

EXCHANGING BYTE DATA VIA THE EINSTEIN'S USER PORT AND THE PC'S PRINTER PORT

By Chris Coxall

This page shows two programming examples using the built in assembler of BBCBASIC for the Einstein to input and output bytes via its USER PORT. Programming requires configuring the Einstein's internal Z80 PIO IC.

Two programming listing examples are given for BBCBASIC FOR WINDOWS to run in a MS/WINDOWS PC to transmit and receive bytes which run in conjunction with the BBCBASIC programs given for the Einstein.

Contents

Programming on the PC	PAGE 2
•BBCBASIC FOR WINDOWS - using	
Programming on the Einstein	PAGE 2
• Z80 PIO and interrupt handshaking -described	
Writing to the z80 pio control port	PAGE 3
•Set up for Mode 0 (output) and Mode 1 (input)	
More About The Einstein's Z80 PIO	PAGE 4
•Einstein's printer port -using.	
Program Listings	
•U_OUT_EN.BBC Einstein User Port Output	PAGE 4
•XP_IN.BBC BBC4W's PC printer port Input	PAGE 6
•U_IN_EN.BBC Einstein User Port Inport	PAGE 10
•XP_OUT.BBC BBC4W's PC printer port Output	PAGE 12
Wiring & pinouts	PAGE 17

Programming on The PC using BBCBASIC FOR WINDOWS

The programming for BBCBASIC FOR WINDOWS has been kept to basic basic so to run in a window as would be expected to run in a full screen DOS version of basic or as would be expected to be how the Einstein or other vintage computers would display running programs. There has been no attempt to take advantage of BBCBASIC FOR WINDOWS ability to use windows features such dialogue boxes or drop down menus that can be used with BBCBASIC FOR WINDOWS.

An an exception to simple programming in the listings are system calls to third party WINIO support files to access computer ports directly when using XP and WIN2000. The WINIO files must be present in the same folder that the bbcbasic for windows files are saved to. The intention of simple programming is to show a basic guide which can be followed and understood by others who have a preference to use other programming for writing windows software. The two programs given are just small enough to run in the free Demo version of BBCBASIC FOR WINDOWS. The required third party WINIO support files come with the BBC4W installation package. See BBC4W HELP index-"Input/output using WINIO ".

Download BBCBASIC FOR WINDOWS <http://www.rtrussell.co.uk/>.

It might be possible for the two BBC4W programs given to exchange data by bytes between two windows PC's via their ECCP printer ports. This has not been tried. Download XP_IN.BBC and XP_OUT.BBC in a zip archive from The Tatung Einstein Computer Website <http://tatungeinstein.cedhost.com/mirror/pio.html>

Einstein Programming for the Z80 PIO and interrupt handshaking

•Basic principles for mode 0 output and mode 1 input.

Programming the Z80 PIO for mode 0 (eight bit output) or mode 1 (eight bit input) with hardware handshaking requires an interrupt routine. The interrupt is generated by a high or pulse high received on the STB pin of the Einstein's user port set by strobe from the PC's printer port. An output of a byte to the PIO's data port will set the RDY pin on the user port high to be received by Acknowledge on the PC's printer port for mode 0 output. For input mode 1 the RDY signal is active when the input register is empty and is ready to accept data from the peripheral device.

The BBCBASIC programs for the Einstein's user port U_IN_EN.BBC and U_OUT_EN.BBC have been put into an Einstein Disk Image. Download from this web page. <http://tatungeinstein.cedhost.com/mirror/pio.html>. A 3.5 inch Einstein floppy can be made from the image in a PC floppy 3½" drive with CPCDiskXP software [DOWNLOAD HERE http://www.cpcmania.com/](http://www.cpcmania.com/).

Files in the disk image can be extracted from the image into PC folder using EDIP
SEE PAGE <http://tatungeinstein.cedhost.com/mirror/edip/edip.html>

WRITING TO THE Z80 PIO CONTROL PORT

•FOR MODE 0 and MODE 1

•**WITH INTERRUPTS ENABLED**

%VVXX1111 . Bits D3-D0 indicate "Set Mode".

Last 4 bits set to 1111 Signifies This Control Word is to set mode.

Bits D5 and D4 are ignored.

Bits D7 and D6 set to 01 sets PIO to mode 1 input

Bits D7 and D6 set to 00 sets PIO to mode 0 output

%VVVVVVV0 . Bit 0 set to 0 sets the low byte interrupt vector set by remaining bits.

D0 is used in this case as a flag bit which when low,

causes V7 through V1 to be loaded into the vector register.

