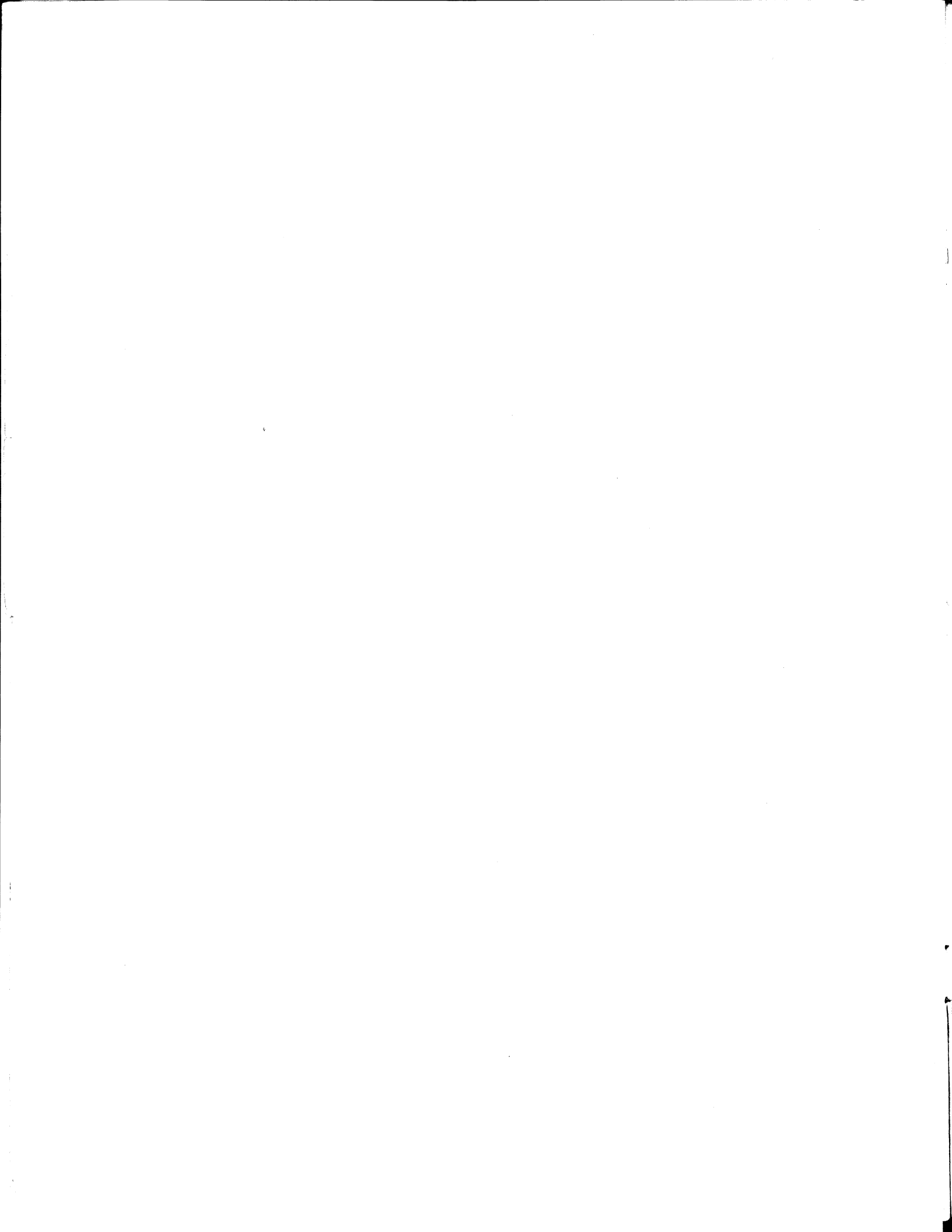


**MICROBUFFER II USERS MANUAL**

May 1, 1982

Version 2.1

Copyright (c) 1982  
All Rights Reserved  
Practical Peripherals, Inc.  
31245 La Baya Drive  
Westlake Village, CA 91362  
(213) 991-8200



## INTRODUCTION

The **Microbuffer II** is an intelligent Centronics-compatible parallel printer interface for the Apple II and Apple II Plus computers. The **Microbuffer II** has up to 32K bytes (16K standard) of on-board memory for data buffering, and provides useful text control functions. For users with certain 'graphics' type printers, the **Microbuffer II** includes an extensive set of advanced Hi-Resolution graphics dump routines.

