

BASIC PRINTER DRIVER

© Interactive Products Corporation, 1977

**PolyMorphic
Systems**

Goleta California, 93017

Using the Poly 88 BASIC Printer Driver
Copyright Interactive Products Corporation, 1977

Section 0 INTRODUCTION TO BPRINT MANUAL

This manual describes the use and operation of a driver for interfacing serial printers to Poly 88 BASIC.

Section 1 provides instruction for the immediate use of BPRINT.

Section 2 contains indepth information regarding BPRINT use and installation.

A program listing for hooking BASIC to HyTypes and DecWriters follows Section 2.

Section 1 SUMMARY FOR IMMEDIATE USE OF BPRINT

Make certain the printer is correctly attached to the printer interface, that it is "on" and not on "LOCAL". It must be ready fo on-line operation.

1.1 Loading BPRINT

BPRINT must be used only with Poly 11K BASIC version A00 and later. Load BASIC (instructions are in your BASIC manual). Stop the cassette tape (BPRINT is on the same side of the tape as BASIC). Hit reset button on the front panel. Type B or P (depending upon the format of your BASIC TAPE), type BPRINT, hit carriage return, and restart the tape. After BPRINT has loaded, BASIC will return to the monitor screen. Be careful if you load BPRINT after BASIC has been running for a while. Loading BPRINT erases any BASIC programs that you might have had loaded in memory. You start "fresh" with BASIC, as if you had just loaded it.

1.2 BASIC Commands for operating the Printer

To:	Type:
Enable the printer	PRINT CHR\$(17)
Disable the printer	PRINT CHR\$(19)
Disable the keyboard	PRINT CHR\$(20)
Enable the keyboard	PRINT CHR\$(18)

Note: Printer must be enabled before keyboard can be enabled.

1.3 Modifying the Baud Rate Used by BPRINT

BPRINT has been set up to work for printers operating at a baud rate of 300 (HyType or DecWriter, for example). This is the most common baud rate used by printers. If your printer uses a different baud rate, a change must be made in the BPRINT program. This can easily be done, either from the front panel mode, or from BASIC. After determining the baud rate used by your printer, refer to the enclosed table. Find the baud rate you need. Look at its equivalent in the Hexadecimal column. Use this number to replace the old baud rate number, 16 hexadecimal.

For instance, if you wish to change the baud rate from 300 baud to 1200 baud, look up 1200 baud. Its equivalent hexadecimal number is 19. Therefore the hexadecimal 19 will replace the old baud rate.

A. Changing the Baud Rate from the Front Panel

Using the example above, we will change the baud rate from 300 to 1200.

- 1) Load BASIC and BPRINT
- 2) Type Control-Z to get to the front panel.
- 3) Type L4A98 followed by a carriage return, to get to memory location 4A98 (whose contents set the baud rate).
- 4) Type the number 19, followed by a space.
- 5) Type SPJ49C0, to get to the start address of BASIC when hooked in with BPRINT.
- 6) Hit a carriage return, and then tupe G. You will now be back in BASIC.

B. Changing the Baud Rate from BASIC.

Again, using the example above, we will change the baud rate from 300 to 1200.

- 1) Load BASIC and BPRINT.
- 2) After a BASIC prompt, type:
POKE 19096,25
- 3) You will now be ready to continue on in BASIC.

The POKE function allows you to directly input a number into memory. As BASIC deals only with decimal numbers, the memory location 4A98 (a hexadecimal number) is converted to its decimal representation, 19096. You must then convert the hex number representing the baud rate to a decimal number (see enclosed table). The number 25 in the example above is the decimal representation of the hexadecimal number 19 used in the previous example. To check this memory location you may use the BASIC function PEEK (see your BASIC manual for an explanation of this function).

1.4 Restarting BASIC

If you should need to restart BASIC (for instance, to return from front panel mode), use the address 49C0, instead of the usual BASIC start address of 2000, if BPRINT is to be used again. When in the front panel mode, restart BASIC by typing SPJ49C0. Then hit a carriage return and type G.

BAUD RATE TABLE

<u>Baud Rate Replacement Numbers</u>		<u>Baud Rate</u>
<u>Decimal</u>	<u>Hexadecimal</u>	
17	11	50
18	12	75
19	13	110
20	14	134.5
21	15	150
22	16	300
23	17	600
24	18	900
25	19	1200
26	1A	1800
27	1B	2400
28	1C	3600
29	1D	4800
30	1E	7200
31	1F	9600

Section 2 THE POLY 88 BASIC PRINTER DRIVER

Section 2 describes the use and operation of a driver for interfacing serial printers to Poly 88 BASIC. This driver is applicable ONLY to Poly 88 BASIC versions A00 and later. The driver program, in conjunction with a Poly 88 printer interface card, and Poly 88 BASIC, provides the ability to print and list on the serial device, and also use the keyboard on that serial device. The driver program is set up for 300 baud devices, such as the HyType, or DecWriter, but may be changed for use with 110 baud devices.