0 in D0 Signifies This Control Word is an Interrupt Vector.

%00001100 interrupt vector low byte set to Hex 12.

HEX 12 is the low byte of a scratch pad address given by the Einstein's Machine Operating System at bootup.

The high byte needs to be in the Z80's I register. This is fixed to HEX FB by the Einstein at bootup.

%VVVVV111

Bits 0, 1 and 2 set to 111 This Signifies Control Word is to set interrupt.

%10000111 Bit 7 set to 1 enables interrupts.

A pdf user manual for the Z80 PIO can be downloaded here. <http://www.z80.info/zip/z80piomn.pdf>

More About The Einstein's Z80 PIO

The Z80 PIO has two ports for 8Bit input and output: Port A and B. On the Einstein port A is configured to be compatible as a centronic's printer port at power up. Electronically this is set as mode 0 output but has a hardware adaptor of a monostable which sets ARDY high as a 1us pulse. With the User Port the BRDY is held high until a high is received by PIO's BSTB pin.

Physically the Einstein's printer port pin out in the manual describes STROBE which is connected to the PIO's ASTB and ACKNOWLEDGE which connects to the PIO's ARDY. The printer port's BUSY, PE and ERROR connect to the Einstein's command/status register.

If the Einstein's printer port is used instead of the user port any of the Einstein's software applications print commands will transfer output data but Additional programming would be needed for input.

U_OUT_EN.BBC for the Einstein

```

10 REM U_OUT_EN.BBC EINSTEIN BBCBASIC PROGRAM
20 REM TEST PROGRAM TO OUTPUT DATA BYTES
30 REM TO THE USER PORT USING INTERRUPT HANDSHAKING
40 REM OUTPUT CAN BE RECEIVED BY A PC ECCP PRINTER PORT
50 REM USING BYTE MODE SET FOR IMPORT.
60 REM RECIEVED ASCII DATA BYTES CAN BE READ TO SCREEN BY
70 REM XP_IN_WIN.BBC PROGRAM RUN BY BBCBASIC FOR WINDOWS.
80 REM
90 REM *| WIRING EINSTEIN USER PORT/PC PRINTER PORT
100 REM *|PC data lines D0 to D7 to Einstein D0 to D7
110 REM *|PC strobe to Einstein STB
120 REM *|PC acknowledge to Eistein RDY.
130 REM *| PLUS ONE OR MORE GROUND
140 REM *
150 ON ERROR GOTO 1030
160 HIMEM=HIMEM-80 :REM CREATES SAFE MEMORY AREA ABOVE BBCBASIC
180 DIM code 80
190 code=HIMEM+1
200 REM configpio =start_code+&11 :REM THESE VALUES ARE TO GIVE
210 REM outbyte =start_code+&3F :REM REFERENCE FOR OTHER ASSEMBLERS
220 REM PAOUT =start_code+&28 :REM IF ASSEMBLE CODE IS ALTERED
230 REM CODE SIZE 64 :REM NEW VALUES WILL BE PRINTED OUT
240 REM BY BASIC LINES 800 to 850
250 PRINT "Assembling code at... &";~code
260 PRINT"PRESS ANY KEY TO CONTINUE ":H=GET
270 FOR pass=0 TO 1
280 P%=code
290 [OPT pass*3
300 .start_code
310 .flag DEFB 0 ;LATCH BYTE FOR INTERRUPT
320 .intrrpt ; INTERRUPT ROUTINE
330 PUSH HL ;
340 PUSH DE ;ENDIF REGISTERS
350 PUSH BC ;
360 PUSH AF ;
370 LD HL,flag ;GET ADDRESS OF flag BYTE INTO HL
380 RES 0,(HL) ;SET BIT 0 OF flag TO 0
390 POP AF ; Note RST ops SHOULD NOT BE
400 POP BC ; USED IN AN INTERRUPT ROUTINE
410 POP DE ;RESTORE REGISTERS
420 POP HL ;
430 EI ;ENABLE INTERRUPTS
440 RETI ;RETURN FROM INTERRUPT
450 .configpio
460 LD A,&12 ;sets low byte of interrupt address
470 OUT (&33),A ;writes to USER port reg &12 %00010010
480 LD A,&0F ;sets PIO to mode 0 output
490 OUT (&33),A ;writes to USER port reg &0F %00001111
500 LD A,&87 ;enables interrupts
510 OUT (&33),A ;writes to USER port reg &87 %10000111

