


ATARI® 400/800™

# LIGHT PEN

OPERATOR'S MANUAL



A Warner Communications Company 

Model CX70  
Use with  
ATARI® 400™ or ATARI 800™  
PERSONAL COMPUTER SYSTEMS

ATARI  
Light Pen  
Operator's Manual

COPYRIGHT 1981 ATARI, Inc.

Every effort has been made to ensure that this manual accurately documents this product of the ATARI Computer Division. However, because of the ongoing improvement and update of the computer software and hardware, ATARI, INC. cannot guarantee the accuracy of printed material after the date of publication and cannot accept responsibility for errors or omissions.

CONTENTS

1. INTRODUCTION
2. POSSIBLE USES OF THE LIGHT PEN
3. CONNECTING THE LIGHT PEN
4. ATARI LIGHT PEN DEMONSTRATION PROGRAM CASSETTE
  - Loading Instructions
  - 8K Version of PALLET: A Light Pen Painting Program
  - 16K Version of PALLET
5. PROGRAMMING EXAMPLE 1: Using the Light Pen with ATARI BASIC
  - How Programming Example 1 Works
  - Experimenting with Programming Example 1
6. PROGRAMMING EXAMPLE 2: Using PENBUF - A Light Pen Utility Program
7. PROGRAMMING EXAMPLE 3: A Keyboard Substitute
8. PROGRAMMING EXAMPLE 4: A Teaching Aid
9. PERIODIC MAINTENANCE

APPENDICES

- A. BASICS OF LIGHT PEN OPERATION
- B. ATARI LIGHT PEN CHARACTERISTICS
- C. TROUBLESHOOTING: What to Do If Your ATARI Light Pen Does Not Respond as Expected
- D. PENBUF FLOW CHART
- E. PENBUF ASSEMBLY LANGUAGE LISTING
- F. DESCRIPTION OF PENBUF OPERATION
- G. NOTES ON MEMORY CONSERVATION/PENBUF MEMORY USAGE
- H. GENERAL PROGRAMMING NOTES

ILLUSTRATIONS

Figure	Page
1 Sample Menu	3
2 Television Screen Display for 8K PALLET Program	9
3 8K Version of PALLET Program Listing	11
4 Television Screen Display for 16K PALLET	13
5 Flowchart for 16K PALLET	19
6 16K Version of PALLET Program Listing	21
7 Color Mode Numbers	23
8 Programming Example 1: Using the Light Pen with ATARI BASIC	29
9 Television Screen Display for Programming Example 1	30
10 PENBUF Loader	33
11 Programming Example 2: Using PENBUF	34
12 Programming Example 2: Modified to Run in Graphics 0	35
13 Television Screen Display for Programming Example 2	36
14 Programming Example 3: A Keyboard Substitute	37
15 Television Screen Display for Programming Example 3	38
16 Programming Example 4: A Teaching Aid	39
17 Television Screen Display for Programming Example 4	40
A.1 Television Screen Simulation	45
A.2 Television Screen Plus Light Pen	46
D.1 PENBUF Flow Chart	53
F.1 Typical Insert-Sort	59
G.1 PENBUF Memory Usage	62

## SECTION 1 - INTRODUCTION

The ATARI Light Pen is particularly suitable as an educational and game playing tool as it allows the user to interact directly with the computer. By pointing to an item on the screen, you can say "There's the game piece I want to move," and "There's where I want to move it." The most common use, however, is to select an item from a list or menu, as in a multiple-choice test.

Section 2 of this manual describes menu selection and other more sophisticated uses of the light pen. Sections 3 and 4 give instructions for connecting the light pen to the computer and loading the programs on the demonstration cassette. In Sections 5 through 8, you will find several more demonstration programs to help you understand how to write BASIC programs using the pen for your own purposes. These examples will lead you to invent many ways to use the light pen in your programs. In these examples, we assume you have a working knowledge of ATARI BASIC as defined in the ATARI BASIC Reference Manual. Although the demonstration cassette programs and all of the programming examples in this manual are in BASIC, you may use the pen with any of the ATARI programming languages.

General programming notes, showing the basic rules for the use of the light pen are provided in Appendix H. Read through the manual thoroughly first to acquaint yourself with the operating principles of the pen. In Section 9, you will find details of the care necessary to maintain the pen in its best operating condition.

ATARI Light Pen Operator's Manual

SECTION 2 - POSSIBLE USES OF THE LIGHT PEN

Menu selection, which is a common use for the ATARI Light Pen, may take many different forms. One form of light pen selection is illustrated by the Keyboard Substitute program, provided as Programming Example 3 (see Section 7). In this program, the pen is used to select the letter to be typed on the screen. Programming Example 4, the Teaching Aid Program, uses the pen in two different menus. The demonstration program provided on the cassette uses the pen as a menu selector as well as a graphics tool.

By using the ATARI Light Pen with programs you have written, you can form a picture, a plot, a set of letters or groups of instructions on the television screen. With the pen, you can point to one of those items on the screen. The computer will read signals coming from the pen and can interpret those signals so that it knows what action to take next. As an example, you can produce a list of the different programs you have available on the diskette in your ATARI Disk Drive. The list (or menu) might appear as shown in Figure 1.

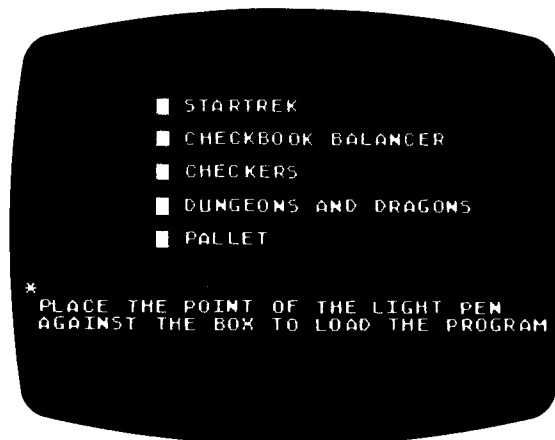


Figure 1 - Sample Menu

When you follow the instruction at the bottom of the television screen, your program will sense that the switch in the tip of the light pen is pressed and will call the light pen subroutine.

The pen can also serve as a teaching device when a student uses it to select the correct answer from a multiple-choice set of answers placed on the television screen. A form of this is demonstrated in Programming Example 4, a Teaching Aid, in Section 8.

In each of these programs, items are placed at various known locations on the screen. When the light pen registers are read, the program compares the values obtained there to the values known to represent

## ATARI Light Pen Operator's Manual

---

the position of the test characters on the screen. This information tells the computer what to do next.

You may extend this control function to sense a selected menu item and perform a complete function. Such functions may include: turning on the Program Recorder, sounding an alarm, choosing a mathematical formula to be executed, selecting a specific disk file, and others.

You may use the light pen as a high-speed joystick (game paddle) to move the cursor to a new location on the screen directly rather than using the keyboard or paddle cursor controls. In games such as tic-tac-toe, chess, or checkers the light pen can "pick up" a specific item from a selected position on the game board or place that item or a marker at another game board position.

In business applications, an office planner can use the light pen to pick up a graphics-represented piece of furniture and reposition it in another area in his graphics-represented room.

You can experiment with the ATARI sound generators by plotting one or more organ keyboards on the screen. You can use the light pen to "play" each one of the organs by selecting notes from each of the keyboards. Each key represents one note of a selected sound generator, and each keyboard represents a separate sound generator.

There are certain applications for which keyboard input would be too complex or time-consuming. In a case like that, the computer can do all of the work, determining the pen position and plotting items on the screen on that basis.

A Light Pen Demonstration Program Cassette has been provided to illustrate such an application. The cassette contains two copies of a program called PALLET, with an 8K version on Side 1 and a 16K version on Side 2. Each version allows you to draw with the light pen on the screen by plotting colored dots at each position occupied by the pen. The 16K version includes many extra features. These programs are described in Section 4, ATARI Light Pen Demonstration Program Cassette. These suggested uses are only a beginning. You will probably invent many other ways to use the ATARI Light Pen in your own programs.



SECTION 3 - CONNECTING THE LIGHT PEN

The light pen connector is exactly like the game controller connectors. It can be inserted into any one of the four game controller ports on the front of your ATARI Personal Computer System.

The demonstration programs have been written to look at controller jack #1 (far left jack). Later in the manual you will be told how to use the pen in the other jacks (see Appendix H). But for now, insert the pen into controller jack #1.



