FROM

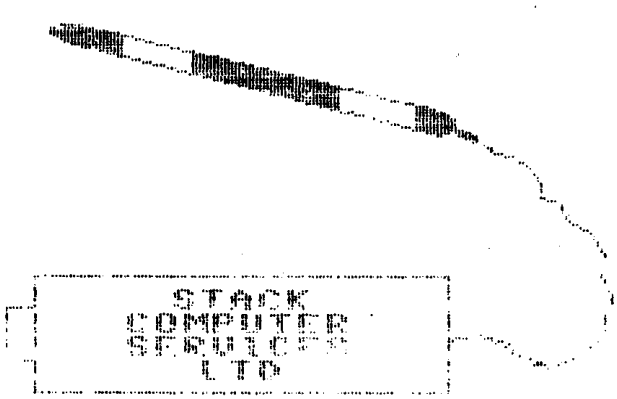
STACK COMPUTER SERVICES LTD

THE COMPLETE LIGHT PEN

CONTENTS

INTRODUCTION

The LIGHT PEN for the COMMODORE 64 consists of a light sensor cased in a pen. This is linked via a flexible lead to a connector which fits into the games port at the side of the 64. The Principal behind the LIGHT PEN is quite simple and is explained in the following pages.



Your STACK LIGHT PEN package should contain the following items. If there is anything missing, please return the complete package unused to your local dealer for a replacement.

PACKAGE CONTENTS

Stack Light Pen
Two cassette tapes
Warranty Registration Card

USING THIS MANUAL

This manual contains basic routines to help the novice, and it is suggested that the user makes certain that he understands each step before moving on to the next. There is a list of conventions used in this manual, hints on general maintenance, and remedies for some common problems in the appendix.

ONLY ROUTINES PRINTED IN THIS LIGHT PEN SECTION OF THE MANUAL MAY BE COPIED AND USED AS PART OF THE USER'S OWN CODE.

CONVENTIONS

Certain conventions have been used in the program listings in this manual and they are displayed here for reference. Any notation used will be shown by its being enclosed in square brackets: [...]

Mnemonic	Keys typed	Functions
cls	Shift/Clr	Clears screen
crh	Clr	Moves cursor to home position
crd	Crsr down	Moves cursor down one line
crw	Ctl/2	Turns colour under cursor white
rvn	Ctl/9	Turns reverse field mode on
rvf	Ctl/0	Turns reverse field mode off
sp	Space bar	Prints a space (sp*11 is 11 spaces)
l	Specified by user	Each 'l' represents one digit.

GENERAL MAINTENANCE

Your TV screen should be kept as clean as possible since dust or static accumulates on its surface and this will gradually clog or degrade the Light Pen. If this occurs cleaning the screen and blowing across the tip of the pen will restore the response.

Always store your Light Pen and cassettes away from strong magnetic fields.

INFORMATION

In order to provide the user with as much support as is practicable, we would appreciate it if any useful hints or comments could be forwarded in writing to:

DISTRIBUTED BY
MEEDMORE DISTRIBUTION
28 FARRIERS WAY INDUSTRIAL ESTATE
NETHERTON MERSEYSIDE
L30 4XL

051 521 2202

DISCLAIMER

Whilst every effort has been made to provide a flexible, reliable and above all low-cost product, STACK COMPUTER SERVICES LTD wish to point out that no claim is made for complete compatability with any other equipment or program. The information given is believed to be accurate but no liability can be accepted for the consequences of any error. Ours is a policy of continued development and we therefore reserve the right to alter the design or specifications without prior notice.

BOOK ONE
LIGHT PEN MANUAL

INDEX

TAPE INDEX

SIDE ONE	SIDE TWO
Crossword Twister	Shuffler
Life	Simon
Seek and Destroy	Othello
Draughts	Go-go
Concentration	Lost in the Labyrinth

LOADING INSTRUCTIONS

Insert tape into C2N and rewind to the start of the required side.

Plug Light Pen into PORT ONE;

If the desired game is either LIFE or LOST IN THE LABYRINTH type the following

LOAD "LIFE 1" or **LOAD "LOST 2"**

and press **«RETURN»**.
When either program has loaded type

LOAD

and press **«RETURN»**.
When the second part has loaded type

RUN

and press **«RETURN»**.

For any other game type

LOAD "(filename)"

and press **«RETURN»**.

Press the PLAY button on the C2N.

When the program has loaded type

RUN

and press **«RETURN»**.

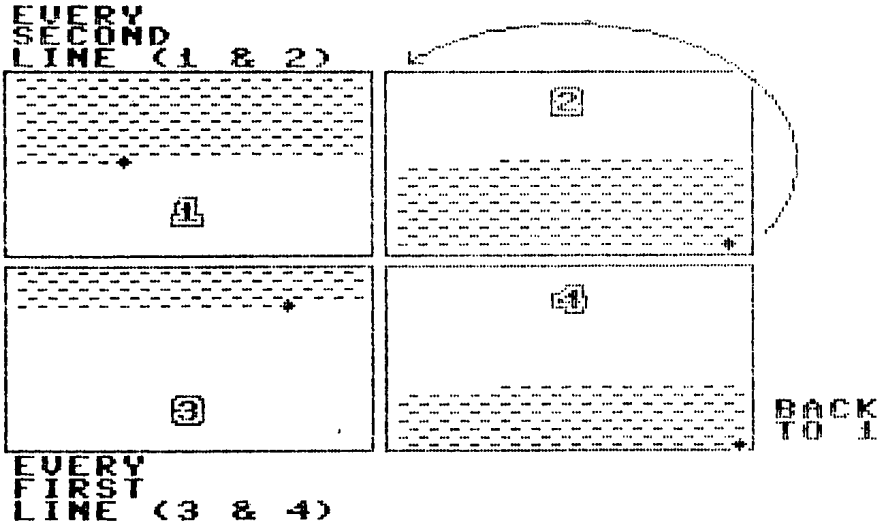
CHAPTER ONE

YOUR TV PICTURE

The picture on your television set is made up of the traces of light caused by a fast moving 'spot' of illumination. This spot begins in the top left corner of the screen and moves across to the right, then it flies back to the left and down a line, and so on until it reaches the bottom right of your screen when the process is repeated.