Data buffering increases data processing efficiency by freeing the Apple and the operator from the wait normally experienced while printing. The **Microbuffer II** will allow the Apple to print and process simultaneously. The **Microbuffer II** will accept data as fast as the Apple can send it (up to the buffer size) and return control of the computer to the user while it handles the printing. Additional data may be sent to the **Microbuffer II** without waiting for previous jobs to be completed.

Graphics capable printers supported include: Epson MX70 and MX-80, MX-80 F/T (with Graphtrax), Epson MX100, C.Itoh Pro-Writer, NEC 8023, Anadex 9000/9001/9500/9501, and IDS Paper Tigers. Each graphics printer requires a specific model of the **Microbuffer II** to enable use of the graphics routines. The data buffering and text control functions are common to all models so one **Microbuffer II** may be used on any parallel printer when graphics are not going to be used.

The **Microbuffer II** is designed to be easy to install and easy to use, please read all instructions carefully before starting. If you are unfamiliar with Apple hardware or software ask your dealer for assistance. The **Microbuffer II** was developed to enhance the computing power of your Apple by making data output less time consuming and effort-free.

## INSTALLATION INSTRUCTIONS

The **Microbuffer II** is slot independent and may be used in any Apple II or Apple II+ slot except zero. If using Pascal or CP/M always use slot one.

1. **IMPORTANT: TURN OFF THE POWER**, failing to do so may cause serious damage to the computer and the **Microbuffer II**. Do not touch the gold card edge fingers.
2. Install the printer cable (supplied) on the **Microbuffer II** by plugging it on to the double row of pins so that the cable extends from the connector toward the rear of the Apple when the board is in place.
3. Consult the DIP-switch section near the end of the manual and make any desired changes in the default settings.
4. Remove the cover from the Apple by pulling up on the rear edge of it until the fasteners release, then slide the cover towards the rear of the Apple and lift out.
5. Insert the **Microbuffer II** into any of the peripheral slots, except zero, at the rear edge of the Apple circuit board. Slots are numbered from left to right (0 to 7) when looking at the Apple from the keyboard end. Printer interfaces are usually installed in slot #1. Press the **Microbuffer II** into the slot with a rocking motion until it is firmly seated.
6. Route The cable through one of the vertical openings at the rear of the Apple and replace the cover.
7. Plug the cable into the printer, the connector is shaped so that it will fit only one way.
8. Turn the Apple power switch on, install paper in the printer, turn on the printer power, and set the printer on-line.
9. You are now ready to use your new **Microbuffer II**. The **Microbuffer II** is activated from BASIC by typing **PR#n<cr>**, where (n) is number of the slot where the card is installed, and <cr> means press the 'RETURN' key. The printer should print the BASIC prompt and linefeed if everything is OK.
10. The **Microbuffer II** software includes a self-test function. The self-test is activated by holding down the Apple 'CTRL' key while simultaneously pressing the 'I' key, then releasing these two keys and pressing the 'V' key. A diagnostic message will print on the Apple screen and the printer will print a one page test pattern.
11. Consult the help section of this manual if necessary.

## FUNCTIONAL DESCRIPTION

The **Microbuffer II** contains an intelligent controller, control software in ROM and hi-speed RAM for data buffering. The RAM allows the **Microbuffer II** to accept data from the Apple at up to 4,000 characters per second to the limit of memory available. If the amount of printed data is less than the buffer size, the Apple will complete it's dump in a matter of seconds. When the amount of printed data exceeds the buffer size, the **Microbuffer II** will respond to the Apple as a normal printer interface taking one line at a time. This will result in an approximate time savings of nine minutes for a 80 cps printer assuming a thru-put of 60 cps, when using a 32K buffer size. Time savings will vary with line length. For output that consists of very short lines, like assembler listings, the time savings will be greater because printers take longer to print a line feed and carriage return than to print normal characters.

