

## BASIC X - BASIC TAPE PRINTER SOUND ENHANCEMENTS - BASIC X

### INITIALISING BASIC X

If you have BASIC 5 or TOOLKIT, these must be initialised first.

Now initialise BASIC X by jumping to the first address of the supplied BASIC X EPROM ( 8800 or 9800 ). If you are using BASIC X alone, go to the monitor (RESET M) and then type 8800 or 9800 followed by 'G'.

If you have BASIC 5, use the GO\$ command to gain entry to BASIC X.

Disk users should note that BASIC X will work with ROMDOS, but not with any of the versions of OS65D.

### BASIC X Vectors

The print vector is used for printer support. To disable auto reset of the vector use

POKE3,76:POKE4,199:POKE5,168

which disables vector initialisation.

BASIC X offers many new words. All are accessible from program or the immediate mode. No prefix is needed before the new words - they become 'built-in'.

The new words are:-

PRINTER	SOUND	BASIC & TAPE
VIA	PSG	REPEAT...UNTIL
PAR	RESET	HANG
LFO	NOISE	ELSE
RSPO	CHAN	APPND
LPRNT	ENV	CREATE
LISP	VOL	OPEN
IN	MODE	CLOSE
OUT	REG	DUMP
	XREG	GET

PRINTER PRINTER PRINTER PRINTER PRINTER

### Serial Printer

This is connected to the RS232C interface on the UK101/OHIO. See your UK101 or OHIO users manual. In case of difficulty, contact PREMIER.

### Parallel Printer

The parallel printer driver works through your 6522 VIA. Connect the pins as follows:-

VIA connection	PRINTER connection
D0 Port B	D0
D1 Port B	D1
D2 Port B	D2
D3 Port B	D3
D4 Port B	D4
D5 Port B	D5
D6 Port B	D6
D7 Port B	D7
CB1 Port B	ACK
CB2 Port B	STROBE

If you want a parallel output to drive a newly bought printer and have not yet purchased the necessary interface, PREMIER produce a VIA board which naturally works admirably with BASIC X! See current price list for costs.

### PRINTER COMMANDS

#### VIA n

This tells BASIC X where your 6522 VIA can be found on the memory map. 'n' is the address of the first register of the 6522, in decimal. Used for parallel printer.

#### PAR

This command tells BASIC X that you are using a parallel printer.

#### LFO

Stops all linefeeds being sent to parallel printer. This command should be used if your printer gives automatic line-feed after carriage return. To ascertain this, consult the manual supplied with your printer.

#### RSPO

Reset printer option to those on initialisation. ie serial printer.

### LPRNT

This has the same format as PRINT in BASIC, but the printout will also go to the printer. This command is especially useful where output from a program (results of a test, etc) are sporadic.

### LISP

This lists the current BASIC program on the printer. The LISP instruction supports the normal LIST syntax. Eg:-

LISP-999 will list the program to printer up to line 999

### IN

Switches IN the printer until another printer command is executed, OR program returns to immediate mode.

### OUT

Switches the printer OUT.

## SOUND GENERATOR

Check that your programmable sound generator has the number AY-3-8910 or AY-3-8912 on it. If it has not the sound commands may not work. If you have more than one sound chip, PSG allows a powerful flexibility between them as switching between chips becomes simple.

If you have the PREMIER PSG, ignore the next paragraph.

The address in memory of your PSG is required if you are not using the PREMIER board. This is the location you poke in order to set up the address of the internal register. The sound locations MUST be consecutive in memory; if your sound board works through a PIA the sound commands in BASIC X will not work. Before using Sound, execute PSGA n where n = the memory location of the sound chip in memory (NOT PREMIER PSG).

### SOUND COMMANDS      SOUND COMMANDS      SOUND COMMANDS

#### RESET

resets the sound generator (ie. turns it off) on the PREMIER PSG only, or one located at the same addresses. For other systems use

```
10 FORT=0TO15:REG,0:NEXT: REG7,255
```

#### NOISE n

This sets the frequency of the random noise. n should be in range 0....31. If the value of n is one, then a 2KHz frequency will be generated (when running a normal 1Mhz clock).

