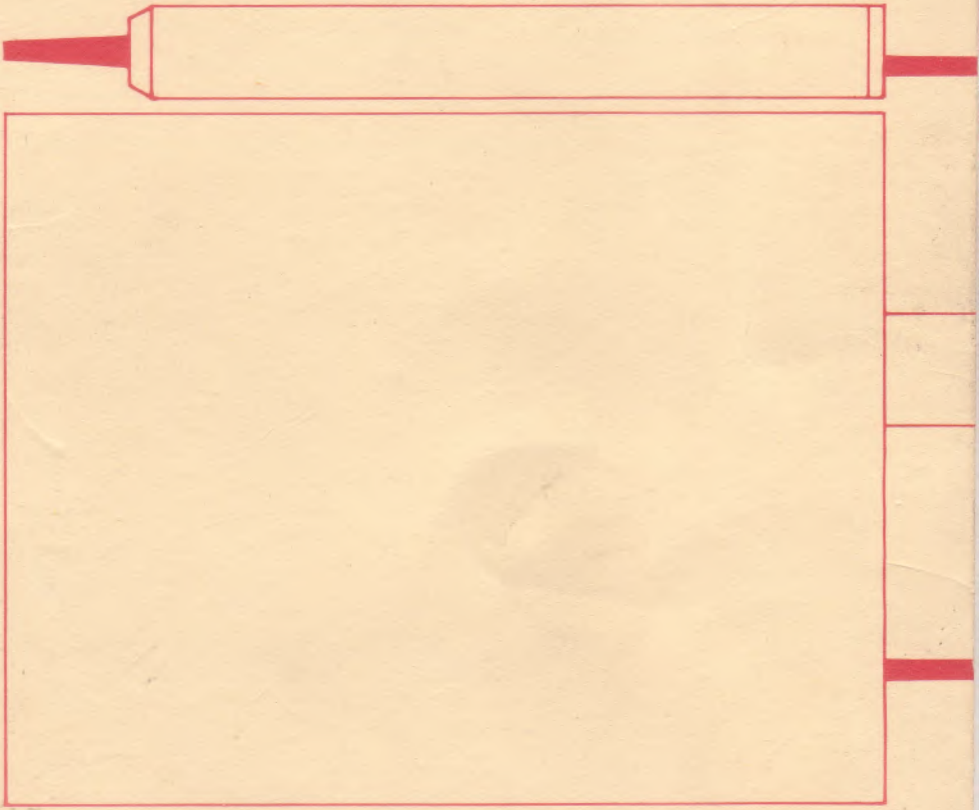
RH Electronics Sales Limited

Light Pen for the BBC Microcomputer

USER'S MANUAL



Outstanding features of the R.H. Electronics Sales Ltd. Light Pen

- SOFTWARE CONTROL, via 6 bit control word, of the operational mode of Pen hardware.
- GOOD DESIGN of Pen Probe. Light, small and very strong with a narrow tip allowing clear visibility around operating area.
- PUSH TIP operation of switch to signal user decisions to computer.
- STATUS INDICATOR LED mounted at rear of Pen.
- SIMPLE TO INSTALL Just plug into Analogue connector.
- NARROW FIELD OF VIEW due to light guide optics.

R H Electronics Lightpen Manual

INTRODUCTION

There are a number of devices which enable the user to interact with a computer. The standard ASCII keyboard is the most common of these devices with Light Pens probably a close second. Commercial Light Pens were developed shortly after the introduction of video displays, but until very recently were found only on expensive and specialised systems. The early Light Pens were designed to work with special displays, optimised for the purpose (vector rather than raster scan displays, sometimes doped with infra-red phosphors). A Light Pen allows manipulation of information on a video display. Not to be confused with a bar code reader.

The R.H. Electronics Light Pen is not designed to read bar code. It interacts directly with information represented by illuminated areas on a video screen, ignoring ambient lighting conditions. It is designed to give high performance with raster scan displays (such as domestic colour or black and white T.V.s and high resolution monitors) used in conjunction with the B.B.C. Microcomputer.