This 'spot' is the trace of light that the Light Pen detects.

INTERLACED DISPLAY



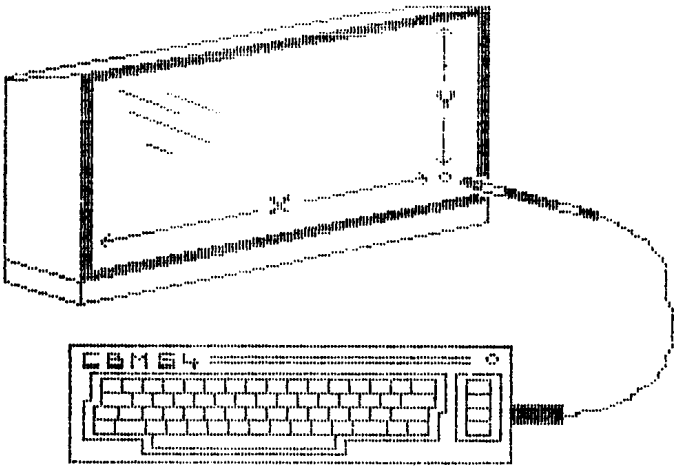
THE LIGHT PEN AND DISPLAY

In the end of the Light Pen is a light sensor which detects when the 'spot' is passing the tip of the pen and tells the 64 that is has 'seen' it. The 64 records the location of the spot in its VIC chip. This location is recorded in two registers in the form X, Y:

Reg. 53267 holds the X (horizontal) position,

Reg. 53268 holds the Y (vertical) position.

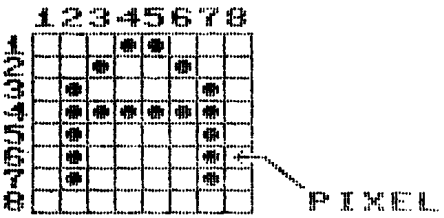
HOW THE LIGHT PEN SEES THE DISPLAY



LIGHT PEN SENSITIVITY

Each letter or number that can be displayed by your 64 on the TV screen consists of 64 points of light called pixels.

So each character looks like this:



This means that you can plot squares of character size (8 x 8 pixels) in Low-Resolution, or single pixels in High-Resolution. Stack PAINTBOX, included free in this package, enables the user to plot points, lines, circles, etc. in High-Resolution.

USE OF COLOURS

As you know the Light Pen produces a reading taken from the pulse of light that builds your TV picture. However, the sensitivity of the pen depends largely on the colours used. For example, BLACK and RED do not emit sufficient light for the pen. As a general rule use only the lighter colours.

CHAPTER TWO

WORKING AREA

The working area available on your TV screen is the BACKGROUND area within the BORDER. Type in and run the following short program:

```
10 POKE53280,0:POKE53281,1
20 PRINT PEEK(53267),PEEK(53268)
30 PRINT "[cls]":GOTO 20
```

Plug your Light Pen into PORT ONE, (i.e. the port farthest from the power switch), and move it around the screen. The numbers shown on the screen are the X and Y coordinates of the pen position on the screen.

However, the screen consists of 25 rows of 40 columns, so in order to obtain an accurate reading a slight adjustment is necessary.

Run your program again and point the Light Pen at the TOP LEFT corner of the BACKGROUND, (NOT THE BORDER).

When the numbers become steady or nearly steady, make a note of the number most commonly shown in each column

These numbers will be the OFFSETS required to suit your TV display, the left column being known as X9 and the right as Y9.

Now type in and run the following program, substituting the approximate values of X9 and Y9 for your own:

```
10 X9=35:Y9=51
20 POKE53280,0:POKE53281,1
30 X=INT((PEEK(53267)—X9)/4)
40 Y=INT((PEEK(53268)—Y9)/8)
50 PRINT "[cls]";X,Y
60 GOTO 30
```

Notice now that the X values range from 0-39, (i.e. 40 columns), and the Y values range from 0-24, (i.e. 25 rows).

(If your readings are different from these, adjust your values of X9 and Y9 until they give an accurate reading.)

Now that you have values for X9 and Y9. they can be used instead of the values used throughout this manual. Also, if you load and list your Stack Game, near the beginning of the listing you will find a line like this:

```
X9=35:Y9=51:REM PEN ALIGNMENT
```

Enter your values instead of 35 and 51 and then save your game with the new values permanently in your program.

HOLDING THE PEN

The most effective way of holding your Light Pen is the same way as holding any normal pen. For the touch contacts to be used property, the metal band nearer the lead should be touching the area of skin between your thumb and index finger, so that all you have to do to make a contact is touch the other metal band with the end of your index finger. If you are having difficulty with the contacts registering please see the appendix.

USING THE TOUCH CONTACT

A feature of the Light Pen is a touch contact which allows you to control the light sensor. Without this feature the Light Pen could not be used to take selective readings, as all readings would register in your programs, and the Pen would be uncontrollable.

The following short program demonstrates the use of the touch contacts:

```
10 PRINT "[cls]";
20 TC=PEEK(653)AND4
30 IF TC=0 THEN 20
40 PRINT "CONTACT MADE"
50 END
```

Line 20 reads the Touch Contact register, 653, and then performs a logical AND between the contents and 4; the result of this operation being stored in the variable TC.

Line 30 tests TC. If TC is equal to zero, then the contacts have not been touched. If TC is not equal to zero, then the contacts have been touched and the program will print 'contact made' on the screen.