The software in ROM controls all of the functions of the **Microbuffer II** and is different for each graphic printer. The non-graphic functions of the **Microbuffer II** are the same for all versions of the firmware. This will allow the **Microbuffer II** to be used on several printers in all modes except graphic. The Universal **Microbuffer II** will typically be shipped with Epson firmware.

## OPERATION OF THE MICROBUFFER II

The **Microbuffer II** is Apple parallel interface printer card and is activated by typing PR#n<cr>, where 'n' is the slot number that it is in, and <cr> indicates pressing the 'RETURN' key.

The **Microbuffer II** firmware contains an extensive set of advanced text formatting and graphics dump commands. These commands are initiated by sending a control character, followed by one or more letters and/or numbers to **Microbuffer II** from either the Apple keyboard or from within operating programs.

Control characters have ASCII decimal values of zero (0) to thirty-one (31) and are not part of the normal displayable character set. Control characters are formed by pressing a special control key while pressing one of the keys 'A' thru 'Z'. When you are using Apple BASIC and you want to send a control sequence to the **Microbuffer II** from the keyboard the command is <CTRL-I>. To send a <CTRL-I>, you press the 'CTRL' key then press the 'I' key, while you continue to hold the 'CTRL' key, release the 'I' key and finally release the 'CTRL' key. The control sequence gets the 'attention' of the **Microbuffer II** which then waits for the command. The control character is then followed by a specific set of characters that will activate the function desired. The command is complete when the last character in the sequence is entered. All control commands to the **Microbuffer II** are initiated by a <CTRL-I> when in Basic.

When operating in Pascal or CP/M the commands are the same but the control character is <CTRL-Q> (CP/M and Pascal interpret <CTRL-I> as a tab character).

When activating or controlling the **Microbuffer II** from within an operating program, the control characters are the same and are formed by printing the equivalent characters to printer.

Apple BASIC:

<CTRL-I> is equivalent to CHR\$(09)

CP/M & Pascal:

<CTRL-Q> is equivalent to CHR\$(17)

From BASIC:

PR#n<cr> Turns the **Microbuffer II** on, where n is the slot# the **Microbuffer II** is in. This command must be issued before any of the following commands are issued.

PR#0<cr> Turns off the **Microbuffer II**

To issue a command from the keyboard in BASIC, turn on the card using PR#n, then (to set the page length to 60) type <CTRL-I>60P<cr>. to issue the same command from within a program use this statement:

```
5 PR#n
10 PRINT CHR$(9);"60P"
```

When operating from within Apple DOS the command sequence would be:

```
10 PRINT CHR$(4);"PR#n";CHR$(09);"60P"
```

Where CHR\$(4) = <CTRL-D>

## CLEARING THE BUFFER

The buffer may be cleared by pressing the Apple 'RESET, key if switch 2 on the **Microbuffer II** is on/closed, the RESET key will have no effect on the state of the buffer if switch 2 is off/open. From software the Zap command may be used.

**TEXT FORMATTING**

The **Microbuffer II** provides many useful text formatting and printer control features, including, auto skip-over-perforation, set left margin, set line length, dump text screen, basic listing formatter, etc. These functions can be initiated from the keyboard or under program control. Each function is used by issuing a control character followed by the letter or number for the specific function, and terminated by a carriage return.

**TEXT COMMANDS**

Do not include the "<" or ">" in the printer command string. They are listed here only to improve command clarity. When in CP/M or Pascal the control character is <CTRL-Q>.