#### CHAN n, x, y

This command sets the tonal output on channel n, where n is in the range 1 to 3, with coarse tune x and fine tune y. The n value entered refers to channel A, B and C on the sound chip data sheet. Ranges of values are:-

n	1 to 3
x	0 to 15
y	0 to 255

#### ENVx,y,n

This command is used to control the envelope shape and duration of sound. The ranges are:-

x	0 to 255
y	0 to 255
n	0 to 15

The variable n refers to the envelope shape and you should consult the data sheet for its effect. The command

ENV200,100,0

will put 100 in register 11, 200 in register 12 and 0 in register 13. This effect will produce a 'single shot' output.

#### VOLn,x

sets up the volume on channel n to value x. Ranges are

n	1 to 3
x	0 to 16

The normal range of x is 0-15; the value 16 should only be used if envelope control is required.

VOL1,12

will put twelve in register B, channel 1.

#### MODE"xyz"

The mode instruction sets the function of each channel according to the contents of the string, thus allowing total flexibility as the string may be manipulated like a normal BASIC string

x	= channel 1
y	= channel 2
z	= channel 3.

Each character can be one of the following.

0	= channel off
1	= noise
2	= tone
3	= tone and noise.

thus MODE"021" will enable no output on channel 1, tone on channel 2, and noise on channel 3 which will effectively put 221 in register 7.

#### REGn,x

sets the register n with the value x, where the range for x is 0 to 255 or as applicable to each register contents range on the data sheet. The range for n is 0 to 15. Thus REG1,8 will put 8 into register.

There is no error checking to ensure that you do not exceed the maximum value that any particular register will store. This means that REG8,32 will in fact put 0 into register 8. A look at the numbers 32 and 31 in binary form will demonstrate this better.

31	00011111	where the five bits used are
32	00100000	
	---xxxxx	



Another point worth noting is that register 7 is an active low select - to enable any function the relative bit of register 7 needs to be low '0' and all others need to be high '1'. Therefore for no output set all the bits high with REG7,255

## XREG

will load variable x with the contents of register n. Try  
FORJ=0TO15:XREGJ,V:PRINTJ,V:NEXT

or

XREG7,J:PRINTJ

and you will see that the variable 'J' now equals the contents of register 7.

\*\*\*\*\*

Here's a program to try with sound:-

```
10 MODE"220":VOL1,15:VOL2,13:TI=140
20 READA,B,C:IFB=200THENRESET:END
30 CHAN1,0,A:CHAN2,B,C:FORJ=-0TOTI:NEXT:GOTO20
40 DATA81,,,90,,,81,,,90,,,81,,,113,,,91,,,103,,,0
50 DATA123,1,212,123,1,83,,,210,210,,,130,,,123,,0
60 DATA113,2,123,113,1,83,,1,13,130,,,131,,,113,,0
70 DATA103,1,212,103,1,83,,,210,130,,0
80 DATA81,,,90,,,81,,,90,,,81,,,113,91,,,103,,0
90 DATA123,1,212,123,1,83,,,210,210,,,130,,,123,,0
100 DATA113,2,123,113,1,83,,1,13,130,,,103,,,113,,0
110 DATA123,1,212,123,1,83,,,210,,200,0
```

## NEW BASIC WORDS

### REPEAT,...UNTIL

This is a new loop structure. The instructions after REPEAT are executed until the UNTIL is reached. If the conditions after UNTIL fails then the instructions are repeated. Try

REPEAT:A=A+1:?A;UNTIL A=10

in the immediate mode. Note the colon before UNTIL (if it's not the first command on a line). UNTIL may be on the same line as REPEAT or anywhere below it. REPEAT....UNTIL loops may be nested - same conditions apply as for FOR....NEXT loops. Example:-

```
10 REPEAT
20 X=X+1
30 Y=X*X+Y
40 rest of program
900 UNTILX > 58
```

