LIGHT PEN

by Chris Barlow and Martin Taylor

Being Atari owners we all possess a computer with the ability to accept a light pen input. But if you have ever tried to obtain a ready built unit, you may have been amazed at the cost and lack of available sources. This is partly due to the lack of software that the use of this device requires, and the difficulty in manufacturing a reliable piece of hardware. Some manufacturers have attempted to produce such a device but, due to marketing considerations i.e. cost and predicted sales, the resulting hardware leaves a lot to be desired. In this article we present a Light Pen which should cost, in components, less than half the price of a commercially available unit with, in our opinion, a superior performance.

Method

To explain how a Light Pen works, you must first have an idea of how the television picture system is generated. A TV picture is basically constructed from a number of lines produced on the phosphor coating on the inside of the screen. The original TV system used 405 lines, but today 625 lines is the accepted UK standard. However, the Atari does not use all 625 lines. The phosphor on the screen will glow when electrons, produced from the electron gun, strike it. This will produce a single spot of light on the TV screen. To produce lines it is necessary to deflect the electron beam across from left to right, thus obtaining the horizontal or X axis. When the line has reached the right-hand side of the screen it is then deflected back to the left-hand side of the picture and down slightly to produce the next scan starting position. During the return period of scan, the electron gun output is blanked in order not to generate spurious lines. The downward scanning, or Y axis, continues until all 625 lines have been drawn, at which time the beam is made to return back to the top starting position. This is an over simplified description and, in reality, the total procedure is much more involved.

The Light Pen is designed so it can detect a pulse of light coming from the screen. The computer has the job of determining the X and Y coordinates of this light pulse. These values are obtained from the internal register set of the Antic Display processor. Since the position of the light pulse on the screen is directly related to the time it took to get there from the beginning of the first scan position, the hardware can determine X and Y values and store these in two hardware registers. When programming in Basic the X and Y values are obtained by PEEKing locations 564 for X and 565 for Y. The user's software has then to interpret

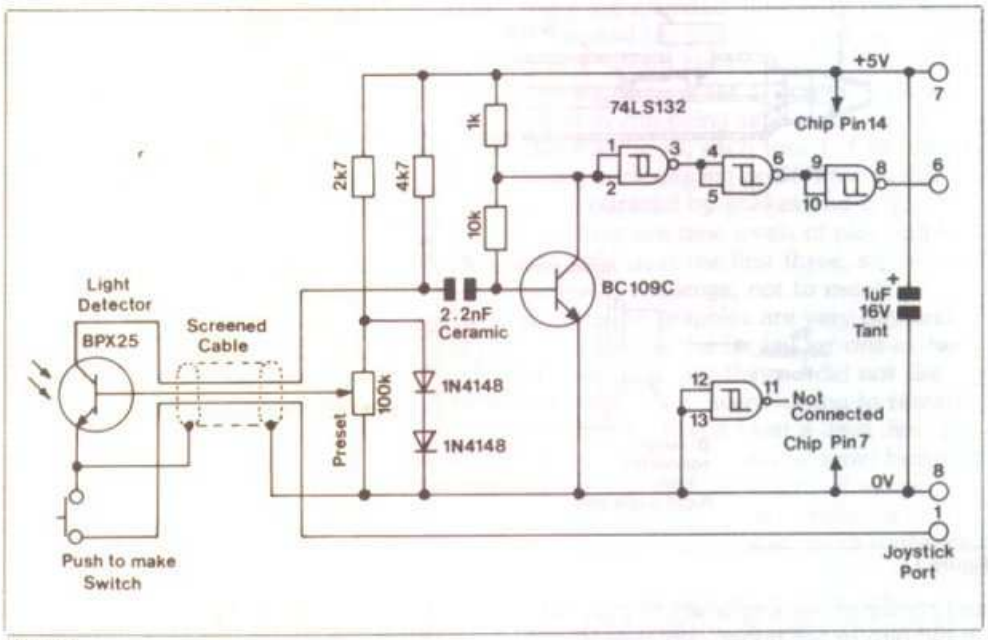

Figure 1

these values in order to obtain screen position related values. The horizontal, or X, location (564) will return a value of 78 for the extreme left-hand side of the screen, increasing in increments of one, up to a value of 227. Then something rather strange happens, the value jumps to zero and then increments up to a final value of 8 for the extreme right-hand side. This is a fault due to the Atari hardware and not to the Light Pen. It appears that Atari are not concerned with this fault since it has not been corrected in their new range of computers. Although a problem, it can be rectified by software means. The vertical, or Y, location (565) will return a value of 16 for the extreme top of the display, incrementing by one to value of 111 at the extreme bottom of the display. This register appears to function correctly. The values stored at these two locations are updated when any of the four joystick trigger inputs are used.