```

U_OUT_EN.BBC Einstein listing continued

```

520  PUSH HL
530  LD HL,intrrpt ;CASE HL INTERRUPT START ADDRESS
540    LD (&FB12),HL ;CASE USER PORT SCRATCH PAD INTERRUPT ADDRESS
550    POP HL
560    IM 2      ; ENABLE INTERRUPT 2 MODE
570    RET
580    .PAOUT   ;CODE TO OUTPUT BYTE ENDWHILE IN MEMORY LOCATION outbyte TO
USER PORT
590    PUSH HL
600    PUSH DE      ;ENDIF REGISTERS
610    PUSH BC
620    PUSH AF
630    LD HL,flag   ; FLAG BYTE ADDRESS FOR USER PORT INTERRUPT.
640    .POLL
650    BIT 0,(HL)
660    JR NZ,POLL
670    LD A,(outbyte)
680    OUT (&32),A   ; outport Parallel port data reg.
690    SET 0,(HL)
700    POP AF
710    POP BC
720    POP DE      ;RESTORE REGISTERS
730    POP HL
740    RET
750    .outbyte DEFB 0   ;USER CASES BYTE FOR OUTPUT
760    .end RET
770    ]
780    NEXT pass
790    CALL configpio
800    PRINT "CODE BEGINS AT ADDRESS &";~start_code
810    PRINT "END CODE ADDRESS &";~end
820    PRINT "configpio =start_code+&";~configpio-start_code
830    PRINT "outbyte =start_code+&";~outbyte-start_code
840    PRINT "PAOUT =start_code+&";~PAOUT-start_code
850    PRINT "CODE SIZE "; end-start_code
860    PRINT "PRESS ANY KEY TO START"
870    H=GET
880    REM THE EXAMPLE CODE CODE BELOW CAN BE CHANGED
890    REM THE INSTALLED MACHINE CODE CAN BE USED TO
900    REM SEND ANY BYTE OUT TO THE USER PORT
910    REM BY LOADING outbyte ADDRESS WITH THE
920    REM BYTE TO BE SENT AND CALLING PAOUT
930    REPEAT
940      D$=INKEY$(0)
950      FOR I=32 TO 126
960        ?outbyte=I
970        CALL PAOUT
980        PRINT" ";I;
990      NEXT I

```

U_OUT_EN.BBC Einstein listing continued

```

1000      UNTIL D$="A"
1010      PRINT"END END END END "
1020      STOP
1030      REM
1040      REPORT:PRINT "AT LINE ";ERL:STOP

```

XP_IN.BBC for the PC

```

*|      XP_IN.BBC
*|PC WINDOWS PROGRAM TO RUN IN BBCBASIC FOR WINDOWS
*|Windows BBC4W program XP_IN_large.BBC works with
*|U_OUT_EN.BBC running in the Einstein
*|
*|It works in conjunction with the Einstein
*|with linked PC ECCP printer port with the user
*|port on the Einstein. It will output data bytes
*|to the program U_IN_EN.BBC running on the Einstein.
*|
*|      WIRING EINSTEIN USER PORT
*|PC data lines D0 to D7 to Einstein D0 to D7
*|PC strobe to Einstein STB
*|PC acknowledge to Einstein RDY.
*|Plus one or more ground lines.
*|
*|
*|NEXT LINES CHECK TO SEE IF WINIO FILES ARE IN THE SAME DIRECTORY
SYS "GetFileAttributesA", "winio.vxd" TO ret%
IF ret%=-1 PRINT "ERROR winio.vxd not in directory ""@dir$:STOP ELSE PRINT"winio.vxd ok"
SYS "GetFileAttributesA", "winiov1.sys" TO ret%
IF ret%=-1 PRINT "ERROR winio1.sys not in directory ""@dir$:STOP ELSE PRINT"winiov1.sys ok"
SYS "GetFileAttributesA", "winiov1.dll" TO ret%
IF ret%=-1 PRINT "ERROR winio1.dll not in directory ""@dir$:STOP ELSE PRINT"winiov1.dll ok"
*|NEXT TWO LINES TO SHUT DOWN WINIO ON PROGRAM ERRORS AND AT CLOSE
*|IF WINIO IS NOT CLOSED IT WILL NOT ALWAYS INSTALL AGAIN WITHOUT A REBOOT
ON CLOSE SYS ShutdownWinIo% :PRINT:REPORT:STOP
ON ERROR SYS ShutdownWinIo% :PRINT:REPORT:STOP
datap%=&378:      *|PRINTER LPT1 PORTS
status%=&378+1:   *|
controll%=&378+2: *|
eccport%=&378+&402: *|
DIM dataport% 1: *|MEMORY LOCATIONS TO STORE
DIM eccp% 1 :    *|PRINTER PORT READ AND WRITE BYTES
DIM statusp% 1 : *|FOR WINIO SYSTEM CALLS
DIM controllp% 1: *|SEE HELP Input/output using WINIO