SECTION 4 - ATARI LIGHT PEN DEMONSTRATION PROGRAM CASSETTE

LOADING INSTRUCTIONS

1. Insert an ATARI BASIC (Computing Language) cartridge into your ATARI 400(TM) or ATARI 800(TM) computer console cartridge slot. (Insert in the LEFT CARTRIDGE slot on the ATARI 800 computer console.)
2. Make sure the ATARI 410(TM) Program Recorder is properly connected to the computer console, and to a wall or power outlet. (See your Program Recorder Operator's Manual for further details if necessary.)
3. Turn your television set on.
4. Turn on the computer console by pressing the POWER switch on the right side of the console to ON.
5. If all equipment is properly connected and powered up, your television screen should be displaying the READY prompt, with the white square "cursor" just below.

NOTE: If you have problems loading the program, and if you have other peripherals in addition to the Program Recorder attached to the computer console, try disconnecting the other peripherals and connecting the Program Recorder directly to the computer console to isolate the problem.

6. Insert Side 1 of the Light Pen Demonstration Program Cassette (8K version) into the ATARI 410 Program Recorder. If your computer has 16K RAM or more installed, try the 16K version on Side 2 of the cassette. Press the REWIND button on the Program Recorder to wind the tape back to the beginning if it is not already at that position. When the tape stops, push STOP (STOP/EJECT).
7. Type CLOAD on the computer keyboard and hit [RETURN]. The "beep" sound is a reminder to press PLAY on the Program Recorder. Press PLAY and hit the [RETURN] key again on the computer console.

Through the window of the Program Recorder, note that the tape is turning. The beeps and other sounds you hear coming from the television speaker tell you that the program is being loaded into computer RAM.

8. When the screen displays the READY prompt again, type RUN and press [RETURN]. The television will display the ATARI logo and the message LOADING LIGHT PEN. You will hear another beep as the program automatically begins to load into the computer. When loading is completed, the menu will be displayed and the program is ready for use.

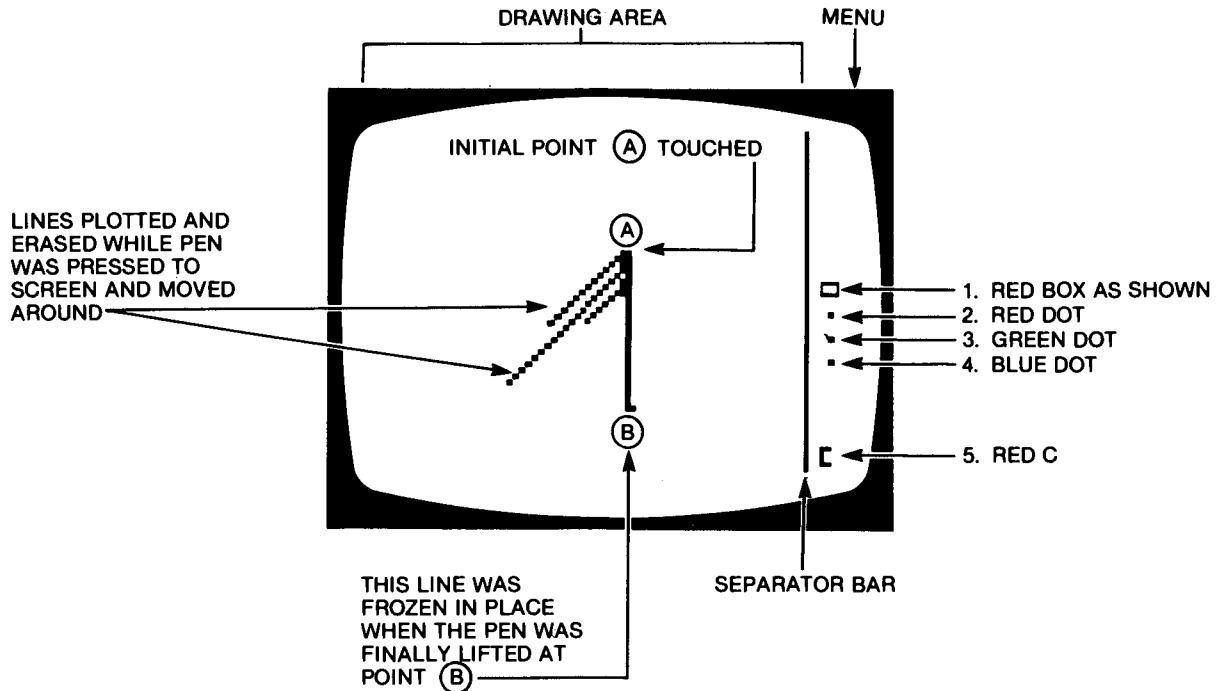
8K VERSION OF PALLET: A Light Pen Painting Program

Once the 8K version of PALLET has been loaded, the screen will display a menu block drawn along its right-hand edge. There are five possible menu selections provided in this program. The top selection indicates background color mode. If the pen is currently in this mode, any points plotted with the pen will be in the background color. This has the effect of erasing any other color which may have been plotted there previously.

The next three menu choices are the color selectors. When you press the pen against the screen at any one of these spots, the color of the points plotted by the pen will be in the color you selected.

The bottom menu selector is the CLEAR block. When you press the pen against this block and lift it, the screen is cleared and reinitialized.

To understand how the program works, place the pen against the screen somewhere near the center. A colored dot will appear if the pen is not in the background color mode. Continue to press the tip of the pen against the television screen and move the pen across the screen. A line will follow the pen as it moves. This line extends from the original spot at which the pen was placed, to the spot now occupied by the pen. It appears to be elastic (like a rubber band) because the line is continuously being plotted and then erased as the pen is moved. Lift the pen from the surface. The line you have drawn is now frozen in the last position it occupied. Draw a few more lines this way.



FUNCTIONS: (If pen is pressed to this character)

1. Set Pen color to background ("erase" color if plotted)
2. Set Pen color to red for plotting
3. Set Pen color to green for plotting
4. Set Pen color to blue for plotting
5. Clear entire screen, redraw the menu

Figure 2 Television Screen Display for BK PALLET Program

Press the pen against one of the other color blocks in the menu area. Now draw one or more lines. The pen color is now the new color you selected. Try the third color also.

When you wish to erase the screen and start over, press the pen against the bottom box in the menu area. As it is lifted, the screen will clear and the menu will be redrawn.

The program listing for the BK version of PALLET is shown in Figure 3. (Refer to Appendix G for additional notes on this program listing.) At this point the listing will seem somewhat cryptic, especially lines

ATARI Light Pen Operator's Manual

9000/9010. However, these characters are machine language stored as ATASCII characters to conserve space. Once you have read through Sections 5 through 9, turn back and reread the listing. It should seem fairly straightforward.

If you have an 8K machine, we strongly recommend that you read the listing, the flowchart, and the description of the 16K version to learn how much the pen can really do.

```

1 REM 8K VERSION OF "PALLET"
10 REM MOD A
40 DIM A$(117):GOSUB 9000
45 PENCOLR=3:LOC=1536:POKE 205,3
70 GOTO 910
80 REM MOD J
105 POKE 77,0
110 Q=INT(Y/8)
125 IF PEEK(632)=14 THEN 125
130 IF Q>=5 AND Q<=8 THEN 450
135 IF Q=11 THEN 910
140 REM MOD E
150 GOSUB 2000
160 XP=X:YP=Y:X1=X:Y1=Y
170 GOSUB 2020
190 COLOR 0:PLOT XP,YP:DRAWTO X1,Y1
210 COLOR PENCOLR:PLOT XP,YP:DRAWTO X,Y
220 X1=X:Y1=Y
230 IF PEEK(632)=14 THEN 170
240 GOTO 150
449 REM MODS L,M,N,O
450 PENCOLR=Q-5:COLOR PENCOLR:GOTO 150
870 REM MOD R
910 GOSUB 1000:GOTO 150
980 REM MOD C
1000 GRAPHICS 23:COLOR 1:SETCOLOR 4,0,8
1015 SETCOLOR 0,4,14:SETCOLOR 1,12,14:SETCOLOR 2,9,14
1020 PLOT 141,0:DRAWTO 141,95
1080 PLOT 146,41:DRAWTO 146,45:DRAWTO 152,45:DRAWTO 152,41:DRAWTO 146,41
1090 PLOT 149,51
1100 COLOR 2:PLOT 149,59
1110 COLOR 3:PLOT 149,67
1160 PLOT 150,89:DRAWTO 148,89:DRAWTO
148,93:DRAWTO 150,93
1200 RETURN
1210 REM MOD I
2000 IF PEEK(632)=15 THEN 2000
2020 Q=USR(LOC)
2040 X=PEEK(206):Y=PEEK(207)
2045 IF X<0 OR Y<0 OR X>159 OR Y>95 TH
EN 2000
2048 IF X>140 THEN POP :GOTO 110
2050 RETURN
9000 A$=""
9010 A$(LEN(A$)+1)=" "
9020 Q=1535:FOR X=1 TO LEN(A$):POKE Q+
X,ASC(A$(X,X)):NEXT X:RETURN

```

READY

Figure 3 8K Version of PALLET Program Listing

## 16K Version of PALLET

Figure 4 is a television screen display for this program. A flow chart for the program is shown in Figure 5, and Figure 6 is the complete program listing.