Sometimes it is also desirable to pause your program until the Touch Contacts are released, i.e. to prevent repeating. The following lines do this:

```
10 TC=PEEK(653)AND4
20 IF TC=4 THEN 10
```

In the next chapter example programs are given with a line by line description of the role the Light Pen plays.

CHAPTER THREE

LOW-RESOLUTION PLOTTING

The following program is an example as to how the Stack Light Pen can be used for such applications as graph drawing and bar charting, or for drawing simple pictures. Each line is explained in detail for the novice:

10 X9=24:Y9=51:REM PEN ALIGNMENT

This line holds the offsets used throughout the program. For further explanation see the section WORKING AREA in chapter two.

```
20 BR=53280:BD=53281
30 SC=1024:CO=55296
40 PRINT "[cls]";
50 POKEBR,0:POKEBD,1
```

Line 20 uses 'BR' and 'BD' as constants referring to the BORDER and BACKGROUND registers respectively.

Line 30 uses 'SC' and 'CO' as constants referring to the start of screen memory and the start of colour memory respectively.

Line 40 clears the screen.

Line 50 turns the BORDER colour black and the BACKGROUND colour white.

For further explanation please consult your C64 Reference Guide.

```
60 TC=PEEK(653)AND4
70 IF TC=0 THEN 60
```

These two lines check the contacts on the pen to see if they have been touched.

Line 60 reads the contents of register 653 and then performs a logical AND between the contents and 4. Register 653 is where the information from the contacts, NOT the light sensor, is held. If the contacts have been touched, then the result of this operation will be 4; if not, the result will be 0.

Line 70 checks the result against the number 0. If the result from line 60 is 0, then the program jumps back to line 60. This will recur until the result from line 60 is 4. This time the check in line 70 will not be true, and so the program will go on to line 80.

For a detailed description of the logical operators, please consult your C64 Reference Guide.

80 X=PEEK(53267)—X9:Y=PEEK(53268)—Y9
90 X=INT(X/4):Y=INT(Y/8)

Lines 80 and 90 read the position of the pen on the screen and store it in the form X,Y. For an explanation of the registers used please see chapter one.

100 IF (X<0 OR X>39 OR Y<0 OR Y>24) THEN 150

This line checks both X and Y to make sure they are in the correct range.

110 PO=(Y*40)+X
120 SP=SC+PO:CP=CO+PO

Line 110 takes the Y reading, multiplies it by 40, and then adds the result to X to obtain the number of character positions from the top left of the screen to the position of the pen.

Line 120 calculates the actual screen position (SP) by adding the character position to the start of the screen memory, and also calculates the actual colour position in the same way.

130 POKESP,160
140 POKECP,3

Line 130 displays a square on the screen at the pen position, and line 140 colours the square cyan.

150 GOTO 60

Line 150 repeats the plot sequence from line 60 onwards.

HIGH-RESOLUTION PLOTTING

This program is an example of the high accuracy of the Stack Light Pen. The program allows you to plot single pixel points on a High Resolution, (Hi-Res), screen; the same basic theory you can see in your copy of Stack PAINTBOX.

10 X9=24:Y9=51:REM PEN ALIGNMENT

As used before, line 10 holds your offsets.

20 POKE53272,PEEK(53272)OR8

This line changes the internal screen memory pointer to point at the hires screen, which in this case starts at 8192.

For further details please consult your C64 Reference Guide.

30 POKE53265,PEEK(53265)OR32

This line turns on Hi-Res mode by setting bit 5 of register 53265.

40 GOSUB 170

Line 40 jumps to line 170 to clear the screen and then control returns to line 50.

```
50 FOR I=1024 TO 2023
60 POKEI,1
70 NEXT
```

Lines 50 to 70 change the background colour to white.

80 GETA$:IF (A$="C") THEN GOSUB 160

This line accepts a key from the keyboard. If the letter 'C' is pressed then control is passed to the CLEAR SCREEN routine: if any other key, or no key at all, is pressed, control passes on to line 90.

90 IF (PEEK(653)AND4)<>4 THEN 80

Line 90 tests the touch contact on the Light Pen. If no contact control is passed back to line 80.

```
100 X=INT((PEEK(53267)—X9)*2)
105 Y=INT((PEEK(53268)—Y9))
110 X2=INT(X/8):Y2=INT(Y/8)
```

Line 100 reads the X and Y registers and subtracts the offsets.

Line 110 calculates the character positions of X and Y, and stores them in the variables X2 and Y2.

```
120 PO=8192+(Y2*320)+(X2*8)+(YAND7)
130 BI=7—(XAND7)
```

Line 120 calculates the byte in which the pixel is to be set.

Line 130 calculates the bit which is to be set in that byte.

For a more detailed description please consult your C64 Reference Guide.

140 POKEPO,PEEK(PO)OR(2↑BI)

This line sets the correct pixel on the screen, the logical OR preventing this action clearing any other pixels set in the same byte.

150 GOTO 80

Line 150 transfers control back to line 80 thus repeating the above process.

```
160 POKE53280,5:REM GREEN BORDER
170 FOR I=8192 TO 16191
180 POKEI,0
190 NEXT
200 POKE53280,0:REM BLACK BORDER
210 RETURN
```

Lines 170 to 190 clear the hires screen by turning off all the pixels. Lines 160 and 200 change the border colour so that the user knows what is happening. Line 210 passes control from this routine to the command immediately after the one that called it.