2.1 Installing the Printer Driver

To load the printer driver, BASIC must first be loaded. When BASIC has started, and has printed its version message, the printer driver may be loaded. Hit reset on the front panel, and make sure the printer device is attached to the Poly 88, and is powered on. Now load the driver, by typing B or P, depending on the type of tape you have, followed by BPRINT (the normal tape booting procedure). This tape will auto-start, and you will see the BASIC herald one again. At this time, the driver is initialized, and you may load your BASIC program. Note that after performing its initialization, the printer driver "cold starts" BASIC. This means that any program you had loaded is erased. You will also notice that the number of free bytes available has decreased about 500 bytes, this is the space taken up by the printer driver and the printer buffer.. If the message "Nuts!" is displayed on the screen, rather than BASIC restarting, you have tried to use the driver with an earlier version of BASIC, or BASIC was not loaded.

Using the Poly 88 BASIC Printer Driver
Copyright Interactive Products Corporation, 1977

BASIC must be loaded before loading the printer driver and BASIC version A00 or later must be used because of the "floating patch" - it dynamically hooks itself into BASIC.

2.2 Using the Printer (and its Keyboard, if Any)

The printer driver is "attached" to the character output path in the system; it handles each character that is output to the video screen. The functions of starting and stopping printing, enabling and disabling the keyboard on the serial device, are all done by sending control codes to the video screen through the printer driver. These control codes and their functions are:

<u>Name</u>	<u>Value</u>	<u>Function performed</u>
XON,DC1,ctl-Q	11H/17	Enable the printer
XOFF,DC3,ctl-R	13H/19	Disable the printer
TAPE,DC4,ctl-S	12H/18	Enable the keyboard
TAPE,DC4,ctl-T	14H/20	Disable the keyboard

After an XON is sent to the screen, all characters sent to the screen by BASIC (excluding graphics characters used by PLOT, or characters placed on the screen through POKE to modify memory) will also be sent to the printer. Sending XOFF stops this process. Because the printer device is much slower than the screen, the characters that are to be printed are first placed in a 256 byte buffer. This buffering allows the screen to proceed at a higher speed until the buffer fills. This also means that you may send the XOFF to the printer, and it may keep printing for a while because of the characters remaining in the buffer. Sending a DC2 to the screen enables the keyboard on the serial device (if one exists). From that time, any keys struck on the keyboard before a DC4 is sent to the screen, will appear to the system just as if they were sent by the

Using the Poly 88 BASIC Printer Driver
Copyright Interactive Products Corporation, 1977

normal keyboard on the Poly 88 (NOTE: this INCLUDES control-Y AND control-Z). Because of the type-ahead buffering done by BASIC, some characters may be present in the buffer when the DC4 is sent. These characters may be deleted either by typing PRINT CHR\$(24). Control-X is 18H, or 24 in decimal.

2.3 Using Tapes

Once the printer driver has been installed, a certain amount of care must be used in loading and saving files on cassettes. Because the serial printer interface and the cassette interface run on the same channel, only one may be operational at a time. For this reason,

THE PRINTER AND THE PRINTER KEYBOARD MUST
BE DISABLED BEFORE USING CASSETTES IN BASIC.
THIS MAY BE DONE BY THE DIRECT BASIC STATEMENT
PRINT CHR\$(19), CHR\$(20)

Failure to do this will result in a period of normal, polite behavior, resulting in a sudden and indiscrete stop when the printer buffer fills up. If this happens BASIC must be restarted in a special way to again initialize the printer driver.

2.4 Restarting BASIC with the Printer Driver

If Basic becomes "wedged", or must be restarted, it should be restarted at address 49C0 INSTEAD of 2000 if the printer driver is to be used again. Restarting BASIC at 2000 or 2003 after the front panel RESET button has been used will NOT RECONNECT THE PRINTER DRIVER. Again, when the system is restarted at 49C0, BASIC is COLD STARTED, CLEARING THE PROGRAM.

Using the Poly 88 BASIC Printer Driver
Copyright Interactive Products Corporation, 1977

The printer interface is composed of three distinct sections:

- 1) initialization
- 2) interrupting processing
- 3) character interception.