- <CTRL-I> A X Perform an **Automatic** linefeed with every carriage return. The **Microbuffer II** normally defaults to this.
- <CTRL-I> B Enable the printer **Bell**. If your printer has a bell, printing a <CTRL-G> (CHR\$(7)) will ring the bell.
- <CTRL-I> C X Cancel the printer bell. This will keep <CTRL-G> in program listings from ringing the printer bell. The **Microbuffer II** normally defaults to this.
- <CTRL-I>(n)C Set the **Control** character to any value (n= 0 to 255). This command allows the interface control character to become a printable character which can be imbedded in an APPLEWRITER file or become a value which allows unusual data to be passed through the card without being falsely interpreted as a command character.
- <CTRL-I> D X Set all text formatting parameters to the **Default** values. This command is used to clear the **Maintain** mode and has the effect of issuing a PR#n in the normal mode. The default parameters are; page length=0, line length=dipswitch #1 setting, left margin=0, auto linefeed at carriage return, disabled printer bell, and disable bit 8.
- <CTRL-I> E X Send an **Escape** character to the printer. If the Apple is equipped with Integer Basic, the escape character cannot be sent because of the lack of the CHR\$ function.

- <CTRL-I> F X Format the BASIC listing. This command formats a Basic listing by taking multiple statement BASIC program lines and breaking them up at the colons and placing each statement on a new line indented. This makes program logic clearer and listings easier to read. When activated, typing LIST will produce a formatted listing. This mode should be turned off when not listing a program because any output will be formatted at included colons. This mode is turned off by issuing a <CTRL-I><cr> or issuing any other interface command.
- <CTRL-I> H Activate the High order bit. This will enable the Apple to send the 8th data bit to the printer allowing the special character sets on some printers to be used.
- <CTRL-I> I Turns on the Apple screen. This will allow simultaneous output to the screen and printer. <CTRL-I> O has the same effect to be compatible with other cards.
- <CTRL-I> K Kill auto linefeed sent with carriage return. The automatic linefeed with each carriage will be turned off and linefeed control will then be under program control.
- <CTRL-I> L X Reset the Left margin.
- <CTRL-I>(n)L Set the Left margin 'n' columns from the left edge of the paper. The margin can be set from 0 to 255.
- <CTRL-I> M Enable Maintain mode. When in this mode, all print formatting parameters are maintained when leaving and reentering the card. This allows the parameters to be entered only once for as long as the Apple is on and maintains them when entering Pascal, CP/M, basic etc.. This is especially useful when running programs which do not allow issuing of interface commands. <CTRL-I> D resets this mode.
- APLE WORKS  
<CTRL-I> N X Set the line length to 0. This prevents the card from issuing carriage returns if the line length is exceeded.
- <CTRL-I>(n)N Set the printer line length to 'n' characters from the left margin with No video. The line length can be set from 0 to 255.
- <CTRL-I> P X Reset the page length to 0. This eliminates the auto skip-over-perforation function.



- <CTRL-I> (n) P Set the Page length to 'n' lines. The **Microbuffer II** will print six line feeds and reset its line counter after 'n' lines have been printed. This allows auto skip-over-perforation. Standard continuous form paper has 66 lines per page so <CTRL-I> 60P <cr> will allow automatic skip-over-perforation with a three line margin at the top and bottom of the paper. If 'n' is set to 0 this function is disabled and printing will be continuous. Note: Issuing this command resets the line counter and should be issued after setting the paper to the top of form. The page length can be set from 0 to 127.
- <CTRL-I> S *NOK* Dump the present 40 column **Screen** to the printer. The printout will occur at the present left margin. This command will not dump an 80 column screen. Eighty-column screens do not use the Apple memory mapped video and therefore cannot be accessed.
- <CTRL-I> (n) S Dump the present 40 column **Screen** to the printer starting at line #(n) and stopping at line #23. This command is particularly useful for dumping a portion of the screen, such as a Visicalc window.
- <CTRL-I> T Set **Transparent mode**. This command turns off all interface text formatting and command recognition. This is useful when a program such as a graphics screen dump or word processor sends characters through the interface which might be falsely interpreted as interface commands. This command is also useful when a program has it's own text formatter and the insertion of carriage returns, etc.. are undesirable. Reinitializing the card (PR#n) will clear this mode. **Caution.** If the **Maintain mode** is also set, the only way to reset the **Transparent mode** (since the interface cannot recognize commands) is to clear the high-order bit at location \$4F8+slot. This can be done from basic by typing POKE 1272+slot,PEEK(1272+slot)-128.
- <CTRL-I> V **Verify the operation of the Microbuffer II.** This command checks the on board firmware and indicates the result at the top of the screen. It then generates a one page test pattern to verify most internal components of the card and sends this to the printer.
- <CTRL-I> X *X* Deactivate output of bit-8. This is the opposite of <CTRL-I> H.