The Pen generates a video strobe pulse (VSP) corresponding to its' position on the screen (providing there are illuminated pixels at that location). On the rising edge of the VSP the current character address +2 is latched into the Light Pen register of the 6945 CRTC (Cathode Ray Tube Controller Chip). Due to the internal architecture of the 6845 there is a period (7% of system time, at the most) during the change of character address, when erroneous data may be latched into the Light Pen register. This is a characteristic of the CRTC in the micro, not of the R.H. Light Pen. As the clock for the data latched into the Light Pen register is the character clock, the maximum definition of the Pen is limited to one character position.

The R.H. Electronics Light Pen is unique in that various operating modes may be selected by the driver program, allowing the programmer to choose the best operating mode for any application.

The software supplied on the Light Pen cassette consists of a MACHINE CODE driver routine and several BASIC demonstration programs. Further software is available and the range is being continually extended.

Setting Up

When the Light Pen and Interface have been removed from the packaging, ensure the B.B.C. Microcomputer is SWITCHED OFF and plug in the D-type connector, attached to the ribbon cable, into the Analogue socket on the back of the Computer.

1) Switch the Computer back on again.
2) Put the Light Pen Cassette into your cassette recorder and check that the tape is fully rewound
3) Ensure that PAGE=&E00
4) Type CHAIN"" [ret]
5) Press PLAY on the recorder, follow the instructions on the screen and wait for the programs to load.
6) Once 'DRIVER', 'RELOC' and 'LPDRIV' have been executed the LIGHTPEN driving software will be in place and PAGE will have been set to &1300.
7) All the programs on Side 1 can now he CHAINed in the normal manner.
8) To move from one program to another on Side 1 press [Escape] and CHAIN "program title" [ret].
9) To use DRAW (side 2) you do not need to go through steps 3 to 8

NOTE: Don't forget to switch off the recorder when not loading.

Programs on the Light Pen Cassette

The programs on the cassette are in the following order:-

Side 1

DRIVER	(instructions)
RELOC	(machine code loader and relocator)
LPDRIV	(light pen machine code Driver)
TRIM	(Auto-trim in any mode)
SETUP	(Align your lightpen with your display)
DRAFTS	(Use your 'pen with this classic game)
KWARTET	(Another game for you to study and use)

Side 2

DRAW	(free-hand drawing in Mode 0)

(The program called D.AC is part of DRAW)

SWAN	(A Mode 2 picture using various programs from our range, drawn by a member of staff).

Driver

This program asks the user where in memory he/she wishes the Light Pen Driver to reside. When the Driver has been loaded in, PAGE will be set above this to protect it from being over-written. Normally the Driver will be loaded in at &E00 for cassette users or &1900 for disk users.

Reloc

The Driver must be able to reside anywhere in memory, and as relocatable written machine code is very inefficient, this program is necessary to relocate the Machine code to the address specified in the Driver program.

Lpdriv

This is the actual Driver machine code and is automatically loaded into the address specified in the Driver program.

Setup

The Setup Program is included so that you can set the contrast and brightness of your television to allow the Light Pen to detect the video pulses correctly.

Trim

The Trim Program is used to adjust the offset in software, which calculates the point detected on the screen with respect to the physical position of the Pen tip. In practice this is altered one character position to the left or to the right. Some programs written for the Light Pen have a procedure at the beginning of them to set the trim (See 'Draw').

Using the Light Pen Driver

On loading the Driver a BASIC program will already be in memory consisting of many Functions and Procedures which allow the Light Pen to be used easily from a BASIC program. The utilities available are:-

PROC_LPINIT	Initialise Light Pen interrupts
PROC_LPEXIT	Disable Light Pen interrupts
FN_POINT	X Byte Characters from last UPDATE
FN_YPOINT	Y Character from last UPDATE
FN_XTEXT	X Character from last UPDATE
FN_XGRID	X Grid from last UPDATE
FN_YGRID	Y Grid from last UPDATE
PROC_XDIV(n)	Divide screen into Xgrid of 0 - (n-1)
PROC_YDIV(n)	Divide screen into Ygrid of 0 - (n-1)
FN_SWITCH	Switch status (0-Released, 1-Pressed)