The initialization section, starting with the label START in the accompanying assembly listing, verifies that the proper version of BASIC is loaded, modifies the starting memory limit in BASIC, and attaches itself to wormhole 1, the character output wormhole. In this manner, all calls to that wormhole will be vectored through the entry point labeled COUT. As characters are sent to wormhole 1, they are examined by COUT. If the character is among XON, XOFF, DC2, or DC4, we transfer to the special processing routine that handles that character. If we see a XON, we go to CXON to set up the interrupt handler and the buffer pointers. TISR is set up as the interrupt processing routine to be called when we get an interrupt from the 8251 USART. The output flag, OFLG is set non-zero to indicate that characters are to be buffered. The ring buffer insertion and removal pointers (TPP and TGP for put and Get) are set, and the USART is started after calling the monitor SETUP routine to define the USART mode. Note that this processing is done with the interrupts DISABLED. This is because the pointers TPP and P are "interrupt alterable", that is, they are altered at the interrupt level. If we did not disable the interrupts at the start of CXON, once we set the address of our interrupt routine, TISR into TINT, the USART could interrupt us, with TPP and TGP having undefined contents (with undefined results!). If the character was not a special one, we test the flag OFLG to see if the printer is enabled.

Using the Poly 88 BASIC Printer Driver
Copyright Interactive Products Corporation, 1977

If the flag is zero, we are not enabled, and we go to CEXIT to return through the normal wormhole processing. If we are buffering characters for the printer, (OFLG non-zero), we transfer to CCR if the character is a carriage return, and to POKE if it is not, (both actions buffer the character) and then we exit.

Because the video driver in ROM performs the equivalent "line feed" action when given a carriage return, we must process carriage returns specially. CCR outputs the carriage return, and a number of padding characters (for DecWriter delay and such), and then outputs a line feed to the buffer. POKE is called to place characters into the ring buffer. We disable interrupts, as we are going to use TPP and TGP, which can be altered by TISR, which runs at the interrupt level. If there is room in the buffer, we place the character in it, and update the pointer (TPP). If there is no room, we go to HANG to enable interrupts and wait. We will wait in this manner until we have room in the buffer for the character. This is the reason the front panel light flashes on and off when driving the printer. The buffer fills up, and we start waiting for an empty slot to put the character in. We enter TISR as a result of an interrupt from the 8251 USART. If it is an interrupt caused by a keyboard character, we test the flag IFLG, and if it is non-zero, we get the character from the USART, and jump into BASIC to process it. If IFLG is zero, we "drop it on the floor". When we detect that the transmitter buffer is empty (the pointer TTP and TGP are equal), we will "give it a fish" - feed it a DEL code. Otherwise, we take the next character from the buffer, update the pointers, and send it to the USART. (The ring is on buffer on a 256 byte boundary, to simplify the coding.) TPP is used as

the "put" pointer, for inserting characters. TCP is used as the "get" pointer. When these pointers are equal, the buffer is empty. If they are equal after one has been decremented (and checked for wrap-around), then the buffer is full. This need only be checked by the insertion routine.

2.6 Special Problems

When driving a terminal such as the HyType, which does not require padding character following the carriage return, the routine PAD may be eliminated. One way of accomplishing this is to modify the first byte of PAD from a 3E to a C9; changing the MV1 into a RET instruction. Making this change eliminates the padding characters from being placed in the buffer. If other than a 300 baud device is used, the bytes following the call to SETUP (location 4A98 in the listing) must be changed to denote the new speed and parity format for the device.