LIGHT PEN CONTROLLED MENU

This program is the third and last example of Light Pen programming, and is included because it shows how a program may be completely Light Pen controlled.

```
10 X9=24:Y9=51:REM PEN ALIGNMENT
20 POKE53280,5:REM GREEN BORDER
30 POKE53281,6:REM BLUE BACKGROUND
```

The above lines have been explained in the previous chapters.

```
40 PRINT CHR$(14)
50 PRINT CHR$(8)
```

Line 40 turns on lower case mode, and line 50 locks that mode in so that it cannot be reversed via the Shift/Commodore keys.

```
60 PRINT "[cls crd crw sp*11]STACK[sp]C64[sp]Lightpen"
70 PRINT "[sp*18]MENU"
80 PRINT "[sp*18]===="
```

The above lines set out the headings for the menu. If in any doubt about the symbols used please see conventions.

```
90 PRINT "[crd sp*3](1)[sp*3]Option[sp*2]One[sp*15 rvn sp rvf]"
100 PRINT "[crd sp*3](2)[sp*3]Option[sp*2]Two[sp*15 rvn sp rvf]"
110 ……
……
……
180 PRINT "[crd sp*3](10)[sp*2]Option[sp*2]Ten[sp*15 rvn sp rvf crh]"
```

The above lines are part of the main screen of the program, and it is from this screen that the program is controlled, and so a full program listing has been included at the end of this section.

```
190 S=54272
200 POKES+5,9
210 POKES+6,0
220 POKES,193
230 POKES+1,44
240 POKES+24,15
```

Lines 190 to 240 set the sound effect up. In a menu driven program it is important to have both visual and audible messages so that the user knows precisely where he is at all times.

Line 190 holds the Sound Interface Device (SID) base address.

Line 200 sets the Attack and Decay of the envelope, where Attack=0 and Decay=9

Line 210 sets the Sustain and Release of the envelope, where Sustain=0 and Release=0

Line 220 sets the Low frequency of the note to be played, and line 230 sets the High frequency. The actual note is F-5.

Line 240 sets the volume to maximum by setting register S+24 with 15.

For further explanation of sound please consult your C64 Reference Guide.

250 GOSUB 2720

This line jumps to line 2720 to read the pen and test the contacts.

```
260 IF Y=5 THEN 380: REM OPTION ONE
270 IF Y=7 THEN III: REM TWO
280 IF Y=9 THEN III: REM THREE
290 IF Y=11 THEN III: REM FOUR
300 ……
……
……
350 IF Y=23 THEN IIII:REM TEN
360 GOTO 250
```

The above lines check the Y reading of the pen against the known Y coordinates of each of the option boxes. If any of the checks are true, control of the program will jump to the appropriate line. If none of the checks are true, line 360 will jump back to read the pen again.

```
370 REM ==== OPTION ONE CHOSEN ====
380 PRINT "[cls crd sp*14]Option[sp]One"
390 PRINT "[sp*14]=========="
400 PRINT "[crd]……"
410 PRINT "……"
420 ……
……
……
580 PRINT "……"
590 GOTO 2670
```

The program lines 370 to 590 show the standard layout of an option screen. There is a maximum of 20 lines of informative text, i.e. Text between the title and return box.

Line 590 jumps to the routine that prints the return box and tests the pen.

```
2660 REM ==== RETURN BOX ====
2670 PRINT "[crh crd*22]"
2680 PRINT "[sp*12]Return to Menu[sp rvn sp rvf crh]";
2690 GOSUB 2720
2700 GOTO 60
```

Lines 2660 to 2680 print the return box, and line 2690 jumps to the pen read routine to check the box. If the box has been pointed to by the pen and the contact touched, line 2700 will send the program back to line 60.

```
2710 REM ==== READ PEN POSITION ====
2720 POKES+4,32
2730 X=PEEK(53267)—X9:Y=PEEK(53268)—Y9
2740 X=INT(X/4):Y=INT(Y/8)
```

Line 2720 releases the sawtooth waveform for voice one.

Lines 2730 and 2740 read the X and Y positions of the pen and calculate its position in terms of character positions.

```
2750 PO=1024+X+(Y*40)
2760 IF PEEK(PO)<>160 THEN 2730
```

LINE 2750 calculates the pen position in screen terms, and then checks that position to see whether or not it is the return box, (160 is the character code of a reverse field space). If that position is not the box control passes back to line 2730 to read the pen again.

```
2770 POKEPO,32
2780 POKEPO,160
```

The above lines 'flash' the box by changing it to a space and then back to a reverse field space.

```
2790 IF (PEEK(653)AND4)<>4 THEN 2730
```

This line tests the touch contact. If there is no contact then control passes back to line 2730 to read the pen again.

```
2800 POKES+4,33
2810 RETURN
```

Line 2800 gates the sawtooth waveform for voice one, and line 2810 passes control back to the command immediately after the one that called the routine.