- <CTRL-I> This changes the control command character to  
 <CTRL-Q> 'Q'. Any control character may be used in place of  
 'Q'. The control character must not be one <CTRL-  
 Q> <CTRL-I> will change the control command  
 character back to 'I'. The '^' (shift-'N')  
 character is also allowed as a control character.
- <CTRL-I> Z Zap the buffer memory. This command will stop the  
 printing and eliminate the data previously sent to  
 the buffer. Caution. If this command is used  
 during a graphics screen dump, the printer  
 power may have to be toggled off to return the  
 printer to the default condition on since the  
 interface may still be in a graphics mode. Note:  
 If switch 2 is on/closed pressing the RESET key on  
 the Apple will clear the buffer, if switch is  
 open/off pressing the RESET key will not effect  
 the state of the buffer.

## GRAPHICS ROUTINES

The **Microbuffer II** provides for printers capable of Hi-Resolution dot graphics an advanced set of on-board graphics dump routines. All graphics commands are initiated by sending a <CTRL-I> (or <CTRL-Q> for CP/M or Pascal) followed by a 'G' and from none to all of the additional graphics sub-commands and a carriage routine. Graphics screen dumps can be initiated from the keyboard or within a program. On some smaller printers, the Double command can only be used when the image is rotated due to the insufficient number of dots available. Some printers require the installation of special "Graphics Chips" in order to do high resolution graphics. The Epson MX-80 and Okidata 82A are examples.

The following are some examples of the uses of the graphics commands; <CTRL-I> GLIR <cr> will dump the primary HI-RES screen at the Left margin, Inverted, and Rotated 90 degrees.

Two or more pictures can be joined to form a larger picture or a continuous dump, this is also known as a CHART RECORDER MODE. This is made possible by using a space instead of a carriage return at the end of a graphics command sequence. The second picture is specified by simply typing the sub-commands and not re-issuing the <CTRL-I> G. Example; <CTRL-I> GIL<sp>SIL<cr>. This sequence will dump the primary HI-RES page, Inverted, at the Left margin and then immediately following and contiguous with first picture the Secondary HI-RES page, Inverted, also at the Left margin. For longer sequences simply use a space in between groups of sub-commands. If no sub-commands are used, simply print a space for each picture. A carriage return is always required to dump the last picture and terminate the graphics mode.

GRAPHICS COMMANDS

Format of the graphics command:

<CTRL-I> G [sls2...sn]      Initiates the graphic dump. The [sls2...sn] indicates one or more optional sub-commands. Each sub-command may be used only once per dump.

GRAPHICS SUB-COMMANDS

- No option      The primary HI-RES screen will be printed centered on the page, with each white dot on the screen printed as a black dot on the printer.
- G D              The graphics screen will be printed **Double** size.
- E                Print an **Emphasized** image using twice as many dots in the same area resulting in a darker image.
- I **INVERSE**      Invert the graphics image. Each dot on the screen will appear on the paper as a white space against a black background.
- L **Left margin**      The graphics image will be printed at the current **Left** margin.
- M                Dump the **Mixed** HI-RES and text screen. Some programs use the mixed text and graphics mode of the Apple for annotating and displaying various types of data. This command allows that image to be placed on the paper. The four lines of text at the bottom of the screen will always be from the primary page regardless of which HI-RES screen is printed. Note: This command can be used in the non-rotated form only. On some printers the text portion may be appear slightly narrower than the screen. This is due to the difference in width of the screen characters versus the printer characters.
- R **Rotated 90**      Dump the image **Rotated 90** degrees from horizontal.
- S                Dump the **Secondary** page of graphics. The command '2' will perform the same function to provide compatibility with software on other cards.