Circuit

To attempt to construct this project, some knowledge of electronics and the ability to solder will be necessary. As can be seen, in the circuit diagram (figure 1), there are very few components necessary to obtain a working Light Pen. The most important is the Light Detector. It must have good sensitivity and fast reaction characteristics. The BPX25 photo-transistor meets both requirements, at a modest cost. This device is equipped with its own built-in optical lens, which is made of glass. This point is worth noting, since, if direct contact is made with the glass of the TV screen, scoring may occur. To prevent this the BPX25 should be recessed into a plastic tube of some description. To obtain maximum sensitivity and operating speed, it is necessary to bias the base of the transistor. The voltage required is quite small, approximately 0.5 Volt. This voltage is adjustable by the 100k preset from 1.2 Volts down to 0 Volts. In practice the preset wiper position comes out about half way round its travel. The 1.2 Volts at the top end of the preset is generated by two silicon diodes in series, and forward biased. The current through the diodes is limited by the 2.7k resistor connected to the +5 Volt supply, taken from pin 7 of the joystick port.

When the photo-transistor detects a light pulse, the amount of current flowing through it changes. The current is limited through the device by the 4.7k resistor in its collector circuit. These changes in current cause a voltage change at the collector of the photo-transistor. The voltage pulses are then coupled, via a 2.2nF ceramic capacitor, into the base circuit of the BC109C transistor. This device performs the necessary level change to obtain TTL logic levels. The final stage of shaping the pulse is achieved by using a 74LS132, a quad two input Nand Schmitt Trigger. As can be seen, only three of the four gates are used. The final component in the circuit is a 1uF 16 Volt tantalum bead capacitor across the supply rails, which removes any spurious noise on the power lines. The output of the final gate is fed to pin 6 of the joystick port. The ground connection is made to pin 8.

In the prototype, a push-to-make switch was used as a trigger for the Light Pen. The switch was simply connected between pin 1 and pin 8 of the joystick port. The final construction

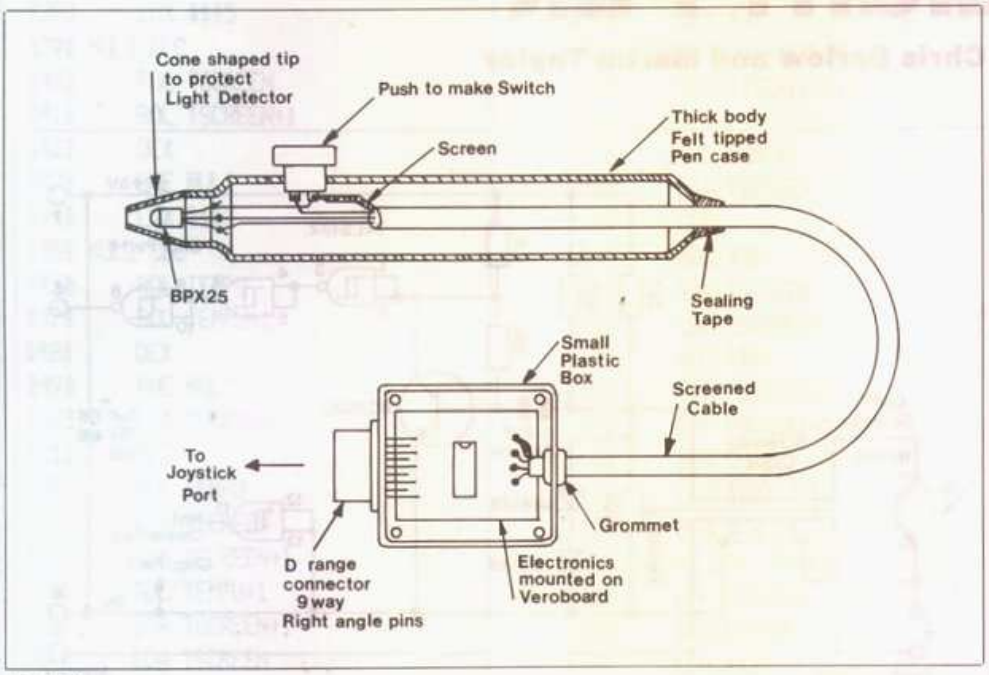

Figure 2

and choice of housing is left to you, but an old biro or felt-tipped pen case is ideal for the pen itself, and the electronics can be housed in a small plastic box (see figure 2). An electronic consideration though is that the cable linking the photo-transistor to the circuit board must be screened to prevent stray interference pickup. The prototype used 4 core overall screened cable, but individual screened audio cable is quite acceptable. The electronics can be constructed on a small piece of Vero board, or for the more ambitious you could etch your own PCB. Connection to the joystick is via a standard D-Type 9-way connector.