PROGRAM LISTING

```
10 X9=24:Y9=51:REM PEN ALIGNMENT
20 :
30 REM
40 REM         STACK C64 LIGHTPEN
50 REM
60 REM              EXAMPLE MENU
70 REM
80 REM
90 :
100 POKE53280,5:REM GREEN BORDER
110 POKE53281,6:REM BLACK BACKGROUND
120 :
130 PRINT CHR$(14):REM LOWER CASE ON
140 PRINT CHR$(8):REM LOCK LOWER CASE
150 :
160 PRINT "[cls crd sp*11]STACK[sp]C64[sp]Light Pen"
170 PRINT "[sp*18]MENU"
180 PRINT "[sp*18]===="
190 PRINT "[crd sp*3](1)[sp*3]Option[sp*2]One[sp*14 rvn sp rvf]"
200 PRINT "[crd sp*3](2)[sp*3]Option[sp*2]Two[sp*14 rvn sp rvf]"
210 PRINT "[crd sp*3](3)[sp*3]Option[sp*2]Three[sp*12 rvn sp rvf]"
220 PRINT "[crd sp*3](4)[sp*3]Option[sp*2]Four[sp*13 rvn sp rvf]"
230 PRINT "[crd sp*3](5)[sp*3]Option[sp*2]Five[sp*13 rvn sp rvf]"
240 PRINT "[crd sp*3](6)[sp*3]Option[sp*2]Six[sp*14 rvn sp rvf]"
250 PRINT "[crd sp*3](7)[sp*3]Option[sp*2]Seven[sp*12 rvn sp rvf]"
260 PRINT "[crd sp*3](8)[sp*3]Option[sp*2]Eight[sp*12 rvn sp rvf]"
270 PRINT "[crd sp*3](9)[sp*3]Option[sp*2]Nine[sp*13 rvn sp rvf]"
280 PRINT "[crd sp*3](10)[sp*2]Option[sp*2]Ten[sp*14 rvn sp rvf]"
290 :
300 S=54272:REM SID BASE ADDRESS
310 POKES+5,9:REM ATTACK=0 DECAY=9
320 POKES+6,0:REM SUSTAIN=0 RELEASE=0
330 POKES,193:REM NOTE LO BYTE
340 POKES+1,44:REM NOTE HI BYTE
350 POKES+24,15:REM VOLUME=15
360 :
370 GOSUB 1290:REM READ PEN POSITION
390 IF Y=5 THEN 510:REM OPTION ONE
400 IF Y=7 THEN 580:REM TWO
410 IF Y=9 THEN 650:REM THREE
420 IF Y=11 THEN 720:REM FOUR
430 IF Y=13 THEN 790:REM FIVE
440 IF Y=15 THEN 860:REM SIX
450 IF Y=17 THEN 930:REM SEVEN
460 IF Y=19 THEN 1000:REM EIGHT
470 IF Y=21 THEN 1070:REM NINE
480 IF Y=23 THEN 1140:REM TEN
```

```
490 GOTO 370
500 :
510 REM • OPTION ONE CHOSEN •
520 :
530 PRINT "[cls crd sp*14]Option[sp*2]One"
540 PRINT "[sp*14]==========="
550 :
560 GOTO 1210
570 :
580 REM • OPTION TWO CHOSEN •
590 :
600 PRINT "[cls crd sp*14]Option[sp*2]Two"
610 PRINT "[sp*14]==========="
620 :
630 GOTO 1210
640 :
650 REM • OPTION THREE CHOSEN •
660 :
670 PRINT "[cls crd sp*13]Option[sp*2]Three"
680 PRINT "[sp*13]============="
690 :
700 GOTO 1210
710 :
720 REM • OPTION FOUR CHOSEN •
730 :
740 PRINT "[cls crd sp*14]Option[sp*2]Four"
750 PRINT "[sp*14]============"
760 :
770 GOTO 1210
780 :
790 REM • OPTION FIVE CHOSEN •
800 :
810 PRINT "[cls crd sp*14]Option[sp*2]Five"
820 PRINT "[sp*14]============"
830 :
840 GOTO 1210
850 :
860 REM • OPTION SIX CHOSEN •
870 :
880 PRINT "[cls crd sp*14]Option[sp*2]Six"
890 PRINT "[sp*14]==========="
900 :
910 GOTO 1210
920 :
930 REM • OPTION SEVEN CHOSEN •
940 :
950 PRINT "[cls crd sp*13]Option[sp*2]Seven"
960 PRINT "[sp*13]============="
970 :
```

```
980 GOTO 1210
990 :
1000 REM • OPTION EIGHT CHOSEN •
1010 :
1020 PRINT "[cls crd sp*13]Option[sp*2]Eight"
1030 PRINT "[sp*13]============"
1040:
1050 GOTO 1210
1060 :
1070 REM • OPTION NINE CHOSEN •
1080 :
1090 PRINT "[cls crd sp*14]Option[sp*2]Nine"
1100 PRINT "[sp*14]==========="
1110 :
1120 GOTO 1210
1130 :
1140 REM • OPTION TEN CHOSEN •
1150 :
1160 PRINT "[cls crd sp*14]Option[sp*2]Ten"
1170 PRINT "[sp*14]==========="
1180 :
1190 REM • RETURN BOX •
1200:
1210 PRINT "[crh crd*22]"
1220 PRINT "[sp*12]Return[sp]to[sp]MENU[sp rvn sp rvf crh]";
1230 GOSUB 1290
1240 :
1250 GOTO 160
1260 :
1270 REM • READ PEN POSITION •
1280 :
1290 POKES+4,32:REM GATE BIT OFF
1300 :
1310 X=PEEK(53267)—X9:Y=PEEK(53268)—Y9
1320 X=INT(X/4):REM X AND Y
1330 Y=INT(Y/8):REM COORDINATES
1340 :
1350 PO=1024+X+Y*40:REM SCREEN POSITION
1360 :
1370 IF PEEK(PO)<>160 THEN 1310
1380 POKEPO,32:REM FLASH
1390 POKEPO,160:REM BOX
1400 :
1410 IF (PEEK(653)AND4)<>4 THEN 1310
1420 :
1430 POKES+4,33:REM GATE BIT SET
1440 :
1450 RETURN
```

END OF LISTING.

APPENDIX

COMMON ERRORS

When programming with a Light Pen for the first time, some errors are bound to occur. This section hopes to deal with the most common errors.

Fault	Cause	Remedy
Pen not functioning at all.	Pen in wrong port.	Plug pen into port one.
Pen giving incorrect readings.	(1): Offsets incorrect.	See chapter two. section one, 'Working Area'.
	(2): Colours too dark.	Use lighter colours in your program.
	(3): Brightness or colour controls not high enough.	Adjust TV controls to enable pen to see colours better.
	(4): Light from an external source falling on the screen.	Reposition TV in relation to other light sources.
	(5): Dirty pen or screen, or static on screen.	Clean both pen and screen.
Touch Contact not working.	(1): Pen being incorrectly held.	See the section, Holding The Pen
	(2): Hands too dry, good bridge not being made between contacts.	Moisten hands slightly.

BOOK TWO
STACK PAINTBOX

INDEX

COPYRIGHT

STACK PAINTBOX PACKAGE

CONCEPT

STACK PAINTBOX is a Light Pen based development system which fully exploits the graphics capabilities of your Commodore 64. The program has been written entirely in the machine code and operates around the High Resolution Bit Map Mode, giving 320 x 200 pixel screens.