Special thanks go to R. Hustvedt for explaining the techniques used in dynamic patches.

```

;
; Driver for hooking BASIC to Hytypes and Decwriters.
; For use with BASIC version A00 and later.....
;
; Device is to be loaded AFTER starting BASIC. Hooks int
; interrupts, wormholes, and BASIC. Send it an XON, and
; that time until you send it an XOFF, all characters
; sent to the screen thru wormhole 1 will appear on the
; Send it a DC2 (TAPE), and any keystrokes on the thing'
; keyboard will be plotzed into the keyboard buffer. Thi
; stopped by sending a DC4 (not TAPE) to the screen.
;
2000      BGO      EQU      2000H      ; start BASIC
200E      BTXT     EQU      200EH      ; version text in BASIC
205D      BMLL     EQU      205DH      ; lower memory limit in BASIC
206B      BKB      EQU      206BH      ; BASIC keyboard interrupt routi
;
02AD      SETUP   EQU      2ADH        ; usart setup code in rom
0064      IORET   EQU      64H         ; Interrupt return point.
0C16      TINT    EQU      0C16H      ; USART interrupt vector
0C24      WH1     EQU      0C24H      ; wormhole 1 for output
;
000D      CR      EQU      0DH         ; carriage return
000A      LF      EQU      0AH         ; line feed
007F      JUNK    EQU      7FH         ; junk padding code sent
0011      XON     EQU      11H         ; printer on
0013      XOFF    EQU      13H         ; printer off code
0012      DC2     EQU      12H         ; keyboard on
0014      DC4     EQU      14H         ; keyboard off code
49C0      CODE    EQU      49C0H      ; above BASIC
4B00      BUF     EQU      4B00H
4C00      MEND    EQU      BUF+256    ; 256 byte buffer
;
49C0      ORG     CODE
49C0 C3204A JMP     START      ; startup code-check and hook in
;
; Interrupt service routine. We just got an interrupt
; from the stinking 3251.
;
49C3 DB01   TISR:   IN      1          ; what does the thing want
49C5 1F          RAR
49C6 DADE49   JC      WRT          ; jmp/wants a character.
49C9 1F          RAR
49CA D26400   JNC     IORET         ; jmp/just harassing us.
;
; USART has a character. If IFLG is nonzero, foist off o
; BASIC's keyboard interrupt logic, otherwise drop it.
;
49CD 3A1C4A   LDA      IFLG
49D0 B7          ORA      A
49D1 CAD949   JZ      DROP          ; jmp/drop it on the floor.
49D4 DB00          IN      0
49D6 C36820   JMP     SKB          ; leap off into BASIC!
49D9 DB00   DROP:   IN      0          ; get the character
49DE C36400   JMP     IORET         ; and split.
;
; 8251 wants a character to send out.
;

```

```

49DE 2A1A4A WRT: LHLD TGP
49E1 3A184A LDA TPP
49E4 BD CMP L ; anything in buffer?
49E5 CAF649 JZ FISH ; jmp/nope, give it a fish.
49E8 7E MOV A,M ; get chr from buffer
49E9 2D DCR L ; dink pointer
49EA C2EE49 JNZ WRTI ; jmp/no wrap
49ED 2D DCR L ; reset ptr.
49EE 221A4A WRT1: SHLD TGP ; save pointer
49F1 D300 OUT 0 ; shove chr out the door
49F3 C36400 JMP IORET ; and split.
;
; It wants a chr; we don't have one- give it a fish.
;
49F6 3E7F FISH: MVI A,JUNK
49F8 C3EE49 JMP WRTI ; Take that!
;
; Text we search for
;
49FB 506F6C79 TEXT: DB 'Poly 88 BASIC version A'
49FF 20383820
4A03 42415349
4A07 43207665
4A0B 7273696F
4A0F 6E2041
;
; Gripe text
;
4A12 4E757473 NUTS: DB 'Nuts!',0
4A16 2100
;
; Various flags and such.
;
4A18 TPP: DS 2 ; buffer put pointer
4A1A TGP: DS 2 ; buffer get pointer
4A1C IFLG: DS 1 ; input process flag
4A1D OFLG: DS 1 ; output process flag
4A1E WORM: DS 2 ; old wormhole contents
;
; Startup code. Check for the proper version of BASIC.
;
4A20 F3 START: DI
4A21 210E20 LXI H,BTXT
4A24 11FB49 LXI D,TEXT
4A27 0E17 MVI C,23 ; pointers and length to check
4A29 1A CL: LDAX 0
4A2A BE CMP M ; is this the right version?
4A2E C2504A JNZ NOPE ; jmp/nope, I quit.
4A2E 23 INX H
4A2F 13 INX D
4A30 0D DCR C
4A31 C2294A JNZ CL
;
; Now diddle memory limits in BASIC
;
4A34 21004C LXI H,MEND
4A37 225D20 SHLD BHL ; poke!
;
; clear the flags for input and output
;

