever they are needed rather than inserting and removing ROMs.

NOTES:

1. Care must be taken that you do not infringe copyright by this
procedure!!!
2. Recently certain software houses have taken to protecting their
language ROMs with the effect that the ROM must be present in the I/O
Processor for the software to run in any Second Processor. If this *is*
the case, the technique described above for storing languages on disk
will NOT work.
3. WORDWISE PLUS is not completely compatible with the B2P, and BO
standard WORDWISE PLUS can not be run. However a special 'Hi' version
giving 42K of text storage with 80 column preview, *is* available
directly from PMS, on disk.

6.3 Games and Other Commercial Software.

Most games are NOT designed to work with *a* Second Processor, and will
therefore not work when the B2P is active. They can however be used
when the B2P is not active.
Other commercial software will work on the Second Processor as long
as it is written to be Second Processor compatible. Most business and
educational programs are Second Processor compatible and should
therefore work on the B2P. Because of the vast range of software
available for the BBC, PMS may not be able to comment on the
compatibility of an individual piece of software - contact the
software house supplying the software for information. Practically
any program that works on the Acorn 6502 Second Processor will
operate correctly on the B2P apart from a very few which are
dependant on TUBE hardware.

SECOND PROCESSOR ADDRESSING

7.1 Extended Addressing for Filing Systems.

When a Second Processor is connected, the current filing system has to distinguish between the I/O Processor's memory and the Second Processor's memory when loading and saving files. It does this by what is called 32 bit addressing. This defines the Second Processor's memory as running from &00000000 to &0000FFFF and the I/O Processor's memory as running from &FFFF0000 to &FFFFFFFF.

Therefore, as leading zeros can be ignored, location &C000 is in the Second Processor and location &FFFFC000 is in the I/O Processor.

For example, suppose you had a machine code program running in the Second Processor, located from 8.1000 to 82000 and this program draws a picture on the screen in MODE 2. You could **save** the program and picture with the following two commands:

*SAVE program 1000 2000

and

*SAVE Picture FFFF3000 FFFF8000

since screen memory is in the I/O Processor.

When saving and loading BASIC programs, the above distinction will not apply. BASIC always loads programs to PAGE in the currently active processor. A program SAVEd from a standard BBC with PAGE at &1900 will automatically LOAD to PAGE at &800 in the Second Processor. This means that BASIC programs will be totally compatible with both processors.

NOTE The filing system present in the BBC must support 32 bit addressing for it to function successfully with the 82P.

7.2 Extended Addressing from BASIC and Machine Code.

BASIC does NOT take account of extended addressing. Therefore ?,! and $ cannot address I/O Processor memory' when running in the Second Processor, and vice versa.

For example, ?&FFFF2000=25 entered when the Second Processor is active will put the value 25 into location &2000 in the SECOND PROCESSOR, not the I/O processor.

However, I/O Processor memory can still be accessed by using OSWORD calls 5 and 6. as detailed in the section on OSWORD in the BBC User Guide and Advanced User Guide.

USING THE B2P FROM MACHINE CODE.

8.1 Guidelines for the machine code programmer.

Official OSBYTEs and OSWORD calls MUST be used in machine code programs designed to operate in the Second Processor. All operating system calls are fully supported with the same call addresses as in the standard BBC. For example, the entry point for OSBYTE is at &FFF4 on both the Second Processor and standard BBC.

The following points should be observed when programming in machine code in the Second Processor.
1. OSBYTE calls between 0 to 127 return a value only in the X register. High numbered OSBYTE calls return values in the X, Y registers and the Carry flag. Non-standard use of low numbered OSBYTEs will cause hardship.
2. OS calls within an event are not allowed from the Second Processor. **eg** printing of characters from within events is not possible.
3. OSBYTE call &82 always returns X=Y=0.
4. OSBYTE call &83 always returns X=0 and Y=8.
5. OSBYTE call &84 always gives the execution address of the currently running program. eg &8000 for standard BASIC and &B800 for HiBASIC.

8.2 Advanced Use of TUBE

The following section is intended for programmers who have a working knowledge of the routines used to directly access the TUBE byte at &FEE5.

TUBE transfer routines initiated by JSRs to &406 are fully implemented, and bytes can be transferred via the TUBE byte at &FEE5 in the I/O Processor. Access to the TUBE is exactly the same as using an Acorn 6502 Second Processor on a BBC. However, the following three points should be borne in mind:
1. The B2P runs at 2MHz, not 3MHz and routines should include sufficient delays to take account of this - 32µS between accesses to &FEE5.
2. Since the TUBE ULA does not exist in the B2P, attempts to directly access the ULA will not work and official calls should always be used to communicate between processors.