```

XP_IN.BBC PC listing continued

```

*| *****
*| *****
*| ***** LOADS THIRD PARTY WINIO LIBRARY *****
*| FILES WINIO.DLL, WINIO.SYS and WINIO.VXD must be in the
*| same directory (folder) as the executable program using them.
*|Note to install WINIO IN 2000 and XP administrative privileges ARE NEEDED
*|*****
SYS "LoadLibrary", "WINIO.DLL" TO winio%
IF winio% = 0 ERROR 0, "Could not load WINIO"
SYS "GetProcAddress", winio%, "InitializeWinIo" TO InitializeWinIo%
SYS "GetProcAddress", winio%, "ShutdownWinIo" TO ShutdownWinIo%
SYS "GetProcAddress", winio%, "GetPortVal" TO GetPortVal%
SYS "GetProcAddress", winio%, "SetPortVal" TO SetPortVal%
SYS InitializeWinIo% TO ok%
IF (ok% AND 1) = 0 ERROR 0, "Could not initialise WINIO"
*| *****
*|
*|*****
PROC_set_byte_mode: *|SETS UP ECCP PARALLEL FOR BYTE MODE
PRINT
PROC_controll_input: *|SETS PRINTER PORT FOR OUT PUT
PRINT"RUN U_OUT_EN.BBC ON THE EINSTEIN THEN "
PRINT"PRESS ANY KEY TO RECEIVE INPUT FROM PRINTER PORT"
H=GET
PRINT""TO CANCEL PRESS ESCAPE ON THE EINSTEIN KEY BOARD FIST ""\
\"THEN ESCAPE ON THE PC KEYBOARD. "
*| *****
*| *****
*| *****
*| *****
*|***** MAIN PROGRAM FOR EXPORTING Bytes TO THE PRINTER PORT
*****
*|
*|
pollbyte=0: *| var to latch an acknowledge pulse
REPEAT
key%=INKEY$(0)
PROC_get
data%=?^dataport%
IF data%=13 PRINT CHR$(data%)
IF data% =>32 AND data%=<126 PRINT CHR$(data%);
UNTIL key$="A" OR key$="a"
SYS ShutdownWinIo%
STOP
*| ***** END OF TO RECEIVE DATA *****
*| *****

```

XP_IN.BBC PC listing continued

```

*| *****DEF PROC_get*****
*|WHEN THE LPT1 PORT HAS BEEN INITIATED AND WINIO INSTALLED
*|PROC_get IS THE ONLY PROCEDURE NEEDED FOR THE USER TO IMPORT A BYTE
*|FROM THE pc PRINTER PORT data%=?^dataport% PUTS IMPORTED BYTE IN data% VAR
DEF PROC_get
REPEAT:PROC_poll_ack:UNTIL pollbyte<>0
PROC_strobe_low
SYS GetPortVal%, datap%, ^dataport%, 1
pollbyte=0
PROC_strobe_high
ENDPROC
*| *****

*| ***** PROC_set_byte_mode DOES WHAT IS SAYS *****
DEF PROC_set_byte_mode
SYS GetPortVal%, eccport%, ^eccp%, 1
byte=?^eccp%
byte=byte AND %00111111
byte=byte OR %00100000
SYS SetPortVal%, eccport%, byte, 1
SYS GetPortVal%, eccport%, ^eccp%, 1
byte=?^eccp%
D$=FN_BIN(byte)
PRINT ""ENHANCED PARALLEL PORT"
PRINT"  BIT PATTEN          ";D$
PRINT" Needed to set ECP port to byte mode.  %001XXXXX"
IF (byte AND %11100000)=%00100000 PRINT " BYTE MODE SET " ELSE PRINT"BYTE MODE NOT
SET "
ENDPROC

*| ***** PROC_controll_input *****
*| ***** SETS LPT1 PRINTER PORT FOR INPUT
*| ***** DISABLES INTERRUPTS
*| ***** SETS STROBE LINE LOW (strobe hardware inverted)
DEF PROC_controll_input
SYS GetPortVal%, controll%, ^controllp%, 1
ctrl=ctrl OR %00100001 :*| ; set bits 0 to 1 inverted 1 sets STROBE line to low.
ctrl=ctrl AND %11101111 :*| ; bit 5 set to 1 for output
: :*| ; bit 4 set to 0 no interrupts set
SYS SetPortVal%, controll%, ctrl, 1
SYS GetPortVal%, controll%, ^controllp%, 1
ctrl=?^controllp%
D$=FN_BIN(ctrl)
PRINT "CONTROLL PORT BIT PATTEN          ";D$
PRINT " Needed bit 5 set to 1 for data output %XX1XXXXX"
PRINT " To set bit 0 to 1. Sets STROBE line to low. %XXXXXXX1"
PRINT " STROBE line hardware inverted"
PRINT " Bit 4 set to 0 for no interrupts set. %XXX0XXXXX"