If you have already tried the 8K version of this program, you will be able to see the additional features which this version provides, primarily because this version assumes there is more memory available to work with. The 8K program had only five menu choices along the right side of the screen. This version has 12 menu choices on the right side. Any of these choices may be selected by pressing the pen to the screen in that area.

Of the 12 choices, the top five are known as drawing modes, the next four are color modes and the last three are functions to be performed.

Within the blocks which define the drawing modes is a segment which acts as an indicator bar. This indicator bar is the same color as the border and appears at the far left edge of the selector box representing the current drawing mode. There is another indicator bar which appears in one of the color selector blocks. This tells you the current color of the pen.

Among the color modes, you can select to draw in red, green, blue or "erase" color (which sets the spot under the pen tip to the background color). These selection modes are identical to those used in the 8K program; however the selector blocks are larger and more clearly defined.

In the 8K version, you only had the ability to draw in the ELASTIC mode. Elastic refers to a line which acts like a rubber band as the pen moves. This program provides the selection of ELASTIC mode by a large E selector box.



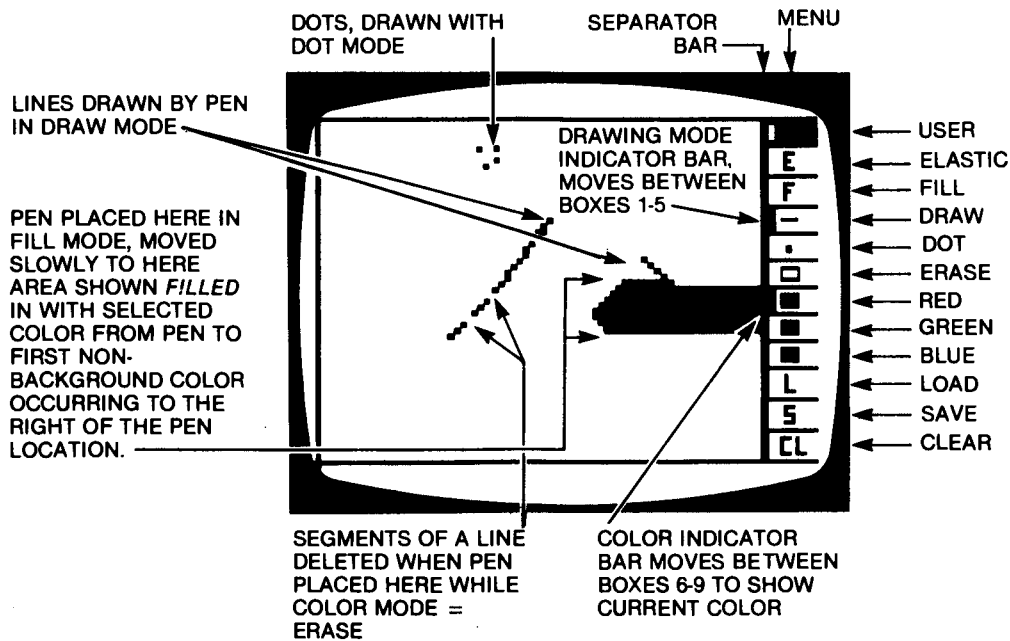


Figure 4 Television Screen for 16K PALLET Program

Functions: (When pen is pressed to this area)

filled in  
block

is the user programmable function

E

sets the pen drawing mode to ELASTIC.

F

sets the pen drawing mode to FILL (see Figure 4).

—

sets the pen drawing mode to produce lines.

dot

sets the Pen Drawing mode to produce dots.

open square

sets the pen color to background color (ERASE mode).

red square

sets the pen color to red.

green square

sets the pen color to green.

blue square

sets the pen color to blue.

L

will load a picture from diskette or cassette (change filename in lines 710, 760).

S

will save the current picture to diskette.

CL

will erase entire screen, then redraw the menu.

### ELASTIC Mode

To understand how the program works, press the pen to the E block. The drawing mode indicator bar will move into this block. Place and hold the pen against the screen to the left of the border bar. As with the BK version, a line will be continuously drawn and erased from the point of initial contact. The line will be frozen in place when you lift the pen. The line is drawn in the color that is shown as the current selection by the color indicator bar.

Press the pen against a different color block. The selector bar will move into this block. The pen will now draw in the new color.

### FILL Mode

The menu block beneath the E is an F selector block, which indicates FILL mode. After you have drawn other lines on the screen in E mode, press the pen to the F block. The indicator bar will move into this square.

In the FILL mode, when the pen is pressed to the screen, a line from the pen to the first occurrence of a color other than the background color, located to the right of the current location of the pen, will be drawn in the currently selected color. As the pen is moved slowly in a line, the area between the pen and the first plotted color to the right of the pen will be filled in with the current color.

If you lift the pen and press it to the screen elsewhere, a line will be drawn between the previously occupied spot and the new spot, and then all uncolored areas to the right of this new line will be filled in with the current color of the pen.

NOTE: To keep this line from being drawn, press the pen to the F before pressing it to the drawing area again.

### DOT Mode

The fifth block down from the top of the menu is the DOT mode. This places dots in the currently chosen color under the pen. The program takes about 1/20 second to calculate the position of the pen. If you draw on the screen in dot mode, the faster you move the pen, the farther apart the dots will be plotted.

NOTE: The calculation time can be changed in line 40 (see Section 6).

### DRAW Mode

The fourth drawing mode block, represented by a short line, is the DRAW mode. It is like the DOT mode, with a connect-the-dots function

added to it. Try to draw on the screen in this mode. If you lift the pen and press it to the screen elsewhere, a line is drawn from the last point occupied by the pen to the new position. If you wish to prevent this interconnect line from being drawn, press the pen to the DRAW mode block before you press it into the drawing area again.

#### USER Mode

The top block in the drawing mode selector is the USER mode module. Space in the program has been allocated for you to define your own function. In the demonstration program, it is defined at lines 6000-6900. Currently, it retrieves the X and Y values of the pen and does nothing with them. You may choose to add anything you wish to this module, such as plotting X and Y in a specific color.

If you have a disk drive, you could have the user module display a certain color in its menu block. When the pen is pressed there, you could rotate the displayed color through the four available colors. A certain color in this block would then specify one of four disk files that can be used to load and save pictures.

Another possibility for the user module is using it to change the hues and luminances of the picture. It can rotate through all possible combinations, or any one of many schemes you could invent. With a little imagination, you can find endless possibilities for this module.

To use the user module, the first change required to the program would be a modification to line 1030, which currently plots only a solid red block in the user module space. (See Figure 6 for the program listing.) You may elect to plot any letter, number, or other figure in this same space. The coordinates are: X between 144 and 159 inclusive, and Y between 1 and 6 inclusive. Then in lines 6030-6900 you could do something with the X and Y values received from GETPOS (maybe plot X, Y). The last line of the user module (line 6900) is GOTO 6020 (creating an infinite loop) because while you are in any one of the DRAW modes, you should stay there indefinitely until the DRAW mode is changed. Don't worry about the tracking of the modes, the moving of the mode bars, range checking, or menu selection because they have been handled by the rest of the program (study the other modes).

ATARI Light Pen Operator's Manual

---

### Clear Function

At the bottom of the menu block is the CL (Clear) function. When you press the pen to this block and then lift it, the screen is cleared and the menu blocks are redrawn.

### Load and Save Functions

The other two functions that can be selected require that you have an ATARI Disk Drive or ATARI 410 Program Recorder connected to your system. These are the Load and Save functions. They allow you to save a picture to diskette or to load a previously saved picture from diskette (or cassette). When you use the Save function, the picture currently on the screen is saved under the filename D1:PEN.OUT if you are using diskette. More than one picture can only be saved on the same diskette if the program's filename is changed after saving it. For instance, you might save your first picture, then use the Disk Operating System (DOS) to change the filename to D1:PEN.1, and then save another picture. To load the D1:PEN.1 program, you would have to change the filename of the second picture to something like D1:PEN.2 first, then change D1:PEN.1 back to D1:PEN.OUT, before attempting to load it.