The program has been designed to enable the user to draw, load, save, print and manipulate monochrome pictures. A full explanation of the features available is given in a later section.

SYSTEM REQUIREMENTS

STACK PAINTBOX REQUIRES:

Commodore 64;

1541 Single Disk Drive Unit, or
C2N Cassette Recorder;

Stack Light Pen;

Suitable Printer:

Commodore:

MPS 801,
MPS 802,
1515,
1525, or

Epson printer:

MX range,
FX range, or
RX range.

Any other printer type that has a Paintbox Printer Driver written for it.
If you wish to use a driver it must be loaded into Paintbox before any attempt at printing is made.

NB:

Any Epson type printer fitted with a STACK VEI HARDWARE INTERFACE will act exactly the same as a Commodore printer.

TAPE INDEX

SIDE ONE	SIDE TWO
Stack PAINTBOX:	PB.SCHEMATIC
(0 to 123).	(0 to 68).
Loading time: 4 mins. 25 secs.	
MESSERSCHMITT:	PB.HPLANS
(123 to 173).	(68 to 123).
	RALLY CAR
	(123 to 173).

LOADING INSTRUCTIONS

Insert PAINTBOX tape into C2N;

Rewind tape fully;

Hold down the SHIFT and RUN/STOP keys simultaneously;

Press the PLAY button on the C2N.

When PAINTBOX has loaded, it will run automatically.

SCREENS

In the PAINTBOX program there are two distinct high resolution screens that can be used for drawing. All PAINTBOX features apply to both screens as do all commands.

PAINTBOX FEATURES

STACK PAINTBOX has facilities for the following features:

AVERAGE
BROAD NIB
CIRCLE
CLEAR
COPY
EXCHANGE
FILL
FINE NIB
GET
INVERT
LINE
LOCK X
LOCK Y
MERGE
OUTPUT
PLOT
POINTS
PUT
QUADRILATERAL
READ
TEXT
UNPLOT
WRITE
ZOOM

COMMANDS

X Exit. This command will provide the correct exit to the Basic Operating System. In case of accidental exit type SYS 22272.

C Clear. Clears current Hi-Res screen and restores defaults.

H Help. Displays Help Screen.

E Exchange. Swaps screens. This key is the only one that repeats.

I Invert. Resets all SET points and sets all RESET points on current screen.

T Text. With this command text can be entered at the pen position. To exit text mode type F1.
Text will normally overwrite whatever is on the screen, but it may be OR'ed or XOR'ed with the screen contents by using F5 or F7 respectively.
Upper case, lower case and ASCII are available, but no graphics. On entering text mode, both cursor and marker are disabled, but both are reset on leaving.

Z Zoom. Set marker and type Z. Marker is the top left area of the zoom screen.
P sets PLOT mode;
U sets, UNPLOT mode.

Q Quadrilateral. Set marker and pen to opposite corners and type Q.

← Set crosshair point with marker.

↑ Connects marker to cursor, and updates marker, with a straight line.

* Circle. Draws circle with the marker being the centre, and the distance between the marker and the cursor being the radius.

P DEFAULT: Sets PLOT mode.

U Sets UNPLOT mode.

F DEFAULT: Sets FINE nib.
Can only be set from POINTS mode.

B		Sets BROAD nib. Can only be set from POINTS mode.
1	DEFAULT:	Sets LINES mode.
2		Sets POINTS mode.
+	DEFAULT:	Pen averaging on.
–		Pen averaging off.

= Copy. This command will copy the contents of the current screen to the other screen, and will leave the current screen unchanged.

/ Merge. This command will merge screens 1 and 2 to screen 1.

F3 Advance BACKGROUND one colour.

F5 Advance FOREGROUND one colour.

F7 Resets BACKGROUND and FOREGROUND colours to Black on White.

@ Catalog. Get disk contents.

£ Status. Get disk status.

R Read. Loads saved picture onto current screen.

W Write. Saves current screen.

O Output. Dumps current screen to printer.

< Get. Set the marker to the top left corner of the area you wish to get, point the pen at the bottom right and type '<'. With this command both the marker and the pen are rounded to the nearest character position. Please note that the maximum length of area you can get is 11 characters.

> Put. Point the pen at the top left corner of the area you wish to place your symbol in and type 'Y. Like get, the pen position will be rounded to the nearest character position. Please note that any symbol that has been got will stay in the PUT buffer until the buffer is cleared. This can be cleared by getting another symbol, thus replacing the old

one, trying an illegal GET, or by trying to fill a degenerate picture — see Fill below.

8 Lock X. This command makes straight line drawing easy. Typing 8 locks the X coordinate meaning that the cursor will always stay on the same vertical line. To unlock type 8 again.

9 Lock Y. Same as above but for the Y coordinate. To unlock type 9 again. Please note that BOTH X and Y may be locked at the same time.

0 Fill. This command is used to fill any completely enclosed area with black, a completely enclosed area being defined as the area surrounded by a continuous unbroken line. This area may have other totally enclosed areas within it, and these will not be filled.
The fill process can be halted at any time by pressing the RUN/STOP key.

WARNING
If an area is too complex to fill, i.e: More than 1109 corners, the program will fill as much as it can and then stop. However, if there are more than 769 corners but less than 1110, the fill will work completely but the symbol in the PUT buffer will be overwritten.

DISK AND TAPE OPERATIONS

STACK PAINTBOX has facilities for loading or saving pictures you have drawn to either disk or tape. If you have a picture you wish to load type 'R' for read. The screen will change and a prompt for the device you wish to load from will appear. Type either 'D' for disk, or 'T' for tape. Another prompt will not appear for the filename; type it in and press «RETURN». The picture will load onto the current screen.