```


XP_IN.BBC PC listing continued

```

IF (ctrl AND %00100000)=32 PRINT "      SET UP FOR IMPORT INSTALLED " ELSE PRINT
"ERROR"
PRINT
ENDPROC

```

```

DEF PROC_poll_ack
SYS GetPortVal%, status%, ^statusp%, 1
stat=?^statusp%
ack%=stat AND %01000000
IF ack%<>0 pollbyte=1
ENDPROC

```

```

DEF PROC_strobe_high
SYS GetPortVal%, controll%, ^controllp%, 1
ctrl=?^controllp%
ctrl=ctrl AND %1111110 :*| ; set bit 0 to 0. Inverted-0 sets STROBE line to high.
SYS SetPortVal%, controll%, ctrl, 1
ENDPROC

```

```

DEF PROC_strobe_low
SYS GetPortVal%, controll%, ^controllp%, 1
ctrl=?^controllp%
ctrl=ctrl OR %00000001 :*| ; set bit 0 to 1. Inverted-1 sets STROBE line to low.
SYS SetPortVal%, controll%, ctrl, 1
ENDPROC

```

```

*| ***** FUNCTION PRINTS OUT BINARY STRING OF VAR *****
*| E,G VAR NUMERIC VALUE 7 WOULD RETURN STRING VAR %00000111
DEF FN_BIN(A%):LOCAL A$
REPEAT A$=STR$(A% AND 1)+A$:A%=A% >>> 1:UNTIL LEN A$=8:A$="%"+A$
=A$
*| *****

```

U_IN_EN.BBC for the Einstein

```

10 REM U_IN_EN.BBC EINSTEIN BBCBASIC PROGRAM
20 REM TEST PROGRAM TO IMPORT DATA BYTES
30 REM FROM THE USER PORT USING INTERRUPT HANDSHAKING
40 REM IMPORT CAN BE RECEIVED FROM A PC ECCP PRINTER PORT
50 REM RUNNING OUT_XP_large IN BBCBASIC FOR WINDOWS
60 REM ECCP SET TO BYTE MODE AND FOR IMPORT.
70 REM
80 REM *| WIRING EINSTEIN USER PORT/PC PRINTER PORT
90 REM *|PC data lines D0 to D7 to Einstein D0 to D7
100 REM *|PC strobe to Einstein STB
110 REM *|PC acknowledge to Einstein RDY.
120 REM *| PLUS ONE OR MORE GROUND
130 REM *
140 ON ERROR GOTO 1110
150 HIMEM=HIMEM-80
170 usrcon=&33
180 usrdata=&32
190 DIM code 80
200 code=HIMEM+1
210 PRINT "Assembling...code at &";~HIMEM+1
220 PRINT"PRESS ANY KEY TO CONTINUE ":H=GET
230 FOR pass=0 TO 1
240 P%=code
250 [OPT pass*3
260 .start_code
270 .flag DEFB 0
280 .intrrpt ;interrupt routine
290 PUSH HL
300 PUSH DE ;save registers
310 PUSH BC
320 PUSH AF
330 LD HL,flag
340 RES 1,(HL) ;at an interrupt byte at address flag has bit 1 set to zero.
350 POP AF
360 POP BC ;retrieve registers
370 POP DE
380 POP HL
390 EI ;enables interrupts
400 RETI ;returns from interrupt routine.
420 .configpio ;ROUTINE TO CONFIGER USER PORT FOR MODE 1 and INTERRUPTS
ENABLED
430 LD A,&12 ;sets low byte of interrupt address
440 OUT (usrcon),A ;writes to user port CONTROL reg &12 %00001100
450 ;sets low byte of interupt vector address
460 ;high byte of interupt vector address in Z80 I reg
470 ;set by Einstein at boot up to &FB.
480 LD A,&4F ;sets PIO to mode 1
490 OUT (usrcon),A ;writes to Einstein user port reg &4F %01001111
500 LD A,&87 ;enables interrupts
510 OUT (usrcon),A ;writes to parallel user port control reg &87 %10000111
520 PUSH HL