While the Load or Save is in process, a dot will appear next to the L or S function block. When the loading or saving is finished, the dot will disappear.

ATARI Light Pen Operator's Manual

FIGURE 5—FLOWCHART FOR 16K PALLET

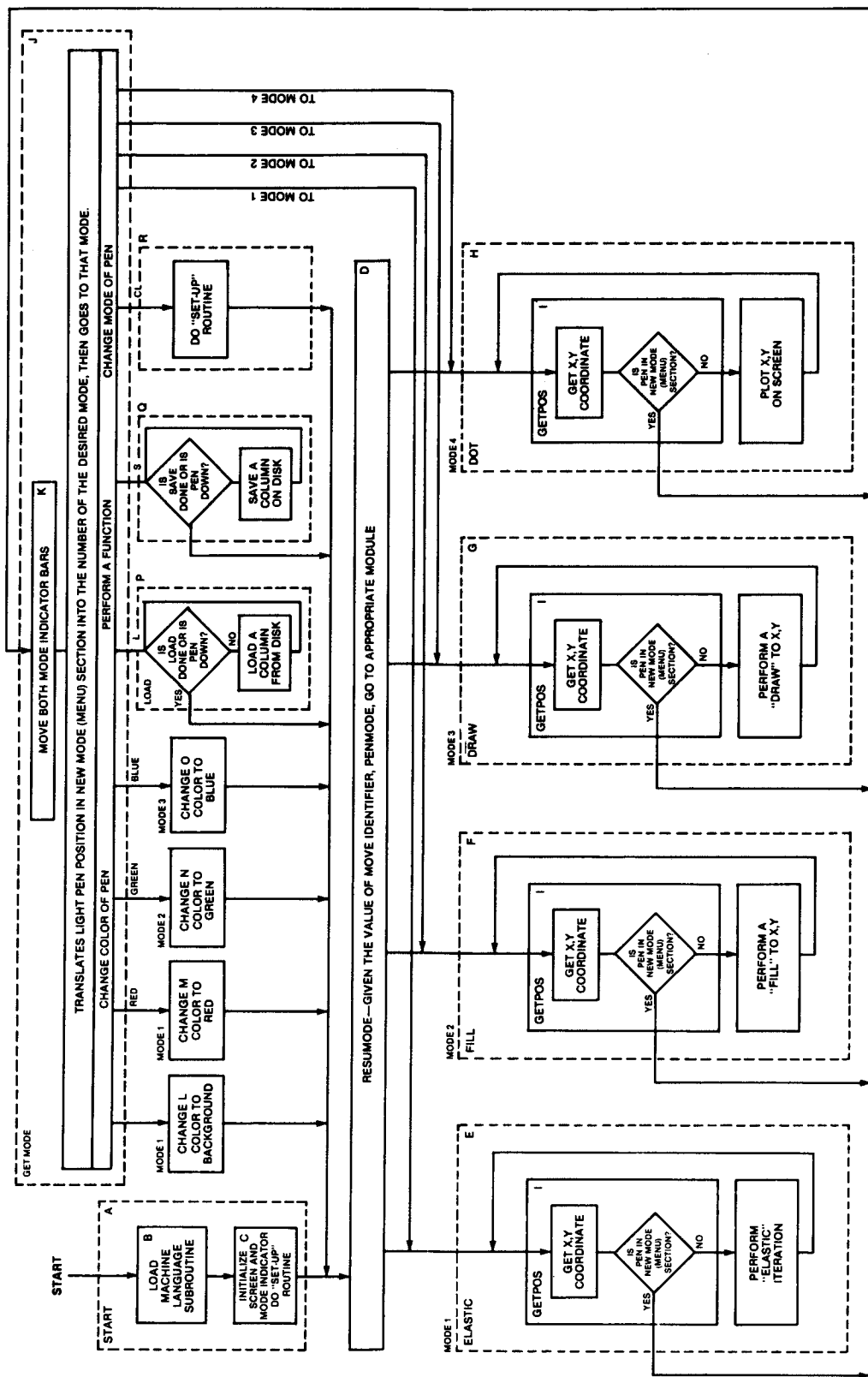


Figure 5 Flowchart for 16K PALLET





```

1 REM "PALLET", LIGHT PEN DEMO
2 REM
5 REM LOAD AND SAVE OPTIONS CURRENTLY WORK ON CASSETTE.
6 REM CHANGE FILE NAME IN LINES 710 AND 760
7 REM FROM "C:" TO "D:PEN.OUT" FOR DISK OPERATION.
8 REM
9 DIM CIO$(7)
10 REM *** MOD A ***
20 REM START
30 GRAPHICS 0:POKE 752,1:POSITION 15,13:PRINT "Hang On..."
40 POKE 205,3:PENBUF=1536:GOSUB 9000:REM LOAD MACH LANG SUB
45 PENMODE=3:PENCLR=3:REM INITIALIZE TO MODE AND COLOR 3
50 GOSUB 1000:REM DO SET-UP
70 GOTO 950:REM OUR MODES ARE INITIALIZED, SO "RESUME" THEM
80 REM *** MOD J ***
100 REM SUB GETMODE
105 POKE 77,0:REM STOP FROM ENTERING ATTRACT MODE
110 Q=INT(Y/8):NPEN=Q:NCCLR=Q-5
120 GOSUB 3000:REM MOVE MODEBARS
125 IF PEEK(632)=14 THEN 125:REM WAIT UNTIL PEN IS LIFTED FROM NEW MODE SECTION
130 ON Q+1 GOTO 6000,150,300,350,400,450,500,550,600,700,750,900
140 REM *** MOD E ***
145 REM ENTER "ELASTIC" MODE
150 GOSUB 2000
152 IF (X1=X) AND (Y1=Y) THEN 150
160 XP=X:YP=Y:X1=X:Y1=Y:REM DEFINE PIVOT POINT
170 GOSUB 2020
190 COLOR 0:PLOT XP,YP:DRAWTO X1,Y1:REM ERASE
210 COLOR PENCCLR:PLOT XP,YP:DRAWTO X,Y:REM NEW LINE
220 X1=X:Y1=Y
230 IF PEEK(632)=14 THEN 170
240 GOTO 150
250 REM *** MOD F ***
300 REM PUT PEN IN FILL MODE
302 GOSUB 2000:PLOT X,Y:REM GET PRELIM POINT
310 POKE 765,PENCCLR
320 GOSUB 2000:REM GET X,Y COORD
330 POSITION X,Y:REM POSTN FOR FILL
340 XID 18,#6,12,0,"S:":REM DO FILL
345 GOTO 320
348 REM *** MOD G ***
350 REM PUT PEN IN DRAW MODE
352 GOSUB 2000:PLOT X,Y:REM GET PRELIM POINT
360 GOSUB 2000:REM GET PEN POSTN
380 DRAWTO X,Y:GOTO 360
390 REM *** MOD H ***
400 REM PUT PEN IN DOT MODE
410 GOSUB 2000:REM GET PEN POSTN
430 PLOT X,Y:GOTO 410
440 REM *** MOD L ***
450 REM CHANGE COLOR TO MODE 0
460 COLOR 0:PENCCLR=0
470 GOTO 950:REM RESUME PREVIOUS MODE
480 REM *** MOD M ***
500 REM CHANGE COLOR TO MODE 1
510 COLOR 1:PENCCLR=1
520 GOTO 950:REM RESUME PREVIOUS MODE
530 REM *** MOD N ***
550 REM CHANGE COLOR TO MODE 2
560 COLOR 2:PENCCLR=2
570 GOTO 950:REM RESUME PREVIOUS MODE
580 REM *** MOD O ***
600 REM CHANGE COLOR TO MODE 3
610 COLOR 3:PENCCLR=3
620 GOTO 950:REM RESUME PREVIOUS MODE
630 REM *** MOD P ***
700 REM LOAD USING CIO
705 COLOR 1:PLOT 144,75
707 POKE 764,12
710 OPEN #1,4,0,"C:"