## PASCAL AND CP/M COMPATIBILITY

The **Microbuffer II** is compatible with both CP/M and Pascal 1.0 and 1.1. Because both Pascal and CP/M use <CTRL-I> as a tab character the **Microbuffer II** will change the control code to a <CTRL-Q> when these systems are used. The only function that isn't available to Pascal is the screen dump. When in the Pascal Filer mode, text formatting commands for the card can be issued from the keyboard by using the Transfer function. This is done by typing T to get into the transfer mode then type CONSOLE:<cr>PRINTER:<cr><CTRL-Q>(command)<CTRL-C>. To list the program APPLE 3:TREE, type T and then the name of the program: APPLE 3:TREE.TEXT<cr> PRINTER:<cr>. In CP/M, to issue commands type <CTRL-P><cr><CTRL-Q>(command)<cr>.

## TECHNICAL INFORMATION

The **Microbuffer II** is configured to 'look' like an Apple serial interface. This was done to insure compatibility with Pascal and CP/M. This is done by the signature bytes located at \$Cs05 and \$Cs07. There are two normal card entry vectors to allow software which works with an Apple serial card or parallel interface. These are located at locations \$Cs02 and \$Cs07.

When CP/M or Pascal output to a typical Apple compatible parallel interface, any on-board firmware is ignored and the data is poked directly to the output port. The acknowledge from the printer causes a page-swap to take place in the parallel I/O firmware PROM. Pascal and CP/M detect the acknowledge by reading a PROM location for a data change during the page-swap.

The hardware on the **Microbuffer II** does not accommodate this page-swapping protocol. This would result in an incompatibility if the card was configured to 'look' like a parallel card. Using a configuration that 'looks' like the serial interface signals CP/M and Pascal to use the on-board firmware.

CP/M and Pascal initiate data transfer by first doing an I/O clear (LDA \$CFFF) to turn off all other 2K expansion ROMs. It then jumps to \$C800 to initialize the card. Each character of data will then be output to the \$C9AA output vector.

For programs that require printer configuration, select the Apple high speed serial interface card.

For machine language programmers, the following information will be useful.

1. Write data to the buffer;  $\$C080 + \$N0$  where 'N' = slot number.
2. Read card status;  $\$C081 + \$N0$  where 'N' = slot number. This is the location that is read to determine if the card can accept another character. Bit D7 is tested for a low indicating another character can be sent. Bit D6 indicates the position of DIPswitch #1. When this bit is set, the DIPswitch is in the closed/on position.
3. ZAP  $\$C082 + \$N0$  where 'N' = slot number. Simply reading this location (when the card status indicates ready) will purge the buffer memory.
4. To access the **Microbuffer II** firmware;

This routine must be used the first time the card is accessed.

```

INIT    LDA $CFFF    Turn off all expansion ROMs.
        LDA $CN00    Turn on slot #1 PROM. N=slot
        LDX $CN      Load "X REGISTER" with the slot number N.
        JSR $C800    Initialize the card.
        RTS          Return.
    
```

This is an example of a character output routine. The character to be sent must be in the accumulator upon entry.

```

OUTPUT  LDX $CN      Load "X REGISTER" with the slot number N
        STA $5B8,X   Save CHAR data to the CHARACTER OUTPUT
                    register
        JSR $C9AA    Go output character to Microbuffer II.
        RTS          Return from subroutine
    
```

5. BASIC COLD START Vector is  $\$CN00$ , where N=slot#.
6. BASIC WARM START Vector is  $\$CN02$  or  $\$CN07$ , where N=slot#.