FN_UPDATE	State of Update latch
PROC_RESET	Reset UPDATE indicator
PROC_SAVE(n$)	Save light pen program
PROC_TRIM(n)	Set trim (-3 to +3)
PROC_LPMODE(n)	Select Light Pen mode n (0-7)

The Light Pen Driver runs under continuous interrupts which means that 50 times a second the Driver is called. Upon each of these calls the Driver checks to see if a video pulse has occured. If it has and the Driver is in the right mode, the Driver will perform an UPDATE which means it will read the Light Pen Register in the 6845 inside the B.B.C. Microcomputer, and use this to ' update' the various registers of the Light Pen Driver. The utilities are now described in more detail with basic examples of their use:-

PROC_LPINIT

This initialises and enables the Light Pen Driver and interrupts. This procedure MUST be called to initialise the 'pen or after a Mode change.

Example: PROC_LPINIT

PROC_LPEXIT

This disables the Light Pen Driver, completely disabling the interrupts which the Driver uses. This should be done whenever the Driver is no longer required.

Example: PROC_LPEXIT

FN_XPOINT

This contains the position of the Pen on an X axis, on a scale of 0 (far left) to n-1 (far right) where n is equal to the number of horizontal bytes which make up the screen in the current screen mode.

```
Examples: X=FN_XPOINT
          PRINT FN_XPOINT
```

FN_YPOINT

This contains the position of the Pen on a Y axis scale of 0 (top) to n-1 (bottom), where n is the number of vertical characters in the current screen mode.

```
Examples: Y=FN_YPOINT
          PRINT FN_YPOINT
```

FN XTEXT

This contains the position of the Pen on an X axis scale of 0 (far left) to n-1 (far right), where n is the number of horizontal characters in the current screen mode

```
Examples: XT=FN_XTEXT
          PRINT FN_XTEXT
```

FN XGRID

This contains the position of the pen on an X axis scale of 0 to n-1, where n is the number of horizontal squares the screen has been divided into (XDIV). Any overflow caused by a remainder after the division will occupy the right side of the screen and return &FF. For a more detailed explanation of this see PROC_XDIV(n) and PROC YDIV(n).

```
Examples: XG=FN_XGRID
          PRINT FN_XGRID
```

FN_YGRID

This contains the position of the pen on a Y axis scale of 0 to n-1, where n is the number of vertical squares the screen has been divided into (YDIV). Any overflow caused by a remainder after the division will occupy the lower area of the screen and return &FF. For a more detailed explanation of this see PROC_XDIV(n) and PROC YDIV(n).

```
Examples: YG=FN_YGRID
          PRINT FN_YGRID
```

PROC_XDIV(n)

The screen can be divided into a grid and the Pen position represented by a set of co-ordinates x and y, where x will range from 0 to the number of x grid squares-1, and y will range from 0 to the number of y grid squares-1.
PROC_XDIV(n) is used to set up the number of horizontal grid squares the screen is to bedivided into, and FN XGRID can be used to find the horizontal grid square position of the Pen.

```
Examples: PROC_XDIV(3)
          PROC_XDIV(XD)
```

PROC_YDIV(n)

This is used to set the number of vertical grid squares the screen is to be divided into, and FN_YGRID can be used to find the vertical grid square position of the Pen.

```
Examples: PROC_YDIV(7)
          PROC_YDIV(YD)
```

FN_SWITCH

This contains the current state of the Light Pen Switch and is updated 50 times a second when the Light Pen Interrupt takes place. A value of 1 is returned for SWITCH CLOSED and a value of 0 for SWITCH OPEN.

```
Examples: SW-FN_SWITCH
          PRINT FN_SWITCH
```

FN_UPDATE

This contains the Pen Driver Update state. When a video pulse is detected by the Pen, the 6845 CRTC chip in the B.B.C. Microcomputer latches the Light Pen Position in one of its' registers. The Driver detects that this has happened and may 'update' the values of FN_XPOINT, FN_XTEXT etc, depending on the Light Pen Mode selected (see PROC LP_MODE(n)).If this 'update' has taken place FN_UPDATE will return a value of 1, if not it will return value of 0. This would prove to be of use in an application where a program is waiting for the Pen to be putto the screen, to find the position to which a line has to be drawn.

```
Example: 10 REPEAT UNTIL FN_UPDATE
         20 PRINT FN_XPOINT,FN_YPOINT
         30 PROC_RESET
         40 GOTO 10
```

(This program will wait until the Pen has been placed against the screen before it prints the position of the Pen. See PROC RESET for further information).