To save a picture type 'W' for Write. As with Read, a prompt for disk or tape will appear. After making your choice the filename prompt will appear. Typing in the filename and pressing «RETURN» will save the current screen.

As a method of verification it is recommended that the following procedure is used:

Copy your picture to screen two by pressing the '=' key. Clear the current screen by pressing the 'C' key.

Type 'R' for read and load the picture just saved back to the current screen.

When using a disk drive two additional commands are available:

'@' will display the directory of the disk, and

'£' will display the current disk status. **WARNING: THE SYSTEM WILL CRASH IF THERE IS NO DISK DRIVE ATTACHED.**

PRINTER OPERATIONS

STACK PAINTBOX has facilities for reproducing screen drawn pictures as printed output, the command being 'O'.

Typing this command will change the screen and you will be asked which printed driver you are using, i.e: Epson, Commodore or a custom Driver, selected by type 'E', 'C' or 'P' respectively.

To indicate printing the border will become green and will become black again when the printing process has finished.

If you have selected the Epson printer option there is no way to stop the printing process once it has begun until the whole picture has been printed. With the Commodore option the printing may be terminated by pressing the RUN/STOP key.

WARNING: If Commodore is selected but there is no printer present the border will be green for approximately 20 seconds and will then return to normal black, BUT IF EPSON IS SELECTED AND NO PRINTER IS PRESENT THE SYSTEM WILL CRASH.

TEMPLATES

Included in your PAINTBOX package are two templates: an Electronic component template and a House Planning template.

To load either of these type:

'R' for READ,
'T' for TAPE

and either

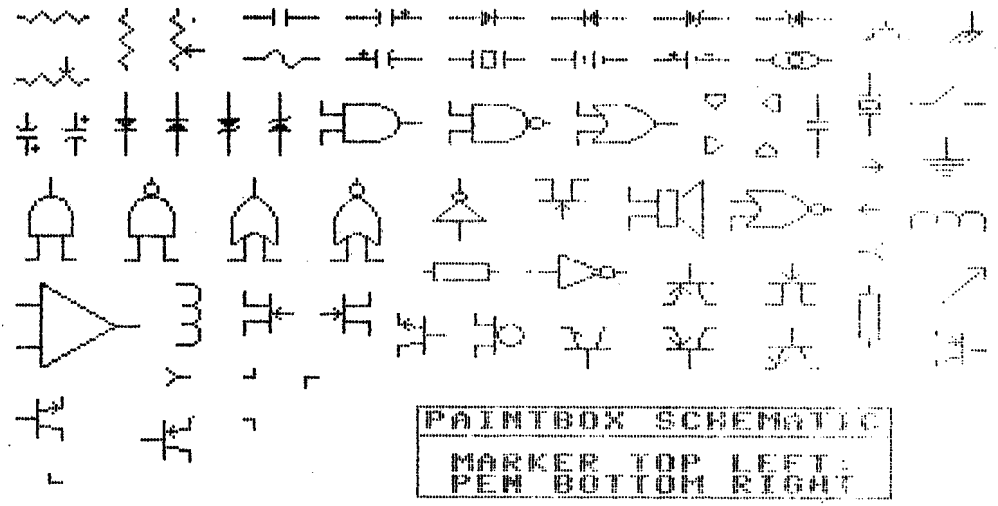
'PB.SCHEMATIC'

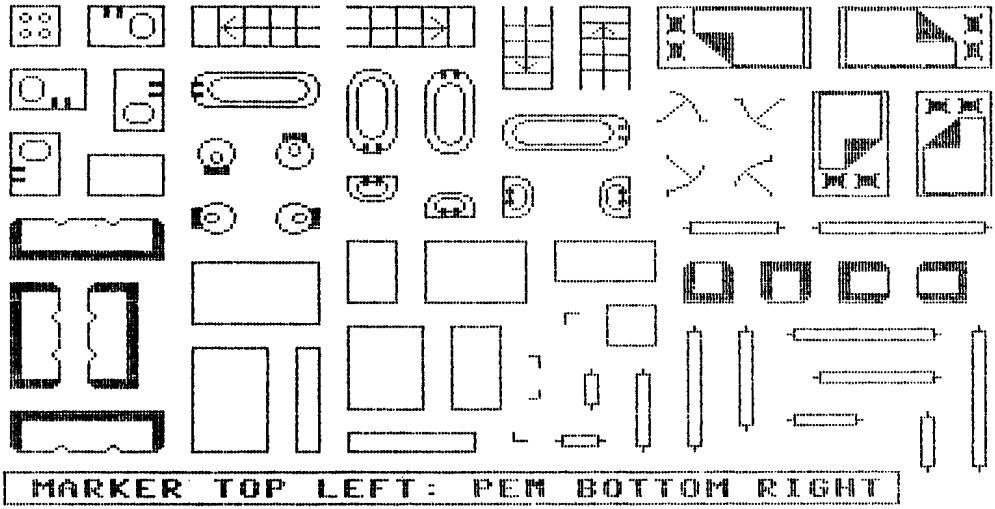
or

'PB.HPLANS'

These templates can be used to construct detailed drawings at designer level using the PUT and GET commands. Simply choose the desired symbol and GET it, swap screens and PUT the symbol in the required position thus building up a complex design quickly and easily.

Please see the COMMANDS section for instructions on PUT and GET.




MARKER TOP LEFT: PEM BOTTOM RIGHT

HINTS AND TIPS

This section of the manual is intended to be a guide for the first time user, enabling him to use Stack PAINTBOX to the full.

SELECTIVE ERASE:

If the user has a filled area and he wishes to erase part of it but not all then the procedure is as follows. Invert the screen and then draw the outline of the area to be erased, fill that area and then invert the screen again. The area will now have been erased.

ZOOM TRACKING:

When in Zoom mode the pen will track better if it is used moving from right to left, rather than left to right, and top to bottom, rather than bottom upwards, due to the way the screen is scanned.