```

## ATARI Light Pen Operator's Manual

```
720 POKE 850,7:LOD=1:GOTO 800:REM SET LOAD FLAG SO "SET UP" MOD WON'T ERASE
730 REM *** MOD Q ***
750 REM SAVE USING CIO
755 COLOR 1:PLOT 144,83
757 POKE 764,12
760 OPEN #1,8,0,"C:"
770 POKE 850,11
800 REM SET UP IOCB AND CALL
810 POKE 852,PEEK(88):POKE 853,PEEK(89)
820 POKE 856,0:POKE 857,15
830 Q=USR(ADR(CIO$),16)
840 CLOSE #1
850 COLOR 0:PLOT 144,83:PLOT 144,75
860 IF LOD=0 THEN 880
865 REM ERASE MODEBARS AND REDRAW
870 PLOT 142,0:DRAWTO 142,95:PLOT 143,0:DRAWTO 143,95:GOSUB 1000
880 COLOR FENCOLOR:GOTO 950:REM RESUME PRE-SAVE OR PRE-LOAD MODE
890 REM *** MOD R ***
900 REM CLEAR SCREEN
910 GOSUB 1000:REM SET-UP
930 REM *** MOD D ***
950 REM RESUME PREVIOUS (OR INITIALIZED) MODE
970 ON (PENMODE+1) GOTO 6000,150,300,350,400
980 REM *** MOD C ***
1000 REM SET UP ROUTINE
1010 GRAPHICS 23+LOD*32:COLOR 1:SETCOLOR 4,0,8:LOD=0
1015 SETCOLOR 0,4,14:SETCOLOR 1,12,14:SETCOLOR 2,9,14
1020 PLOT 141,0:DRAWTO 141,95:DRAWTO 0,95:DRAWTO 0,0:DRAWTO 141,0
1025 RESTORE :FOR X=0 TO 12:READ F:PLOT 142,F:DRAWTO 159,F:NEXT X
1030 PLOT 144,0:POSITION 144,7:POKE 765,1:XIO 18,#6,12,0,"S:"
1040 PLOT 150,9:DRAWTO 148,9:DRAWTO 148,13:DRAWTO 150,13:PLOT 149,11
1050 PLOT 150,17:DRAWTO 148,17:DRAWTO 148,21:PLOT 149,19
1060 PLOT 146,27:DRAWTO 152,27
1070 PLOT 149,35
1080 PLOT 146,41:DRAWTO 146,45:DRAWTO 152,45:DRAWTO 152,41:DRAWTO 146,41
1090 PLOT 152,53:DRAWTO 152,49:DRAWTO 146,49:POSITION 146,53:POKE 765,1:XIO 18,#
6,0,0,"S:"
1100 COLOR 2:PLOT 152,61:DRAWTO 152,57:DRAWTO 146,57:POSITION 146,61:POKE 765,2:
XIO 18,#6,0,0,"S:"
1110 COLOR 3:PLOT 152,69:DRAWTO 152,65:DRAWTO 146,65:POSITION 146,69:POKE 765,3:
XIO 18,#6,0,0,"S:"
1120 PLOT 148,73:DRAWTO 148,77:DRAWTO 150,77
1130 PLOT 150,81:DRAWTO 148,81:DRAWTO 148,83:DRAWTO 150,83:DRAWTO 150,85:DRAWTO
148,85
1160 PLOT 148,89:DRAWTO 146,89:DRAWTO 146,93:DRAWTO 148,93
1170 PLOT 150,89:DRAWTO 150,93:DRAWTO 152,93
1171 NPEN=PENMODE:GOSUB 3140:REM DRAW MODE INDICATOR BAR
1180 NCOLR=FENCOLOR:GOSUB 3240:REM DRAW COLOR INDICATOR BAR
1185 REM INITIALIZE CIO$ MACHINE LANG CIO CALL
1190 FOR X=1 TO 7:READ Y:CIO$(X,X)=CHR$(Y):NEXT X
1200 RETURN :REM FROM SET-UP
1210 REM *** MOD I ***
2000 REM GETPOS - RETURN PEN X,Y COORDINATE
2010 IF PEEK(632)=15 THEN 2010
2020 Q=USR(PENBUF)
2040 X=PEEK(206):Y=PEEK(207)
2045 IF X>159 OR X<0 OR Y>95 OR Y<0 THEN 2010
2048 IF X>140 THEN POP :GOTO 100:REM USER WANTS NEW MODE
2050 RETURN
2060 REM *** MOD K ***
3000 REM SUBROUTINE TO MOVE MODE BARS
3010 IF Y<40 THEN 3100:REM MOVE PEN MODE INDICATOR
3020 IF Y<72 THEN 3200:REM MOVE COLOR INDICATOR
3030 RETURN :REM COMMANDS LOAD,SAVE,CLEAR HAVE NO MODE BAR
3100 REM CHANGE PEN MODE BAR
3110 COLOR 0:REM SET TO ERASE
3120 PLOT 142,PENMODE*8+(PENMODE=0):DRAWTO 142,PENMODE*8+6
3125 PLOT 143,PENMODE*8+(PENMODE=0):DRAWTO 143,PENMODE*8+6
3130 PENMODE=NPEN:REM SET TO NEW PEN MODE
3140 COLOR 1
```

```
3150 PLOT 142,PENMODE*8:DRAWTO 142,PENMODE*8+6
3155 PLOT 143,PENMODE*8:DRAWTO 143,PENMODE*8+6
3157 COLOR PENCOLR:REM RESTORE ORG COLOR
3160 RETURN
3200 REM MOVE COLOR MODE INDICATOR BAR
3210 COLOR 0:REM SET TO ERASE
3220 P=PENCOLR*8+40
3230 PLOT 142,P:DRAWTO 142,P+6
3235 PLOT 143,P:DRAWTO 143,P+6
3240 COLOR 1:P=NCOLR*8+40:REM NEW BAR
3250 PLOT 142,P:DRAWTO 142,P+6
3255 PLOT 143,P:DRAWTO 143,P+6
3258 COLOR PENCOLR:REM RESTORE ORG. COLOR
3260 RETURN
6000 REM *** USER PROGRAMMABLE MODULE ***
6010 REM RIGHT NOW IT ONLY READS PEN COORDINATES
6011 REM TO MAKE IT DO SOMETHING, JUST USE VARIABLES X,Y
6012 REM HOWEVER YOU LIKE BETWEEN LINES 6020 AND 6900
6020 GOSUB 2000
6900 GOTO 6020
7999 REM MENU FIELD DATA
8000 DATA 0,7,15,23,31,39,47,55,63,71,79,87,95
8499 REM CIO CALL ROUTINE DATA
8500 DATA 104,104,104,170,76,86,228
8900 REM MACH LANG ROUTINE
9000 REM "LOADER" LOADS "PENBUF" INTO PAGE 6
9010 RESTORE 9020:FOR X=1536 TO 1652:READ P:POKE X,P:NEXT X:RETURN
9020 DATA 104,162,0,160,6,132,204,173,12,212,201,10,144,3,56,233,228
9030 DATA 24,105,154,160,117,132,203,32,86,6,165,203,24,101,205,133
9040 DATA 203,173,13,212,56,233,16,32,86,6,232,228,205,240,9,165
9050 DATA 20,197,20,240,252,76,7,6,165,205,74,168,177,203,133,207
9060 DATA 169,117,133,203,177,203,133,206,165,206,74,74,133,208,165,207
9070 DATA 74,74,133,209,96,134,206,164,206,192,0,240,20,136,209,203
9080 DATA 200,176,14,136,133,206,177,203,200,145,203,136,165,206,56,176
9090 DATA 232,145,203,96
9100 END
```

The 16K demonstration program is divided into several modules. A description of the function of each of them is provided following a general structural description of the program.

### General Structural Description of PALLET

The PALLET flowchart (Figure 5) illustrates the basic structure of the 16K version of the program. At the program start, the modes are initialized. PENMODE is set equal to 3, which puts the pen in DRAW mode, and PENCOLR is set to 3, which makes the line drawn blue. DRAW module (Mod G) is then entered via RESUMODE (Mod D). The four drawing modes (E, F, G, H) are exited when the pen is positioned in the menu area. The menu position test is performed at Module I, the routine that gets X, Y coordinates. If the pen is in the menu section, GETMODE (Mod J) is executed. This translates the pen position into the mode desired, and then goes there. If the selection is a change color command or a function command, it is performed and the previous drawing mode is resumed via RESUMODE. If the command is a change drawing mode, the new mode is entered immediately.

There are five possible drawing modes. The number representing the current mode is stored in the variable PENMODE. There are, in addition, four possible color modes. The number representing the current color is stored in the variable PENCOLR. Modes are indicated on the screen by two independent mode indicator bars in the menu field of the current modes. Mode numbers are assigned as follows:

DRAWING MODE	PENMODE=	COLOR	PENCOLR=
DOT	4	Blue	3
DRAW	3	Green	2
FILL	2	Red	1
ELASTIC	1	ERASE (set to background color)	0
USER mode	0		

Figure 7 Color Mode Numbers

### MODULE A (LINE 20)

"Start" calls Module B, initializes mode to DRAW and color to blue, calls Module C, then goes to RESUME MODE Module D to begin actual user input.