PROC_RESET

This is used to reset FN_UPDATE after it has been set so that another video pulse can set it and therefore update FN XPOINT, FN_YPOINT etc.

PROC_SAVE(n$)

The Light Pen Driver is stored below PAGE therefore the normal SAVE command will only save your basic program, not the Driver. This is fine if you are prepared to load the Driver every time you wish to continue with your program, but when the program is finished you will want to be able to CHAIN it to automatically run. PROC_SAVE(n$) will allow you to do this because it will save the Driver along with your basic program. You may ask how the program loads and runs with CHAIN when the Driver is at the beginning of the basic program. This is dealt with in a small part of basic at the beginning of the Driver, which on running sets PAGE up to where your program starts, and runs it.

PROC_TRIM(n)

This allows the Light Pen Trim to be set up to adjust the offset in software, which calculates the point detected on the screen with respect to the physical position of the Pen. The trim setting can be between -3 and +3. A setting of plus or minus one is usually sufficient. Note that the trim necessary may vary very slightly from mode to mode.

PROC_LPMODE(n)

This is the mode in which the Light Pen will operate. The Pen can be programmed so that it gives only one video pulse per switch press, one video pulse per switch release or continuous video pulses chile the switch is pressed or various other modes. The modes available are listed and described below.

An UPDATE will take place:-
n=0: once with each switch press and a video pulse
n=1: once with each switch release and a video pulse
n=2: continuously with the switch pressed and video pulses
n=3: continuously with the switch released and video pulses
n=4: continuously with video pulses
n=5: if latch reset and a video pulse
n=6: if switch pressed and latch reset and video pulses
n=7: if switch released and latch reset and video pulses
The term 'UPDATE' above means the process of updating

the Light Pen X and Y registers (FN XPOINT, FN YPOINT etc.).

Mode 0:

This mode will only UPDATE with the first video pulse received after the switch has been pressed. Repeated sequences of one UPDATE per switch press may he obtained.

Mode 1:

This mode will only UPDATE with the first video pulse received after the switch has been released. Repeated sequences of one UPDATE per switch release may be obtained.

Mode 2:

UPDATEs will take place continuously with every video pulse received when the switch is pressed.

Mode 3:

UPDATEs will take place continuously with every video pulse received when the switch is released.

Mode 4.

UPDATEs will take place continuously with every video pulse received.

Mode 5:

An UPDATE will take place if a video pulse is received and the UPDATE latch is reset. This UPDATE will, as all UPDATEs do, cause the UPDATE latch to be set and therefore stop further UPDATEs until the UPDATE latch is reset. This mode is useful in applications where only one UPDATE must take place until the UPDATE latch is reset by the user.

Mode 6:

An UPDATE will take place if a video pulse is received when the switch is pressed and the UPDATE latch is reset. This UPDATE will cause the UPDATE latch to be set, and therfore stop further UPDATEs until the UPDATE latch is reset. This mode is useful in applications where only one UPDATE must take place until the UPDATE latch is reset by the user.

Mode 7:

An UPDATE will take place if a video pulse is received when the switch is released and the UPDATE latch is reset. This UPDATE will cause the UPDATE latch to be set, and therefore stop further UPDATEs until the UPDATE latch is reset. This mode is useful in applications where only one UPDATE must take place until the UPDATE latch is reset by the user.

TECHNICAL INFORMATION

The Machine Code Driver runs under interrupt control and once initialised it will update most of the output table anytime the Pen sees a video pulse.