```

```

530 LD HL,intrrpt ; CASE reg HL address of interupt routine.
540 LD (&FB12),HL ; CASE interupt address into interupt vector at &FB12.
550 POP HL
560 IM 2 ; ENABLE INTERRUPT 2 MODE
570 RET
580 .getbyte ;ROUTINE TO IMPORT A SINGLE BYTE FROM THE USER PORT
590 PUSH HL
600 PUSH DE ;save registers on the stack
610 PUSH BC
620 PUSH AF
630 LD HL,flag
640 .POLL ;LOOP to poll bit 1 of flag set by interrupt routine.
650 LD HL,flag
660 BIT 1,(HL)
670 JR NZ,POLL ;IF flag bit 1 not zero loop back.
680 IN A,(usrdata) ;IF flag bit 1 zero get byte from user port into "A" register.
690 LD (imput_byte),A ;Put byte value of imported byte now in "A" register
700 ;into memory location "imput_byte"
710 SET 1,(HL) ;resets bit 1 of byte at memory location "flag" to 1.
720 POP AF
730 POP BC ;retrieve registers from stack.
740 POP DE
750 POP HL
760 RET
770 .imput_byte DEFB 0
780 .end
790 ]
800 NEXT pass
810 CALL configpio
820 PRINT "CODE BEGINS AT ADDRESS &";~start_code
830 PRINT "END CODE ADDRESS &";~end
840 PRINT "flag=start_code+&";~flag-start_code
850 PRINT "configpio =start_code+&";~configpio-start_code
860 PRINT "getbyte =start_code+&";~getbyte-start_code
870 PRINT "imput_byte=start_code+&";~imput_byte-start_code
880 PRINT "CODE SIZE "; end-start_code
890 PRINT "PRESS ANY KEY TO START"
900 REM USER PROGRAMING CAN BE ADDED HERE
910 REM RECEIVED IMPORT BYTES FROM USER PORT
920 REM ARE FOUND BY CALLING getbyte THEN
930 REM PEEKING THE ADDRESS imput_byte
940 REM "n=?imput_byte" IN BBCBASIC
950 REM
960 REM IN THE EXAMPLE PROGRAM BELOW
970 REM BYTE VALUES 32 to 126 TO BE PRINTED
980 REM TO SCREEN AS ASCII CHARACTERS IN THE PC
990 REM WITH BBC4W RUNNING P_IN_WIN.BBC
1000 REM
1010 PRINT "PRESS ANY KEY TO RUN EXAMPLE PROGRAM"
1020 H=GET
1030 REPEAT
1040 D$=INKEY$(0)

```

U_IN_EN.BBC Einstein listing continued

```
1050      n=?input_byte
1060      CALL getbyte
1070      IF n=13 PRINT
1080      IF n=>32 AND n=<126 PRINT CHR$(n);
1090      UNTIL D$="A" OR n=27
1100      STOP
1110      REPORT:PRINT "AT LINE ";ERL:STOP
```

XP_OUT.BBC for the PC

*OUT_XP.BBC is just small enough to run in
*the trial BBC4W demo version.
*If additional basic lines are added delete
*comments before running in demo version
*a NO ROOM error could mean rebooting
*before WINIO will install again.
*|
*| XP_OUT_large.BBC
*PC WINDOWS PROGRAM TO RUN IN BBCBASIC FOR WINDOWS
*Transfers bytes from the PC printer port to
*Einstein's USER PORT.
*Requires WINIO support files WINIO.VXD, WINIOV1.DLL
*and WINIOV1SYS in the same folder as this program.
*|
*Windows BBC4W program OUT_XP_large.BBC works with

XP_OUT.BBC PC listing continued

*|U_IN_EN.BBC running in the Einstein to send bytes
 *|from the PC printer port to the Einstein's user
 *|port.

*|
 *| WIRING EINSTEIN USER PORT
 *|PC data lines D0 to D7 to Einstein D0 to D7
 *|PC strobe to Einstein STB
 *|PC acknowledge to Einstein RDY.
 *|Plus one or more ground lines.

*|
 *|
 *|This BBC4W Prog will work for win95/98 AND XP
 *|The program is intended to poll acknowledge for
 *|a pulse as output by RDY pin 5 of the
 *|Einstein's Z80 PIO user port.