```

```

4A3A AF          XRA      A
4A3B 321C4A     STA      IFLG
4A3E 321D4A     STA      OFLG
;
; Steal the wormhole for printing on the screen
;
4A41 2A250C     LHLD     WHI+1
4A44 221E4A     SHLD     WORM      ; old contents
4A47 21CF4A     LXI      H,COUT   ; out thing
4A4A 22250C     SHLD     WHI+1   ; hook it up.
;
; Start up BASIC
;
4A4D C30020     JMP      BGO      ; scratch off....
;
; Gripe - this is not the right version of BASIC
;
4A50 21124A     NOPE:    LXI      H,NUTS
4A53 7E         NL:      MOV      A,M
4A54 B7         ORA      A
4A55 CA534A     JZ       NL       ; spin when thru!
4A58 CD240C     CALL    WHI
4A5B 23         INX     H
4A5C C3534A     JMP     NL
;
; POKE puts the thing in A in the buffer. Note that if t
; buffer is full, it will hang you out to dry.
;
4A5F FB         HANG:    EI
4A60 76         HLT
4A61 F1         POP     PSW      ; wait for an interrupt already.
;                                     ; get chr back.
;
4A62 F5         POKE:    PUSH    PSW      ; save this.
4A63 F3         DI      ; don't bug me, I'm busy.
4A64 2A184A     LHLD    TPP
4A67 77         MOV     M,A      ; poke into buffer
4A68 3A1A4A     LDA     TGP
4A6B 2D         DCR    L
4A6C C2704A     JNZ    POKE1
4A6F 2D         DCR    L
4A70 3A1A4A     POKE1:  LDA     TGP      ; see if buffer is full.
4A73 BD         CMP    L
4A74 CA5F4A     JZ     HANG      ; jmp/yup, must wait.
4A77 22134A     SHLD   TPP      ; if not, set pointer
4A7A F1         POP    PSW      ; get thing bck
4A7B FB         EI      ; let the world intrude,
4A7C C9         RET     ; and split.
;
; PAD sends out 10 JUNK characters.
;
4A7D 3E7F     PAD:    MVI     A,JUNK
4A7F 060A     MVI     B,10
4A81 CD624A     PADL:   CALL    POKE
4A84 05         DCR    B
4A85 C2814A     JNZ    PADL
4A88 C9         RET
;
; CX0! handles X0! to start the printer.
;
4A89 F3     CX0N:  DI

```

```

4A8A D5      PUSH    D
4A8B C5      PUSH    B
4A8C 21C349  LXI     H,TISR
4A8F 22160C  SHLD   TINT      ; make sure we have interrupts.
4A92 321D4A  STA    OFLG      ; set flag non-zero, disable
4A95 CDAD02  CALL   SETUP    ; set up USART
4A98 16AA40DA DB      16H,0AAH,40H,0DAH,0
4A9C 00
4A9D C1      POP     B
4A9E D1      POP     D
4A9F 21FF4B  LXI     H,BUF+255
4AA2 22184A  SHLD   TPP
4AA5 221A4A  SHLD   TGP      ; set up buffer pointers
4AA8 3E27    MVI     A,27H
4AAA D301    OUT    I      ; start USART running
;
; CEXIT is the central exit logic
;
4AAC F1      CEXIT:  POP     PSW
4AAD 2A1E4A  LHL    WORM
4AB0 E3      XTHL
4AB1 FB      EI
4AB2 C9      RET
;
; CXOFF handles XOFF sent to us.
;
4AB3 AF      CXOFF:  XRA     A
4AB4 321D4A  STA    OFLG
4AB7 C3AC4A  JMP    CEXIT
;
; CDC"2 is for DC2, to start keyboard up.
;
4ABA AF      CDC4:   XRA     A      ; entry to turn off keyboard
4ABB 321C4A  CDC2:   STA    IFLG
4ABE C3AC4A  JMP    CEXIT
;
; CCR processes a carriage return.
;
4AC1 CD624A  CCR:   CALL   POKE
4AC4 CD7D4A  CALL   PAD      ; cr and nuff padding
4AC7 3E0A    MVI     A,LF
4AC9 CD624A  CALL   POKE    ; and a line feed.
4ACC C3AC4A  JMP    CEXIT
;
; COUT is the entry point for chr output.
; NOTE! we can't leave anything changed!
;
4ACF E5      COUT:  PUSH   H
4AD0 F5      PUSH   PSW
4AD1 FE11    CPI    XON
4AD3 CA894A  JZ     CXON
4AD6 FE13    CPI    XOFF
4AD8 CAB34A  JZ     CXOFF
4ADB FE12    CPI    DC2
4ADD CAB34A  JZ     CDC2
4AE0 FE14    CPI    DC4
4AE2 CABA4A  JZ     CDC4
4AE5 3A1D4A  LDA    OFLG
4AE8 B7      ORA    A      ; do we or dont we?
4AE9 CAAC4A  JZ     CEXIT  ; jmp/we don't.

```


Copyright Interactive Products Corporation,1977

```
4AEC F1          POP      PSW
4AED F5          PUSH     PSW
4AEE FE0D        CPI      CR
4AF0 CAC14A      JZ       CCR      ; jmp/go do CR if needed.
4AF3 CD624A      CALL     POKE     ; jmp/just buffer
4AF6 C3AC4A      JMP      CEXIT    ; if not special, buffer and spl

;
;      That's all, folks!
;

0000            END
```