The BOS ROM.

The BOS ROM contains the B2P Operating System commands (*B2PON and *B2POFF), and a set of TUBE Utilities which are all accessed through STAR commands. All addresses required by the following commands should be in HEX.

9.1 *HELP from the B0S ROM.

Help information is displayed when the command:

*HELP BOS <RETURN>

This lists all the available commands, with syntax, and displays the current status of the Second Processor and Printer Buffer.

9.2 *TDIS (<addr>)

TDIS is a 65CO2 disassembler which disassembles machine code programs residing in the Second Processor memory. Disassembly commences at the specified address, or at &0000 if none is specified. TDIS uses the following keys to control the disassembly.

F - Continuous disassembly.
S - The disassembler will wait after each instruction until a key is pressed.
D - Suppresses MNEMONIC production in the listing.
M - Cancells the effect of the above key.
CTRL B - Sends the disassembly listing to the printer. CTRL C - Sends the disassembly listing to the screen only. SPACE BAR - Causes the disassembler to follow branches in the program.
R - Forces a return from a subroutine jumped to by pressing the SPACE BAR.
ESCAPE - Stops disassembly, and prompts for a new start address. Pressing ESCAPE again will terminate the disassembly completely.

9.3 *TZAP (<addr>)

Enters a powerful, interactive, Second Processor memory editor. The editor displays the memory contents around the address specified after the command. if no address is specified, the editor defaults to &800. TZAP uses the following keys:

CURSOR KEYS - to move up, down, left and right.
CURSOR KEY + SHIFT - to move one screen of up or down, or move to extremes of line.
TAB - Selects data entry mode - ASCII, HEX, Decimal or Binary.
ESCAPE - Exits editor.

NOTE: Since the Second Processor is running while the memory is being edited, it is inadvisable to alter memory locations above &F800 or the stack in page one. Also, because of the way the Second Processor handles the ESCAPE key, the screen display may change when the ESCAPE key is pressed. If this causes hardship, the COPY *may* be used to exit TZAP.

9.4 *TSEARCH <search string> (<addr>)

Searches through the Second Processor's memory for every occurrence of the search string. When a match is found, TZAP is entered with the cursor pointing to the first character in the string. To move to the next occurence, press the COPY key. While in the editor, the ESCAPE key can be used to terminate the search.

NOTE: In TSEARCH the search string may contain characters not directly accessible from the keyboard. These characters are included in the search string using the same format as used when defining function keys, (see APPENIX V "List of characters for use with TSEARCH").

9.5 *TSHIFT <src> <dest> <ext>

Moves a block of Second Processor memory of length (ext), from the source address (src), to the destination address (dest). Both addresses refer to Second Processor memory.

9.6 *IOWRITE <src> <deg> <ext>

Copies a block of Second Processor memory of length (ext), from the source address (src), to the destination address (dest). The destination address is in I/O Processor memory (ie movement across the TUBE).

9.7 *IOREAD <src> <dest> <ext>

Copies a block of I/O Processor memory of length (ext), from the source address (src), to the destination address (dest). The destination address is in Second Processor memory (ie movement across the TUBE).

9.8 *TBUFF <ON/OFF>

Configures the free memory in the I/O Processor between OSHWM and &2F00 as a printer buffer. The size of the buffer is about 4K depending on filing system present.

NOTE: Pressing BREAK switches off the printer buffer.

The current status of the printer buffer is displayed on *HELP.

9.9 *PURGE <ON/OFF>

•PURGE OFF prevents the printer buffer being emptied when ESCAPE, *FX15 or *FX21.3 is issued.
*PURGE ON reverts to normal buffer purging. This is the default setting.

9.10 *GOIO <addr>

Equivalent to the CALL instruction in BASIC, but allows machine code programs in the I/O Processor to be CALLed from the Second Processor.

9.11 *GO (<addr>)

Equivalent to the CALL instruction in BASIC, but allows machine code programs in the Second Processor to be CALLed. This command is useful for executing machine code programs from languages which do not have the equivalent of a CALL instruction.

If no address is specified, the execution address of the current language is called.

9.12 &F800 CODE

A machine code routine located at &F800 allows users to enter OS commands without a language being present. This can be useful for loading machine code programs which overwrite the memory normally occupied by the current language.

To enter this routine either CALL &F800 from BASIC or *00 F800. A * prompt will appear, at which OS commands can be entered.

To return to the previous language, provided it has not been overwritten, type :

*G0 <RETURN>

or

*GO 8000 <RETURN>

or whatever the start address of the language is.