MODULE B (LINE 9000)

Identical to PENBUF loader described in Section 6. Because of memory limitation, the loader for the BK version is different than the loader discussed in Section 6 (see Appendix G).

MODULE C (LINE 1000)

SETUP initializes color hues (lines 1010, 1015), clears screen (by the GRAPHICS command), marks borders of screen (line 1020) draws menu field separators (line 1025), then draws each menu symbol. Each statement (1060-1130) corresponds to a new symbol, and is in the order in which each symbol appears on the screen. Then using two mode indicators, PENMODE and PENCOLOR, it draws the two mode indicator bars (lines 1171, 1180). This is accomplished by entering the second half of the Move Modebar (Module K) subroutine, which is the part that draws the new bar. SETUP then executes a BASIC RETURN either to Module A, Module P, or Module R, which are the only calling routines.

MODULE D (LINE 950)

Use RESUMODE to reenter a drawing mode once it has been exited to either change the color mode or to perform one of the three functions. The variable PENMODE indicates which mode to enter. The color modes and drawing modes are explained earlier in this section.

MODULE E (LINE 150)

Enter ELASTIC mode, which is drawing mode 1 (PENMODE=1). ELASTIC mode first gets an origin or pivot point (defined by the coordinates XP, YP) and then enters a loop in which the old line is erased (line 190) and a line to the new point is drawn (line 210). As long as the pen is down, this loop is repeated (line 230). If the pen is lifted and then set down again, a new pivot point is defined, and the loop entered (line 240). As with all the drawing modes, it exits at line 2048 of the GETPOS call (Module I).

MODULE F (LINE 300)

Enter FILL mode, drawing mode 2 (PENMODE=2). First a preliminary point is fetched and plotted. A loop is then entered which does a FILL command (see ATARI BASIC Reference Manual for more information on this command) from the last point plotted to the current pen position. As with all the drawing modes, it is exited at 2048 in Module I.

MODULE G (LINE 350)

Enter DRAW mode, drawing mode 3 (PENMODE=3). A preliminary point is fetched (line 352) and a loop consisting of getting coordinates and drawing to them is executed. Note that if line 352 is deleted, a point will be fetched (line 360) and a BASIC DRAWTO will be performed to that point. But the DRAWTO draws from the last pixel plotted on the screen. So if line 352 is deleted, the last point plotted will be the mode indicator bar and line 380 will connect the current pen position to the mode bar.

MODULE H (LINE 400)

Enter DOT mode, drawing mode 4 (PENMODE=4). This is the most simple drawing mode. A loop is entered which plots each X, Y position.

MODULE I (LINE 2000)

GETPOS waits until the pen tip is pressed (line 2010), calls the PENBUF machine language routine (line 2020), and retrieves the X and Y values (line 2040). If the pen is out of GRAPHICS Mode 23 (GRAPHICS 7+16 to eliminate the text window) boundaries, the above process is repeated (line 2005). If the pen is in the menu section of the video screen, the BASIC return address is removed from the stack (via the POP command) so a proper GOTO can be performed (line 2048) since GETPOS is called by a GOSUB. In this case, control is transferred to Module J. If the pen is not in the menu section, execution returns to the calling routine, with the variable X and Y containing the latest X and Y positions read from the pen registers, representing the current pen position.

MODULE J (LINE 100)

GETMODE is entered only from Module I, and only when the pen is in the menu section of the screen (line 2048). It first pokes location 77 to stop the operating system from entering the automatic monitor-preserving "attract mode" (line 105). It assumes the user will make a menu selection at least once each 9 minutes. If the attract mode ever begins, just place the pen in the menu section to restore normal activity. (Attract mode is the mode you see after leaving your computer for awhile, when the colors start changing on your television screen.)

After line 105, Module J converts the Y coordinate into the correct menu selection (line 110). Module K is called to move the mode indicator bars (line 120). The program waits at line 125 for the pen to be lifted to prevent entering GETMODE over and over if the pen is held in the menu area. Then activity is directed to the module indicated by the pen location (line 130).

MODULE K (LINE 3000)

This routine moves mode bars from an old location to the newly indicated mode location. The mode indicator bars appear to the left of each selection item in the menu area, only when that mode is active. This module is called from Module C (initialize them) or Module J (move them to a new spot). The routine first decides which bar is selected (color or drawing mode) by looking at the Y coordinates of the pen in the menu area (lines 3010, 3020). No bars are associated with Load, Save, or Clear so it will return with no action if those functions are selected (line 3030). If the new mode is a drawing mode, the bar is moved from mode PENMODE to mode NPEN by erasing the old bar (lines 3110-3125), setting the new PENMODE (line 3130), and drawing the new bar (lines 3140-3157). If the new mode is a color mode, the same method is used. When Module K is called as a subroutine from Module C, the entry point is at that section which draws the new bar. This is because there is no old bar to erase.

MODULE L (LINE 450)

Changes the color of the pen to the background color, BASIC's COLOR 0 command, and sets the color mode identifier, PENCOLR, to 0. This is used to erase.

MODULE M (LINE 500)

Changes pen color to red using BASIC COLOR 1 command, and sets color mode identifier, PENCOLR, to mode 1.

MODULE N (LINE 550)

Changes pen color to green using BASIC COLOR 2 command, and sets the color mode identifier, PENCOLR, to mode 2.

MODULE O (LINE 600)

Changes pen color to blue using BASIC COLOR 3 command and sets the color mode identifier, PENCOLR, to mode 3.

MODULE P (LINE 700)

LOAD first plots the point indicating that a load is taking place. Line 707 simulates a carriage return, which is needed for a cassette load. If you are using diskette, delete line 707. Then in line 710, the file is opened. Line 720 sets the CIO command call to 7, which is a "get characters" command. Control is transferred to the second section at line 800. This section finishes setting up the IQCB and

## ATARI Light Pen Operator's Manual

---

then calls CIO. Line 800 sets the start address to the beginning of the screen display, and line 820 sets the size of the display memory. Thus, when CIO is called, it loads "characters" (bytes) from diskette and puts them into the beginning of the screen memory.

Line 830 calls CIO using a tiny machine language routine stored in CIO\$. The routine is as follows:

```
PLA      ;pop # of parameters
PLA      ;pop low byte of parameter
PLA      ;pop high byte into A
TAX      ;move A to X registers
JMP     CIO ;go to CIO
```

The X register must contain the IOCB number multiplied by 16. We are using IOCB 1 (i.e., OPEN #1) so we pass the integer 16 to the stack, which is transferred to X. Line 850 erases the plotted dot. Line 870 erases the modebars (which were loaded off diskette and are probably incorrect), and then calls SETUP to redraw the menu. Note that SETUP will not erase because LOD=1 (load flag was set in line 720).

### MODULE Q (LINE 750)

The SAVE function first plots a point to indicate that it is saving, then simulates a carriage return for the cassette save in line 757. If you are using diskette, delete this line. Line 760 opens the file for writing, and line 770 sets the CIO command to 11, "Put Characters." Line 810 sets the start of the CIO read to the start of the screen memory. Line 820 sets the number of bytes to be saved.

The CIO is called at line 830 (see Module P description) and closed at line 840. Line 850 erases the indicator dots, and line 880 calls RESUMODE Module D to reenter the drawing mode.

### MODULE R (LINE 900)

CLEAR SCREEN calls the SETUP routine to blank the entire screen, set up the menu again and enter the previous modes.

Beginning with Section 5, the manual will take you through a step-by-step introduction to programming with the ATARI Light Pen. A series of programming examples is presented in increasing order of complexity. The appendices give helpful hints to the advanced programmer and detailed insights into the programs presented.



SECTION 5 - PROGRAMMING EXAMPLE 1:Using the Light Pen with ATARI BASIC

This program demonstrates the ability of the computer to sense the position of the light pen on the screen. It will plot a dot of light each time you place the pen against the screen. The dots will be placed exactly where the pen was resting.