If the user does not wish to use the Procedures and Functions supplied on loading the Driver, these can be simply deleted by 'NEW'. The user can then implement the various functions by copying the relevant Procedure or Function, which will increase the speed of the program very slightly. If the user wishes to use the Driver from machine code, it is a simple case of working out the relevant address for the function required by adding the value of D% to the offset in the Procedure and Function definition list shown below.

Function	Basic	Machine Code
XPOINT	?(D%+38)	LDA &AAAA
XPOINT	?(D%+39)	LDA &AAAA
XTEXT	?(D%+37)	LDA &AAAA
XGRID	?(D%+40)	LDA &AAAA
YGRID	?(D%+41)	LDA &AAAA
UPDATE	?(D%+31)	LDA &AAAA
SWITCH	?(D%+29)	LDA &AAAA
TRIM	?(D%+30)	STA &AAAA
XDIV	?(D%+32)	STA &AAAA
YDIV	?(D%+33)	STA &AAAA
LPMODE	?(D%+42)	STA &AAAA
LPINIT	CALL D%	JSR &AAAA
LPEXIT	CALL(D%+21)	JSR &AAAA
RESET	?(D%+31)=0	LDA £&00 STA &AAAA

Where AAAA represents the relevant address calculated by adding the offset to your value for D%.

Range limits for modes:-

MODE	0	1	2	3	4	5	6	7
FN_XTEXT	80	40	20	80	40	20	40	40
FN_XPOINT	80	80	80	80	40	40	40	40
FN_YPOINT	32	32	32	25	32	32	25	25

(data returned is in the range 0 to limit-1).

Mode 0 Drawing (DRAW, side 2)

1) 'DRAW' is a simple program in both its' use and construction. The parts of interest to the user are as follows:

a) The Trimming Routine.
The routine will set the pen to all of its' usual trim positions (-3 to +3) and take 20 readings for each setting. These are then averaged and the trim with the lowest 'score' is accepted.

b) The OSCLI routine.
The routine accepts a string (passed as a parameter of PROCFI) which decides on whether to 'load' or 'save'. The command line is then built up as $Z% (see user guide for full explanation) and passed to the command line interpreter (&FFF7) for action. The two routines CALLed by L% and S% remove the bottom (visually, upper in memory) section of the screen and replace the same section respectively negating the need for a continuous 'menu' to be present.

c) The Draw routine.
To enable the program to be sufficiently swift and effective the heart of the draw routine is in machine code. It will be of interest to users that RH's new lightpen program 'ART MASTER' uses a (vaguely) similar approach to produce high quality, free hand, graphics which do not have that 'computer feel' about them. Users are strongly advised NOT to attempt to alter the 'Draw' or 'Move' sections of the program.

d) The Title Page routine.
Although it is simple, an interesting point is the use of the 'shifted' function keys to put colours directly into the display lines

e) The program variables.
Unlike the explanation of the lightpen driver earlier in this manual 'DRAW' does not use labelled calls or procedures to enter the software neither does it use labelled addresses or functions to read results. This is due to a restriction on space (particularly in disc systems) and is not to be recommended as standard practice to all but the most capable of programmers or those of you who run out of space. You may like to note that (with a couple of exceptions) the variables used are either of the 'resident integer' variety or directly addressed zero page locations (&70 to &8F). This policy ensures the minimum amount of BASIC stack space is used and, as most lightpen programs will be similar in construction and will not need floating point (real) numbers, will be useful in most applications.

2) Using DRAW.

The instructions at the beginning of the page cover most points, but, the following is not covered.

a) To SAVE press 'S': type in the file title (including the qualifier if required) and press return. To abort this press 'ESCAPE' and 'RETURN'.

b) To LOAD press 'L': then follow the same actions as for 'SAVE'.

c) During the 'trim' phase the pen MUST be kept pressed to the white block (the computer will emit a tone to show that you are on target).

d) Once the screen has cleared to a white background:

Moving the pen around the screen with the nib PRESSED will result in a line following the pen's movements with a delay ie. if you draw a line WITH THE PEN from top to bottom of the screen and then stop WITH THE PEN PRESSED the line will take a short time to catch up.

e) When moving (shift or space): the new line will start at the point the pen was pressed against on the screen. There is no need to keep shift or space depressed... that bit's done in the software!

f) When a screen is saved: the picture saved will be black + white (as for a normal mode 0 screen) regardless of the currently displayed pair of colours.

g) When a screen is loaded: the picture will be displayed in the currently selected pair of colours.