Alternatively, a language can be entered by the normal precedure. eg *BASIC <RETURN>.

APPENDIX I

Transferring the BOS into Sideways RAM.

The BOS ROM is unprotected, and may be transferred onto disk for loading into sideways RAM. Insert the BOS ROM into the BBC and use the standard procedure to copy the contents of a ROM to disk.

Alternatively, with the BOS ROM and Second Processor installed and active, type:

        *IOREAD 8000 2000 4000 <RETURN>

        ♦SAVE"BOSROM" 2000+4000 FFFF8000 FFFF8000 <RETURN>

The BOS ROM is protected by COPYRIGHT ACT, but PMS will permit a SINGLE disk copy of the BOS ROM to be made by an individual user.

APPENDIX II

NPFD Signal from BBC board.

If the 1MHz bus connector is not available, the NPFD signal (pin 12 on the 1MHz bus), may be picked up on the main BBC circuit board. The most suitable point at which to pick up the NPFD signal is on the + side (left side) of diode D15. This component is located immediately behind the 1MHz bus connector. There are 2 small diodes in parallel, and D15 is the one closest to the rear of the circuit board.
If you require to do this modification, PMS can supply (at no charge) an adaptor kit. This uses a miniture clip for D15, thus no soldering is required. Full instructions are supplied with the kit.

APPENDIX III

Fitting a RESET button.

Normally the B2P processor is RESET at power on. In exceptional circumstances, a user may require to be able to RESET the 6502 processor in the B2P manually. The ability to RESET the B2P will only be of value to advanced machine code programmers who are developing software in the I/O Processor, which directly accesses the TUBE.

NOTE: When the B2P is RESET manually the data in the RAM of the B2P will be lost.

Details of how to fit a RESET button can be obtained from PMS.

APPENDIX IV

Entering Hi-BASIC

Starting up Hi-BASIC in the B2P depends upon the priority of the ROM socket that it is inserted into.

Hi-BASIC must be in a ROM socket, with a priority lower than another language - to the left of the other language.

In this case, the default language will be copied into the B2P when it is activated. To use Hi-BASIC, the operating system must be told to enter Hi-BASIC. Since both Hi-BASIC and standard BASIC are accessed by a *BASIC command, the *FX command to enter a language MUST be used. To do this, determine (by trial and error or otherwise) the ROM number (0-15) of Hi-BASIC. Then type in:

      *FX 142,n <RETURN>

where n is the ROM number of Hi-BASIC.

In order to check that you are in fact in Hi-BASIC, PRINT the value of HIMEM. This should be &B800 in Hi-BASIC.

NOTE: Disk based Hi-BASIC can be directly *RUN in the B2P by entering:

      ♦HIBASIC <RETURN>

with the disk containing the file "HIBASIC" in the current drive.

APPENDIX V

List of characters for use in TSEARCH.

If you realy want to, unprintable characters may be searched for using TSEARCH by substituting the appropriate OS control sequence from the table below.
For example to search for the string of bytes -

&A9. &41. &4C, &EE, &FF enter:

                    *TSEARCH "|!)AL|!n|!|?"