Type in the following program, then RUN it.

```
1 REM DEMO PROGRAM #1, LIGHT PEN WITHOUT "PENBUF"  
10 GRAPHICS 23:SETCOLOR 4,12,8:SETCOLOR 0,7,14:COLOR 1  
20 IF PEEK(632)=15 THEN 20  
30 Q=PEEK(54284)  
40 X=Q+154-228*(Q>10)  
50 Y=PEEK(54285)-16  
60 PLOT X,Y  
70 GOTO 20  
80 END
```

Figure 8 Programming Example 1: Using the Light Pen with ATARI BASIC

Through the use of Graphics Mode 7 this program produces a bright green screen. This screen is actually a set of 160 (horizontal) by 96 (vertical) squares, each of which is colored green. Push the light pen against the television screen somewhere near the center and see what happens. A blue dot should appear, opposite the pen tip, on the screen. (If nothing happens, turn to Appendix C, Troubleshooting.)

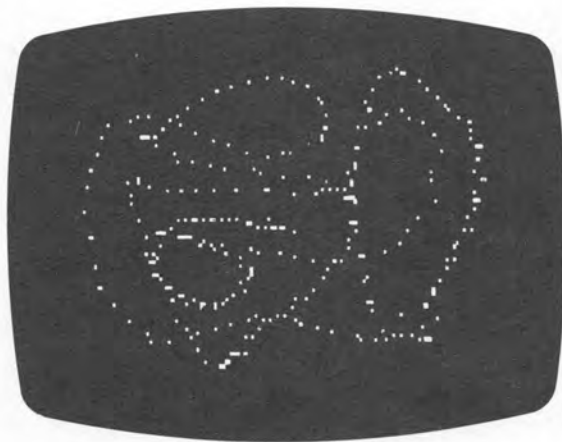


Figure 9 Television Screen Display for Programming Example 1

Try other points on the screen. More points will be plotted as long as you stay within the boundaries of Graphics Mode 23.

NOTE: If you place the pen outside of the GR.23 central boundaries, you will get an "ERROR 3 on LINE 10." Press [S/RESET]. Notice the area of the black border. This is the area the light pen cannot use even though this area is not visible in Graphics Mode 23. To mark these boundaries by drawing a border, add line:

```
15 PLOT 0,0:DRAWTO 159,0:DRAWTO 159,95:DRAWTO 0,95:DRAWTO 0,0
```

Now when you press the light pen to the screen, stay inside the boundaries. You could use a TRAP statement at line 55 as follows:

```
55 TRAP 20
```

to prevent an error. If the PLOT statement results in an error, the program will resume operation at line 20.

Programming Example 1 may also be used with Graphics Mode 7; however, you cannot plot in the text window area at the bottom of the television screen.

#### How Programming Example 1 Works

At line 10, the background color is set and the color used for the plotting is established. Refer to Tables 9.3 and 9.5 of the ATARI

BASIC Reference Manual for color and hue information. A relatively bright background color level has been chosen to assure that the light pen will respond easily. You may experiment by changing the brightness (luminance) and hue of the background as well as varying the brightness control of your television set to determine the minimum levels to which your light pen will respond (see Appendix B).

At line 20, the program looks at location 632. If it contains a decimal 15, it means the light pen is not pressed against the screen. If it contains a decimal 14, the light pen is pressed against the screen. The application of the light pen to the screen closes a switch that is built into the tip. Until the switch is closed, the computer stays in a wait loop at line 20.

At line 30, the computer reads an internal (X) hardware register which has been activated by the light pen. Line 40 adjusts this value to be compatible with Graphics Mode 23 plotting coordinates (range 0-159).

Line 50 causes a read of the Y hardware register and adjusts the value of Y to Graphics Mode 23 (range 0-95).

Line 60 plots the point found in the adjusted hardware registers, the point currently under the tip of the light pen. Then the program returns to the wait loop at 20 until the next switch closing.

#### Experimenting with Programming Example 1

Each time you move the light pen along the screen, you leave a trail of dots behind. Place the light pen in one spot, holding it perpendicular to the screen. Now at that same spot, aim the pen at different angles other than perpendicular. Notice that spots surrounding the one the pen sits on are being plotted also. The pen is detecting light from other parts of the screen (see Appendix B).

Move the pen slowly to draw figures on the screen. Occasionally, a dot will appear at a random location on the screen even though the pen is not near that area. This is caused by random noise from the television or pen. Section 6 shows how to prevent noise from affecting your programs.



SECTION 6 - PROGRAMMING EXAMPLE 2:Using PENBUF, a Light Pen Utility Program

To minimize noise and to preadjust the X and Y values for you, ATARI has developed a machine language program called PENBUF. A BASIC program to load PENBUF into the correct memory locations is presented in Figure 10. The use of a language other than BASIC will work only if its memory is similarly structured. Primarily, Page 6 and locations \$CB - \$D1 must be free.

PENBUF has two functions: To take the statistical median of a number of X, Y readings and to calculate the true Graphics equivalent positions for X and Y as in lines 40 and 50 of Programming Example 1. For details on how PENBUF works, see Appendices D, E, and F.

```
9000 REM "LOADER" LOADS "PENBUF" INTO PAGE 6
9010 RESTORE 9020:FOR X=1536 TO 1652:READ P:POKE X,P:NEXT X:RETURN
9020 DATA 104,162,0,160,6,132,204,173,12,212,201,10,144,3,56,233,228
9030 DATA 24,105,154,160,117,132,203,32,86,6,165,203,24,101,205,133
9040 DATA 203,173,13,212,56,233,16,32,86,6,232,228,205,240,9,165
9050 DATA 20,197,20,240,252,76,7,6,165,205,74,168,177,203,133,207
9060 DATA 169,117,133,203,177,203,133,206,165,206,74,74,133,208,165,207
9070 DATA 74,74,133,209,96,134,206,164,206,192,0,240,20,136,209,203
9080 DATA 200,176,14,136,133,206,177,203,200,145,203,136,165,206,56,176
9090 DATA 232,145,203,96
```

Figure 10 PENBUF Loader Program Listing

The data statements represent the decimal equivalent of the machine language program. The assembly language listing is presented in Appendix E. The PENBUF loader is to be appended to a BASIC program and used as a subroutine.

By calculating the statistical median of a number of readings, a random noise pulse is less likely to be accepted by the program as the indicator of the exact pen position. The median calculation, especially useful in rejecting noise, is discussed in Appendix F. The user can control the number of readings to be taken by changing the number in location 205 (decimal). Location 205 is an input parameter for PENBUF. The routine will not work if this is not POKEd with a number. The range of readings that may be taken is variable from 1 (min) to 69 (max). The number selected should be odd. The more readings that are taken, the less noise will affect the pen; however, it will not respond as quickly. A minimum of three readings normally provides a good response. See Appendix F for a more detailed discussion.

## ATARI Light Pen Operator's Manual

---

The calculation of X and Y coordinates is done by PENBUF not only for Graphics Mode 23, but also for Graphics Mode 0. After executing PENBUF, the Graphics 23 values for X and Y may be obtained from locations 206 (X) and 207 (Y). If you are working in Graphics 0, the positional values for X and Y are found in locations 208 (X) and 209 (Y). By peeking at those locations, the program can determine the coordinates of the pen tip. (Refer to Appendices D and E for the PENBUF flow chart and assembly language listing.)

Type in and RUN the following program:

```
10 REM DEMO PROGRAM, SEC 1.3.5, LIGHT PEN IN GRAPHICS 23
30 GRAPHICS 23:COLOR 2:SETCOLOR 4,0,12:REM LUMINANCE 12 BACKGROUND
40 GOSUB 9000:PENBUF=1536:POKE 205,3
50 REM IS PEN DOWN NOW?
60 PRESSED=(PEEK(632)=15)
70 IF NOT PRESSED THEN 50
80 REM PEN IS PRESSED, SO GO GET A COORDINATE
90 JUNK=USR(PENBUF)
100 X=PEEK(206)
110 Y=PEEK(207)
120 REM PLOT IT
130 PLOT X,Y
140 GOTO 50
9000 REM "LOADER" LOADS "PENBUF" INTO PAGE 6
9010 RESTORE 9020:FOR X=1536 TO 1652:READ P:POKE X,P:NEXT X:RETURN
9020 DATA 104,162,0,160,6,132,204,173,12,212,201,10,144,3,56,233,228
9030 DATA 24,105,154,160,117,132,203,32,86,6,165,203,24,101,205,133
9040 DATA 203,173,13,212,56,233,16,32,86,6,232,228,205,240,9,165
9050 DATA 20,197,20,240,252,76,7,6,165,205,74,168,177,203,133,207
9060 DATA 169,117,133,203,177,203,133,206,165,206,74,74,133,208,165,207
9070 DATA 74,74,133,209,96,134,206,164,206,192,0,240,20,136,209,203
9080 DATA 200,176,14,136,133,206,177,203,200,145,203,136,165,206,56,176
9090 DATA 232,145,203,96
```

Figure 11 Programming Example 2: Using PENBUF

Before you RUN the program, refer to the Note accompanying Programming Example 1. If you go outside the boundaries of Graphics 23, you will get an ERROR 141 on LINE 30. To see the boundary, type the suggested line 15 of Programming Example 1 as line 35 of this program.