*|
 *|
 *| WHEN Linked To The Einstein User Port
 *|All handshaking lines work as expected and data
 *|can be received by U_IN_EN.BBC when run on the Einstein.

datap%=&378: *|PRINTER LPT1 PORTS
 status%=&378+1: *|
 controll%=&378+2: *|
 eccport%=&378+&402: *|
 DIM dataport% 1: *|MEMORY LOCATIONS TO STORE
 DIM eccp% 1 : *|PRINTER PORT READ AND WRITE BYTES
 DIM statusp% 1 : *|FOR WINIO SYSTEM CALLS
 DIM controllp% 1: *|SEE HELP INDEX "Input/output using WINIO"

*|
 *|NEXT LINES CHECK TO SEE IF WINIO FILES ARE IN THE SAME DIRECTORY
 SYS "GetFileAttributesA", "winio.vxd" TO ret%
 IF ret%=-1 PRINT "ERROR winio.vxd not in directory ""@dir\$:STOP ELSE PRINT"winio.vxd ok"
 SYS "GetFileAttributesA", "winiov1.sys" TO ret%
 IF ret%=-1 PRINT "ERROR winio1.sys not in directory ""@dir\$:STOP ELSE PRINT"winiov1.sys ok"
 SYS "GetFileAttributesA", "winiov1.dll" TO ret%
 IF ret%=-1 PRINT "ERROR winio1.dll not in directory ""@dir\$:STOP ELSE PRINT"winiov1.dll ok"
 *|NEXT TWO LINES TO SHUT DOWN WINIO ON PROGRAM ERRORS AND AT CLOSE
 *|IF WINIO IS NOT CLOSED IT WILL NOT ALWAYS INSTALL AGAIN WITHOUT A REBOOT
 ON CLOSE SYS ShutdownWinIo% :STOP
 ON ERROR SYS ShutdownWinIo% :STOP

*| *****
 *| *****
 *| ***** LOADS THIRD PARTY WINIO LIBRARY *****
 *| FILES WINIO.DLL, WINIO.SYS and WINIO.VXD must be in the
 *| same directory (folder) as the executable program using them.
 *|Note to install WINIO IN 2000 and XP administrative privileges ARE NEEDED

XP_OUT.BBC PC listing continued

```

*****
SYS "LoadLibrary", "WINIO.DLL" TO winio%
IF winio% = 0 ERROR 0, "Could not load WINIO"
SYS "GetProcAddress", winio%, "InitializeWinIo" TO InitializeWinIo%
SYS "GetProcAddress", winio%, "ShutdownWinIo" TO ShutdownWinIo%
SYS "GetProcAddress", winio%, "GetPortVal" TO GetPortVal%
SYS "GetProcAddress", winio%, "SetPortVal" TO SetPortVal%
SYS InitializeWinIo% TO ok%
IF (ok% AND 1) = 0 ERROR 0, "Could not initialise WINIO"
*****
*| *****
*|
PROC_set_byte_mode:  *|SETS UP ECCP PARALLEL FOR BYTE MODE
PRINT
PROC_controll_output:  *|SETS PRINTER PORT FOR OUT PUT
*| *****
*| *****
*| *****
*|***** MAIN PROGRAM FOR EXPORTING Bytes TO THE PRINTER PORT
*****
*|
*|
pollbyte=0: REM var to latch an acknowledge pulse
PRINT "  MAIN PROGRAM FOR EXPORTING BYTES TO THE PRINTER PORT "
PRINT "PRESS ANY KEY TO CONTINUE MAIN PROGRAM "
H=GET
PRINT "RUN U_IN_EN.BBC on the Einstein. """"TO CANCEL PRESS ESCAPE ON THE EINSTEIN
KEY BOARD FIST ""\
\" THEN ESCAPE ON THE PC KEYBOARD. "
*|REPEAT and FOR NEXT loop create an example program.
REPEAT
FOR i=32 TO 126
D$=INKEY$(0)
*|The next four lines get a single byte from
*|the printer port and set handshaking.
PROC_strobe_low
REPEAT:PROC_poll_ack:UNTIL pollbyte=1
SYS SetPortVal%, datap%, i, 1
pollbyte=0
PROC_strobe_high
NEXT i
UNTIL D$="A"
SYS ShutdownWinIo%
STOP
*| *****
*|
*| ***** PROC_set_byte_mode DOES WHAT IS SAYS *****