DRAUGHTS (DRAFTS, side 1)

a) Operation

The game is played in exactly the same way as a normal game of draughts would be played ie:

i) players take it in turns to move
ii) more than 1 piece may be taken during one move
iii) pieces reaching their opponents side of the board become 'crowned'
iv) pieces may only move 'forward' until crowned
v) the game reaches its' conclusion when either players' supply of pieces has been exhausted
vi) pieces can ONLY be moved with the lightpen

b) The Program

The program follows the standard 'polling loop and effect' construction. Proc seek is the only section DIRECTLY concerned with data acquisition from the light pen and is used in all instances that data is required.

The program sets up an 8 x 8 grid (the board) both on the screen and in the 'driver and then fills each 'square' with either colour 8 or 15 (re-defined to be black or green). The pieces are inverted onto these squares which causes them to appear as black or white respective to their initial colour.

During operation the program will;

i) Check (using pros seek) that the correct player is taking their turn.
ii) Check that the correct piece (ie black or white) is being moved.
iii) Remember the GRAPHIC position so that the piece can be returned if necessary
iv) On the next nib depression it will check to see if

square to which the piece is being moved is;

a) Empty
b) In the right direction (if the piece is not crowned)
c) Causing a 'take' to occur
d) Not too large a jump

v) Assuming that the above conditions are fulfilled the piece will be placed in its' new square.

vi) If the move was a 'taking' move the 'next player' option will be left open (either player may move), otherwise 'next player' will be changed.

vii) During a 'take' the number of pieces belonging to opponent is reduced by one, and this variable dictates when the game will end.

KWARTET (KWARTET, side 1)

Is a version of the well known 'four in a row' program for two players. As in the 'Draw' program space has had to be saved and for that reason the calls/data areas have been treated as indirection addresses rather than the procedures/functions shown earlier; the outcome is the same.
The vast majority of the program is concerned with screen display handling, erroneous move checking, sounds etc and, as such, is not directly concerned with the lightpen. The following routines are:
Line 280: initialise the 'pen into mode 2
Line 290: Test for switch depression and act accordingly
     Line 710: If the pen was pointed at the screen update X% from x-grid.
Line 730: If the switch is depressed set the switch flag (SW%)
Line 1130: Updates X% from x_grid.
Line 1280: Updates X% from x_grid.

The procedure titles are (for the most part) self explanatory and users should find little problem in navigating their way around.

An interesting use of CLG has been found by the author in PROC WINDOW in that: the blocks dropped by the current player are NOT drawn onto the screen in the usual manner, rather the area to be displayed (or cleared) is treated as a graphics window and then cleared. This is much quicker, and tidier than drawing triangles (which tend to be too slow for 'animation' effects).

Mode 2 Picture File (SWAN, side 2)

To load this picture type:

```
MODE 2
*LOAD "SWAN" 3000
```

This file was created in about 3 hours (lunch breaks mostly!) using GRAPHIC, ARTFUN and ART MASTER and demonstrates the remarkable versatility obtainable with a lightpen, particularly if compared with similar 'keyboard only' packages. The member of staff concerned is not a trained artist, designer etc but was capable of the quality visible due to the ability to experiment without damaging. For further information regarding these exciting software packages please contact us on the 'phone number on the back of this manual.

R.H. Electronics (Sales) Limited, Chesterton Mill, Frenchs Road, Cambridge
CB4 3NP. Telephone: (0223) 311290

R.H. Electronic (Sales) Limited reserves the right to alter the specification of its products as part of a policy of continuous development.

Printed by: S-Print, 24 Milton Road, Cambridge CB4 1JY. Tel: Cambridge (0223) 311334.