Use of the above routine requires that the command character be <CTRL-Q>. This entry makes available all routines on the board which are initiated using command character sequences. This includes graphics dumps, text screen dumps, and page length, line length, and margin adjustments etc..

## DIP SWITCH FUNCTIONS

The **Microbuffer II** is factory set for the proper memory size (16 or 32K), line-length=0, 40 column/video on, Keyboard reset enabled. The dip switches may only be changed with the power off.

## DIP SWITCH SETTINGS

SW -1	Open/Off	40 Column default/video on
SW -1	Closed/On	80 Column default/video off
SW -2	Open/Off	Keyboard reset has no effect
SW -2	Closed/On	Keyboard reset clears buffer
SW -3,4		Not used
SW -5	Open/Off	32k Buffer (4 RAM chips)
SW -5	Closed/On	16K Buffer (2 RAM chips)
SW -6	Open/Off	Buffer Enabled
SW -6	Closed/On	Buffer Disabled

## DIAGNOSTICS/SERVICE

**New card doesn't work.**

If the **Microbuffer II** is being installed for the first time and proper operation is not achieved, make sure all of the installation instructions have been followed exactly. Then check the printer. See if it's on-line, has the paper installed correctly, make sure the cable is properly attached, make sure that printer is turned on, and check for a head or paper jam. If all else fails, contact your dealer.

**A problem develops with an working card.**

If a problem develops with a working card (stops printing etc..) check all connections between the card and the printer, verify (if possible) the proper operation of the printer with a different interface card. If the printer is okay, the problem may exist in the memory portion of the card. The **Microbuffer II** can be set in an unbuffered mode to facilitate the temporary use of the card (if a memory failure occurs) until servicing is possible. Consult the dipswitch setting section of this manual. Should this prove ineffective contact your dealer.

If the Apple contains a Mountain Hardware ROMWRITER, the \$CFFF DIPswitch #1 should be set to the off position when using the **Microbuffer II**. The ROMWRITER does not allow it's expansion ROM to be turned off since it ignores the \$CFFF I/O clear command. Because the **Microbuffer II** uses an expansion ROM, this can cause a malfunction.

### **Software problems**

The **Microbuffer II** is designed to function properly with most software. However it is not possible to guarantee that it will function with all software. Some programs may have to be altered to take advantage of the advanced features of the **Microbuffer II**. Consult with your dealer when purchasing software and arrange to test the software with your **Microbuffer II** before you buy.

### **ADDING RAM**

If you have a 16K **Microbuffer II** you may upgrade it to 32K by installing two additional 64K x 1 dynamic RAM chips. These RAM chips may be purchased from Practical Peripherals, Inc. or from your dealer. Any type of 64k RAM chip is acceptable. **Warning** 16K dynamic RAM chips will not work and may damage the card if used. Before attempting installation, turn off the power to the Apple and remove the **Microbuffer II**. Install the RAM chips carefully by lining up the notch end of the RAM with the notch end of sockets U14 and U16 (the notches point toward the DIP switch). Make sure all pins are inserted into the socket and that none are bent. Set DIP switch 5 to the open/off position. Install the **Microbuffer II** in the Apple and run the self-test. If the test pattern is not correct then turn off the Apple and re-check the DIP switch settings and verify that the RAM chips are installed correctly. If the **Microbuffer II** still does not self-test properly, remove the new RAM chips and reset the DIPswitch to use the card until you can consult your dealer.

### **TRADEMARKS**

Microbuffer ][ is a trademark of Practical Peripherals, Inc.  
Apple II and Apple II Plus are trademarks of Apple Computer, Inc.  
CP/M is a trademark of Digital Research, Inc.  
Visicalc is a trademark of Visicorp, Inc.  
ROMWRITER is a trademark of Mountain Hardware, Inc.  
ProWriter is a trademark of C.Itoh, Inc.  
Paper Tiger is a trademark of Integral Data Systems, Inc.  
AppleWriter is a trademark of Apple Computer, Inc.