```

XP_OUT.BBC PC listing continued

```

DEF PROC_set_byte_mode
SYS GetPortVal%, eccport%, ^eccp%, 1
byte=?^eccp%
byte=byte AND %00111111
byte=byte OR %00100000
SYS SetPortVal%, eccport%, byte, 1
SYS GetPortVal%, eccport%, ^eccp%, 1
byte=?^eccp%
D$=FN_BIN(byte)
PRINT "ENHANCED PARALLEL PORT"
PRINT "  BIT PATTEN          ";D$
PRINT" Needed to set ECP port to byte mode.    %001XXXXX"
IF (byte AND %11100000)=%00100000 PRINT " BYTE MODE SET " ELSE PRINT"BYTE MODE
NOT SET "
ENDPROC

```

```

*| ***** PROC_controll_output *****
*| ***** SETS LPT1 PRINTER PORT FOR OUTPUT
*| ***** DISABLES INTERRUPTS
*| ***** SETS STROBE LINE LOW (strobe hardware inverted)

```

```

DEF PROC_controll_output
SYS GetPortVal%, controll%, ^controllp%, 1
REM PROC_ctrl
ctrl=ctrl OR %00000001    :*| ; set bits 0 to 1 inverted 1 sets STROBE line to low.
ctrl=ctrl AND %11001111    :*| ; bit 5 set to 0 for output
:                          :*| ; bit 4 set to 0 no interrupts set
SYS SetPortVal%, controll%, ctrl, 1
SYS GetPortVal%, controll%, ^controllp%, 1
ctrl=?^controllp%
D$=FN_BIN(ctrl)
PRINT "CONTROLL PORT BIT PATTEN          ";D$
PRINT " Needed bit 5 set to 0 for data output    %XX0XXXXX"
PRINT " Needed bit 4 set to 0 for NO interrupts    %XXX0XXXXX"
PRINT " To set bit 0 to 1. Sets STROBE line to low. %XXXXXXX1"
PRINT " STROBE line hardware inverted"
ENDPROC

```

```

DEF PROC_poll_ack
SYS GetPortVal%, status%, ^statusp%, 1
stat=?^statusp%
ack%=stat AND %01000000
IF ack%<>0 pollbyte=1
ENDPROC

```

```

DEF PROC_strobe_high
SYS GetPortVal%, controll%, ^controllp%, 1
ctrl=?^controllp%
ctrl=ctrl AND %11111110    :*| ; set bit 0 to 0. Inverted-0 sets STROBE line to high.

```

XP_OUT.BBC PC listing continued

```
SYS SetPortVal%, controll%, ctrl, 1
ENDPROC
```

```
DEF PROC_strobe_low
SYS GetPortVal%, controll%, ^controllp%, 1
ctrl=?^controllp%
ctrl=ctrl OR %00000001 :*| ; set bit 0 to 1. Inverted-1 sets STROBE line to low.
SYS SetPortVal%, controll%, ctrl, 1
ENDPROC
```

```
*| ***** FUNCTION PRINTS OUT BINARY STRING OF VAR *****
*| E,G VAR NUMERIC VALUE 7 WOULD RETURN STRING VAR %00000111
DEF FN_BIN(A%):LOCAL A$
REPEAT A$=STR$(A% AND 1)+A$:A%=A% >>> 1:UNTIL LEN A$=8:A$=" "+A$
=A$
*| *****
```

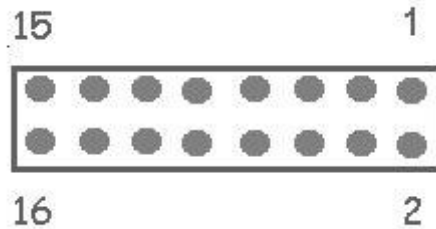

WIRING LINK

USER PORT			PC PRINTER PORT	
PIN			PIN	
2	D0	to	2	D0
4	D1	to	3	D1
5	RDY	to	10	ACK
6	D2	to	4	D2
8	D3	to	5	D3
10	D4	to	6	D4
11	STB	to	1	STB
12	D5	to	7	D5
14	D6	to	8	D6
16	D7	to	9	D7

One or more pins
3 7 9 or 13 GROUND to pins
18-25 GROUND

USER PORT PIN OUT

1	5V
2	D0
3	0V
4	D1
5	RDY
6	D2
7	0V
8	D3
9	0V
10	D4
11	STB
12	D5
13	0V
14	D6
15	5V
16	D7



Einstein User Port

PC PRINTER PORT PIN OUT

1	STB
2	D0
3	D1
4	D2
5	D3
6	D4
7	D5
8	D6
9	D7
10	ACK
11	BSY
12	PAPER
13	SEL
14	AF
15	ERROR
16	INI
17	DSL
18-25	GROUND



PC Printer Port