SHADING:

If the user wishes to shade an area rather than totally fill it, i.e. for graphs etc., then a small area of the desired shading should be drawn. Then simply follow the GET and PUT procedures to shade the area.

HOLES IN FILL:

Before filling an area it is recommended that the user changes the background and foreground colours, eg. to white on red, because black on white may not show up a single pixel missing on a line.

DEMONSTRATION PICTURES

Included in your PAINTBOX package are two demonstration pictures which clearly show the versatility of the program. To load a picture type:

'F1' to start,

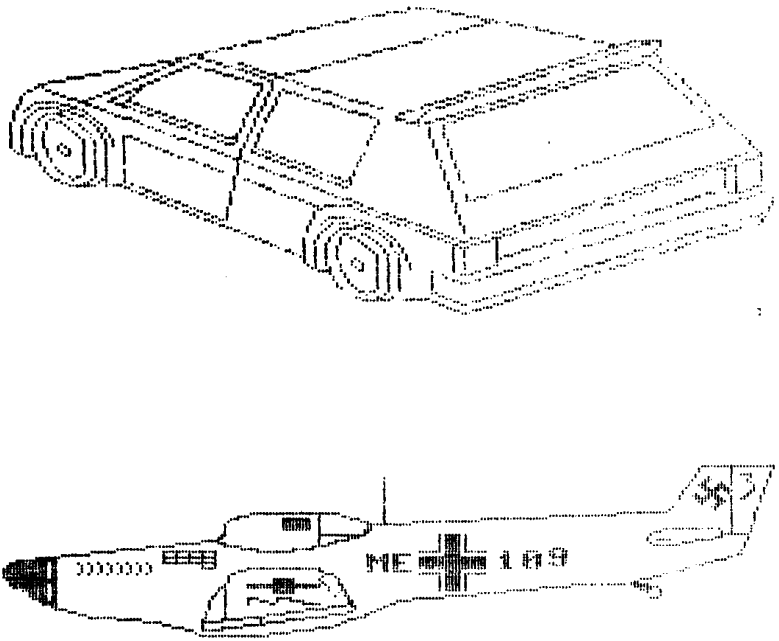
'R' for READ,

'T' for TAPE and either

'RALLY CAR' or

'MESSERSCMITT' for the filename.

All drawings and diagrams in this manual and package have been drawn using STACK PAINTBOX.



BACKUP

BACKUP PROCEDURE

LIGHTPEN GAMES

CROSSWORD TWISTER
SEEK AND DESTROY
DRAUGHTS
CONCENTRATION
SIMON
OTHELLO
GO-GO
SHUFFLER

For all the above games just follow the procedure below:

Rewind the tape to the beginning of the appropriate side, type

LOAD"«filename»"

and press **«RETURN»**.
When the program has loaded type

SAVE"«filename»",8

and press **«RETURN»**.

ALL THE FOLLOWING PROGRAMS REQUIRE A MACHINE CODE MONITOR

LIFE

This game is in two parts. Type

LOAD"LIFE 1",1,1

and press **«RETURN»**.
When the program has loaded enter your monitor and type

.S "LIFE 1",08,C000,C160

and press **«RETURN»**.
Now type

LOAD"LIFE 2"

and press **«RETURN»**.
When this part has loaded type

SAVE"LIFE 2",8

and press«**RETURN**».
PLEASE NOTE that you will have to load both parts from disk to play:

LOAD"LIFE 1",8,1

and press«**RETURN**».
Then type

LOAD"LIFE 2",8

and press«**RETURN**».
When ready type

RUN

and press«**RETURN**».

LOST IN THE LABRYNTH

This game is also in two parts. Type

LOAD"LOST 2",1,1

and press«**RETURN**».
When the program has loaded enter your monitor and type

.S "LOST 2",08,C000,C790

and press«**RETURN**».
Now type

LOAD"LOST 1"

and press«**RETURN**».
When this part has loaded type

SAVE"LOST 1",8

and press«**RETURN**».
As with LIFE, both parts have to be loaded to play.
See above.

STACK PAINTBOX

Paintbox is in six parts:

Basic Loader. This program will have to be re-written for disk.
Please type in the following;

```
10A=A+1:IFA=1THENLOAD"P2",8,1
20IFA=2THEN LOAD"P3",8,1
30IFA=3THEN LOAD"P4",8,1
40IFA=4THEN LOAD"P5",8,1
50PRINT"[cls crd*2]LOAD";CHR$(34);"P1";CHR$(34);",8"
60PRINT"[crd*4]RUN[crh]";
90POKE631,13:POKE632,13:POKE198,2
```

When you have finished type

SAVE"PAINTBOX",8

and press«**RETURN**».

PART 2. Type

LOAD"P2",1,1

and when it has loaded enter your monitor and type

.S "P2",08,1200,1C00

and press«**RETURN**».

PART 3. Type

LOAD"P3",1,1

and when it has loaded enter your monitor and type

.S "P3",08,4000,57F0

and press«**RETURN**».

PART 4. type

LOAD"P4",1,1

and when it has loaded enter your monitor and type

.S "P4",08,5800,5E30

and press«**RETURN**».

PART 5. Type

LOAD "P5",1,1

and when it has loaded enter your monitor and type

.S "P5",08,47A1,4D40

and press«**RETURN**».

PART 1. Type

LOAD"P1",1,1

and when it has loaded enter your monitor and type

.S "P1",08,0801,11F0

and press«**RETURN**».

To backup any templates or pictures; load PAINTBOX, load the picture and then use the save to disk option.

BIBLIOGRAPHY

COMMODORE 64:

USER'S GUIDE

COMMODORE 64:

PROGRAMMERS REFERENCE GUIDE

Both published by:
Commodore Business Machines Incorporated,
and
Howard W. Sams & Co. Incorporated.