This is of course obvious???? if not stick to searching for words!!!
!

```
&0-¦@      &1-¦A      &2-¦B      &3-¦C      &4-¦D      &5-¦E      &6-¦F      &7-¦G
&8-¦H      &9-¦I      &A-¦J      &B-¦K      &C-¦L      &D-¦M      &E-¦N      &F-¦O
&10-¦P     &11-¦Q     &12-¦R     &13-¦S     &14-¦T     &15-¦U     &16-¦V     &17-¦W
&18-¦X     &19-¦Y     &1A-¦Z     &1B-¦[     &1C-¦\     &1D-¦]     &1E-¦^     &1F-¦_
&20-       &21-!      &22-"      &23-£      &24-$      &25-%      &26-&      &27-'
&28-(      &29-)      &2A-*      &2B-+      &2C-,      &2D--      &2E-.      &2F-/
&30-0      &31-1      &32-2      &33-3      &34-4      &35-5      &36-6      &37-7
&38-8      &39-9      &3A-:      &3B-;      &3C-<      &3D-=      &3E->      &3F-?
&40-@      &41-A      &42-B      &43-C      &44-D      &45-E      &46-F      &47-G
&48-H      &49-I      &4A-J      &4B-K      &4C-L      &4D-M      &4E-N      &4F-O
&50-P      &51-Q      &52-R      &53-S      &54-T      &55-U      &56-V      &57-W
&58-X      &59-Y      &5A-Z      &5B-[      &5C-\      &5D-]      &5E-^      &5F-_
&60-`      &61-a      &62-b      &63-c      &64-d      &65-e      &66-f      &67-g
&68-h      &69-i      &6A-j      &6B-k      &6C-l      &6D-m      &6E-n      &6F-o
&70-p      &71-q      &72-r      &73-s      &74-t      &75-u      &76-v      &77-w
&78-x      &79-y      &7A-z      &7B-{      &7C-¦¦     &7D-}      &7E-¯      &7F-¦?
&80-¦!¦@   &81-¦¦A    &82-¦!¦B   &83-¦!¦C   &84-¦!¦D   &85-¦!¦E   &86-¦!¦F   &87-¦!¦G
&88-¦!¦H   &89-¦!¦I   &8A-¦!¦J   &8B-¦!¦K   &8C-¦!¦L   &8D-¦!¦M   &8E-¦!¦N   &8F-¦!¦O
&90-¦!¦P   &91-¦!¦Q   &92-¦!¦R   &93-¦!¦S   &94-¦!¦T   &95-¦!¦U   &96-¦!¦V   &97-¦!¦W
&98-¦!¦X   &99-¦!¦Y   &9A-¦!¦Z   &9B-¦!¦[   &9C-¦!¦\   &9D-¦!¦]   &9E-¦!¦^   &9F-¦!¦_
&A0-¦!     &A1-¦!!    &A2-¦!"    &A3-¦!£    &A4-¦!$    &A5-¦!%    &A6-¦!&    &A7-¦!'
&A8-¦!(    &A9-¦!)    &AA-¦!*    &AB-¦!+    &AC-¦!,    &AD-¦!-    &AE-¦!.    &AF-¦!/
&B0-¦!0    &B1-¦!1    &B2-¦!2    &B3-¦!3    &B4-¦!4    &B5-¦!5    &B6-¦!6    &B7-¦!7
&B8-¦!8    &B9-¦!9    &BA-¦!:    &BB-¦!;    &BC-¦!<    &BD-¦!=    &BE-¦!>    &BF-¦!?
&C0-¦!@    &C1-¦!A    &C2-¦!B    &C3-¦!C    &C4-¦!D    &C5-¦!E    &C6-¦!F    &C7-¦!G
&C8-¦!H    &C9-¦!I    &CA-¦!J    &CB-¦!K    &CC-¦!L    &CD-¦!M    &CE-¦!N    &CF-¦!O
&D0-¦!P    &D1-¦!Q    &D2-¦!R    &D3-¦!S    &D4-¦!T    &D5-¦!U    &D6-¦!V    &D7-¦!W
&D8-¦!X    &D9-¦!Y    &DA-¦!Z    &DB-¦!{    &DC-¦!\    &DD-¦!}    &DE-¦!^    &DF-¦!_
&E0-¦!`    &E1-¦!a    &E2-¦!b    &E3-¦!c    &E4-¦!d    &E5-¦!e    &E6-¦!f    &E7-¦!g
&E8-¦!h    &E9-¦!i    &EA-¦!j    &EB-¦!k    &EC-¦!l    &ED-¦!m    &EE-¦!n    &EF-¦!o
&F0-¦!p    &F1-¦!q    &F2-¦!r    &F3-¦!s    &F4-¦!t    &F5-¦!u    &F6-¦!v    &F7-¦!w
&F8-¦!x    &F9-¦!y    &FA-¦!z    &FB-¦!{    &FC-¦!¦¦   &FD-¦!}    &FE-¦!¯    &FF-¦!¦?
```

RESERVED FOR FUTURE EXPANSION

LAST WORDS

EXTREMELY TECHNICAL SECTION.

This is the bit where we attempt to tell you how our B2P works. The B2P does not have a fancy, expensive ULA chip, but instead simulates the TUBE hardware protocols through some discrete components, and some very neat software. The effect of this is that at language level the TUBE behaves exactly like its ACORN counterpart. However machine code programmers will discover that TUBE software will run slightly slower for two reasons. Firstly the 6502A (or 65C02) processor in the B2P runs at 2MHz *as* opposed to 3MHz on the ACORN, and secondly, the FIFOs on the TUBE ULA are simulated by the neat software we mentioned previously.

The TUBE byte is located at its official location of &FEE5, but the OSWRCH queue etc. are located in page &FD. This is the reasoning behind the extra connection to the 1MHz bus. All of page &FD is used, so other hardware which uses page &FD will be incompatible with the B2P. After all, page &FD IS set aside for a 64K RAM expansion, and the B2P has got 64K of RAM on board!

Oh dear, I seem to have run out of coffee, so I will stop here and leave the explanations on how we start up a processor without ROM and the concept of DRAM refresh by ----