### HANG

Causes program execution to stop until a key is pressed. PEEK(531) gives ASCII value of key pressed.

```
10 X=2
20 IF X=2 THEN Y=3 : HANG : ?PEEK(531)
```

### IF....THEN....ELSE

IF...THEN functions as normal. Instructions after ELSE are executed only if the last IF....THEN failed its test.Try:-

```
10 INPUTA
20 IFA<10THEN?"less than 10":ELSE?"over 10"
```

ELSE may also be teamed with IF to produce ELSE IF, although the restricted line length of the UK101/OHIO computers makes it not as useful as it might be!

### TAPE COMMANDS

APPND"filename"

Allows PREMIER NEW BASIC 4 users to APPEND a program from tape (load and merge with resident program).

### DATA LOAD AND SAVE

CREATE "filename"

This creates a datafile called "filename". It must precede any DATA dumps.

OPEN "filename"

This searches for the named file on tape for retrieving DATA.

CLOSE

Closes a file after saving data. It is not essential but good practice.

DUMP variable(s)

The DUMP command places on tape the contents of the named variables (numeric or string). When dumping more than one variable, separate with commas.

GET variable(s)

Use after opening a file to read in variables. For more than one variable, variables should be separated with commas.

EXAMPLE

```
10 PRINT "PRESS A KEY AFTER STARTING TAPE"
20 HANG:PRINT "SAVING FILE"
30 CREATE "A FILE"
40 FORT=1TO100
50 DUMPT
60 NEXT:CLOSE
70 PRINT "REWIND TAPE AND PRESS PLAY"
80 CLEAR
90 DIMA(100):FORT=1TO100:A(T)=0:NEXT
100 OPEN "A FILE"
110 FORT=1TO100
120 GETA(T):NEXT
130 PRINT "LOADED"
140 INPUT "ARRAY ELEMENT TO TEST (1-100)";N
150 PRINT A(N)
160 GOTO 140
```

When generating and saving data files that contain strings, ensure that ALL strings contain something - do NOT send empty strings to tape. If you do send an empty string to tape, errors will occur when you attempt to read it back. To overcome this problem, try to ensure that all strings which are liable to be saved to cassette contain something, even if it's only a couple of spaces. This routine may be useful:-

```
FOR J=1 TO X : A$(J)=" " : NEXT
```

This type of routine will ensure that all strings contain data.

## ERROR MESSAGES

LOST SYNC

Data lost due to failure to read a variable. Also caused by corrupted data.

GO ERROR

:UNTIL executed without corresponding REPEAT.

IMPORTANT NOTICE - BASIC X, its subroutines and all printed matter appertaining to the use thereof are COPYRIGHT of PREMIER PUBLICATIONS. Copying of the whole or of any part in any medium other than for the personal use of the original retail purchaser is strictly prohibited. We are obliged to pursue an active policy against infringements of our copyrights. PREMIER PUBLICATIONS products are only available by mail order from the address below or from accredited agents. Each BASIC X product is uniquely coded.

Quality of workmanship and materials fully guaranteed. Claims under guarantee MUST be accompanied by the program EPROM. Every care has been taken in the preparation of this product. PREMIER PUBLICATIONS will not however be responsible for claims of loss or of loss of profit howsoever caused from the use or content of this product. It is the purchaser's responsibility to ensure suitability for purpose.

THIS NOTICE DOES NOT AFFECT THE STATUTORY RIGHTS OF THE ORIGINAL RETAIL PURCHASER.

BASIC X was written by D Anderson, the manual by J Hooker.

<C> 1982 PREMIER PUBLICATIONS World Copyright retained

208 CROYDON RD, ANERLEY, LONDON, SE20 7YX

Telephone 01 - 659 - 7131