Programs

Included in this article are three very simple programs, the first of which is used to set up the 100k preset in the circuit. In all three programs we have used joystick port 1, this is because of the Light Pen's switch controlling the value of STICK (0). However, the Light Pen will work in any of the four joystick ports. Program 1 is a simple drawing utility which will produce lines or dots depending upon the state of the function keys. Holding Select down will put the drawing program into dot mode, and the Option key will clear the screen and reset the starting position to the current pen position. Pressing the Light Pen's own button will produce continuous line drawing. To set a new starting point simply place the pen on the screen and press the Select key. To adjust the preset to obtain the correct results simply hold the Light Pen against the screen, press the switch on the Light Pen and move the pen slowly. If the line does not trace the movement, adjust the preset until it does. If you cannot obtain a satisfactory result, try increasing the brightness and contrast controls on your TV. If there is still no response recheck your soldering and construction.

Program 2 is an example of how a Light Pen can be used for Menu-Driven software. Position the pen over the number you wish to choose and press the Light Pen switch. If all is well, a tone will be heard and your selection will be shown at the bottom of the screen.

The final program (Program 3) is a very simple musical instrument, in which you can select both volume and pitch. The sound will only be present whilst pressing the Light Pen switch. The display on the screen is a matrix of square dots with volume increasing down the screen and pitch increasing across the screen, right to left.

In conclusion, we must point out, that the programs shown are by no means good examples of what can be achieved, but are adequate for testing purposes and demonstrating the principles behind Light Pen Software implementation. When writing your own software, you must bear in mind where the screen is dark, no information can be detected by the Light Pen. It is hoped that in the future we will see more software using Light Pens, so get writing!

Program 1

```
10 GRAPHICS 24:COLOR 1
20 X=PEEK(564):X=X-155+X:IF X<1 THE
N X=1
30 Y=PEEK(565):Y=Y-30+Y:IF Y>190 TH
EN Y=190
40 IF PEEK(53279)=3 THEN GOSUB 80
50 IF PEEK(53279)=5 THEN PLOT X,Y:G
OTO 60
60 IF STICK(0)<>15 THEN DRAWTO X,Y
70 GOTO 20
80 GRAPHICS 24:COLOR 1
90 PLOT X,Y:RETURN
```

Program 2

```
10 REM MENU
20 GRAPHICS 2+16:SETCOLOR 0,0,12:SE
TCOLOR 4,4,1
30 PRINT #6;"__ atari 1"
40 PRINT #6;"__ atari 2"
50 PRINT #6;"__ ATARI 3"
60 PRINT #6;"__ ATARI 4"
70 PRINT #6;"__ atari 5"
80 PRINT #6;"__ atari 6"
90 PRINT #6;"__ ATARI 7"
100 PRINT #6;"__ ATARI 8"
110 PRINT #6;"__ atari 9"
120 PRINT #6;"__ atari 10"
130 IF STICK(0)<>15 THEN 150
140 GOTO 130
150 LET I=PEEK(565)
160 IF I<18 OR I>94 THEN 150
170 IF I=18 OR I=19 OR I=20 OR I=21
 OR I=22 THEN M=1:GOSUB 280
180 IF I=26 OR I=27 OR I=28 OR I=29
 OR I=30 THEN M=2:GOSUB 280
190 IF I=34 OR I=35 OR I=36 OR I=37
 OR I=38 THEN M=3:GOSUB 280
200 IF I=42 OR I=43 OR I=44 OR I=45
 OR I=46 THEN M=4:GOSUB 280
210 IF I=50 OR I=51 OR I=52 OR I=53
 OR I=54 THEN M=5:GOSUB 280
220 IF I=57 OR I=58 OR I=59 OR I=60
 OR I=61 THEN M=6:GOSUB 280
230 IF I=65 OR I=66 OR I=67 OR I=68
 OR I=69 THEN M=7:GOSUB 280
240 IF I=74 OR I=75 OR I=76 OR I=77
 OR I=78 THEN M=8:GOSUB 280
250 IF I=82 OR I=83 OR I=84 OR I=85
 OR I=86 THEN M=9:GOSUB 280
260 IF I=90 OR I=91 OR I=92 OR I=93
 OR I=94 THEN M=10:GOSUB 280
270 GOTO 130
280 IF MM=M THEN RETURN
290 POSITION 4,11:PRINT #6;"ATARI="
;M;" "
300 FOR V=15 TO 0 STEP -1:SOUND 0,M
*10,10,V:NEXT V:LET MM=M:RETURN
```

Program 3

```
10 GRAPHICS 4+16:COLOR 1
20 SETCOLOR 4,2,3:SETCOLOR 0,0,15
30 FOR Y=0 TO 47 STEP 4
40 FOR X=0 TO 70 STEP 4
50 PLOT X,Y
60 NEXT X:NEXT Y
70 IF STICK(0)<>15 THEN 90
80 SOUND 0,0,0,0:GOTO 70
90 SOUND 0,PEEK(564)/3,10,PEEK(565)
/10
100 GOTO 70
```