When you RUN the program, the light pen will work exactly as in Programming Example 1. However, the noise spots should now be missing from the display. The differences between Programming Examples 1 and 2 are as follows:

Line 40 uses a logical (true/false) compare equivalent to line 20 in Programming Example 1. Lines 50-70 loads PENBUF into the proper area, defines the entry point to the USR function and sets the number of pen

location readings to be sampled. Line 90 calls the machine language routine PENBUF described above, calculating the true X and Y for future use. Lines 100 and 110 retrieve the X and Y values with no further calculations required. Line 130 plots the point, then the program loops back to the pen-read routine again. As mentioned earlier, the PENBUF loader is to be used as a subroutine at statement 9000-9090.

As specified earlier, PENBUF may be used from Graphics Mode 0 also. Figure 12 is the same program as Programming Example 2., modified to run in Graphics 0 instead.

```

10 REM DEMO PROGRAM #2, LIGHT PEN IN GRAPHICS 0 (STANDARD TEXT MODE)
30 GRAPHICS 0:COLOR 2:SETCOLOR 2,0,9:REM LUMINANCE 9 BACKGROUND
40 GOSUB 9000:PENBUF=1536:POKE 205,5
45 POKE 752,1
50 REM IS PEN DOWN NOW?
60 PRESSED=(PEEK(632)=14)
70 IF NOT PRESSED THEN 50
80 REM PEN IS PRESSED, SO GO GET A COORDINATE
90 JUNK=USR(PENBUF)
100 X=PEEK(208)
110 Y=PEEK(209)
120 REM PLOT IT
130 POSITION X,Y:PRINT "■"
140 GOTO 50
9000 REM "LOADER" LOADS "PENBUF" INTO PAGE 6
9010 RESTORE 9020:FOR X=1536 TO 1652:READ P:POKE X,P:NEXT X:RETURN
9020 DATA 104,162,0,160,6,132,204,173,12,212,201,10,144,3,56,233,228
9030 DATA 24,105,154,160,117,132,203,32,86,6,165,203,24,101,205,133
9040 DATA 203,173,13,212,56,233,16,32,86,6,232,228,205,240,9,165
9050 DATA 20,197,20,240,252,76,7,6,165,205,74,168,177,203,133,207
9060 DATA 169,117,133,203,177,203,133,206,165,206,74,74,133,208,165,207
9070 DATA 74,74,133,209,96,134,206,164,206,192,0,240,20,136,209,203
9080 DATA 200,176,14,136,133,206,177,203,200,145,203,136,165,206,56,176
9090 DATA 232,145,203,96

```

Figure 12 Programming Example 2 Modified to Run in GRAPHICS 0

The only differences are:

- o Line 30 changes graphics mode.
- o Line 45 makes the cursor vanish.
- o Lines 100 and 110 use locations 208 and 209 instead of 206 and 207, to find the GR.0 coordinates.
- o Line 130 is a POSITION, PRINT combination because PLOT does not work properly in GR.0. Another advantage to this

combination is that you are not limited to a single character print.

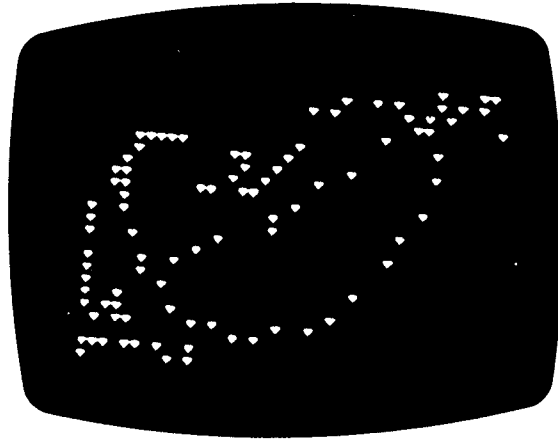


Figure 13 Television Screen Display for Programming Example 2



SECTION 7 - PROGRAMMING EXAMPLE 3: A Keyboard Substitute

This program illustrates a way to substitute the light pen for the keyboard. It produces the complete alphabet and selected punctuation characters across the top of the screen. When you type in and RUN the program, pointing to any letter with the pen will cause that letter to be typed on the screen at the current cursor location. The last two characters, b and r, are added at the end of the character sequence and stand for backspace and carriage return. See Figure 14 for the program listing.

```
1 REM DEMO #3, A KEYBOARD SUBSTITUTE
10 DIM A$(40):POKE 82,0
20 GRAPHICS 0:SETCOLOR 2,0,8:SETCOLOR 1,0,14
30 PENBUF=1536:POKE 205,15:GOSUB 9000
40 A$="ABCDEFGHJKLMNOPQRSTUVWXYZ .,:;!()?`'br"
50 FOR Q=1 TO LEN(A$):PRINT A$(Q,Q);" ";:NEXT Q
60 IF PEEK(632)=15 THEN 60
70 JUNK=USR(PENBUF)
80 PC=PEEK(208):PR=PEEK(209)
90 IF PR>2 THEN 60
95 SUBSC=(40*PR+PC)/3+1
100 SUBSC=INT(SUBSC+0.5)
110 IF SUBSC<39 THEN PRINT A$(SUBSC,SUBSC);:GOTO 300
200 IF SUBSC=39 THEN PRINT CHR$(126);:GOTO 300:REM BACKSPACE CHARACTER
210 PRINT
300 IF PEEK(632)=14 THEN 300
310 GOTO 60
9000 REM "LOADER" LOADS "PENBUF" INTO PAGE 6
9010 RESTORE 9020:FOR X=1536 TO 1652:READ P:POKE X,P:NEXT X:RETURN
9020 DATA 104,162,0,160,6,132,204,173,12,212,201,10,144,3,56,233,228
9030 DATA 24,105,154,160,117,132,203,32,86,6,165,203,24,101,205,133
9040 DATA 203,173,13,212,56,233,16,32,86,6,232,228,205,240,9,165
9050 DATA 20,197,20,240,252,76,7,6,165,205,74,168,177,203,133,207
9060 DATA 169,117,133,203,177,203,133,206,165,206,74,74,133,208,165,207
9070 DATA 74,74,133,209,96,134,206,164,206,192,0,240,20,136,209,203
9080 DATA 200,176,14,136,133,206,177,203,200,145,203,136,165,206,56,176
9090 DATA 232,145,203,96
```

Figure 14 Programming Example 3: A Keyboard Substitute



Figure 15 Television Screen Display for Programming Example 3

The program begins by allocating space for the character string and presetting the left margin to zero. At line 20, the screen color and letter color are set. Line 30 defines the start of PENBUF, sets the number of position reads for which a median is to be calculated, and loads PENBUF into memory.

The letter menu is printed on the screen by lines 40 and 50. Then the program waits in a loop at line 60 until the pen switch is pressed. As with the previous demos, line 70 uses PENBUF to interpret the pen location. Because this is Graphics 0, the X and Y locations are found from locations 208 and 209.

Since only the first three rows (numbers 0, 1, 2) actually contain the characters, the program ignores any pen location in a row greater than 2. If the row number is valid, it transforms the row and column data in a positional subscript into the character string. The character is then printed on the screen.

If the character was either the backspace or return, the function selected is performed. In any case, the computer will loop back to line 60 again, waiting for a further light pen input.

SECTION 8 - PROGRAMMING EXAMPLE 4: A Teaching Aid

The program in Figure 16 demonstrates a multiple-choice test. Type in and RUN the program.

```

10 REM DEMO #4, PEN AS A TEACHING AID (MENU SELECTOR)
15 DIM A$(10)
20 GRAPHICS 0:SETCOLOR 2,3,8:SETCOLOR 1,0,14
25 GOSUB 9000:PENBUF=1536:POKE 205,15:POKE 752,1
30 POSITION 8,10:PRINT "HOW MANY DIGITS?"
40 POSITION 12,12:PRINT "1 2 3 4 5"
50 IF PEEK(632)=15 THEN 50
60 JUNK=USR(PENBUF)
70 C=PEEK(208):R=PEEK(209)
80 IF R<>12 THEN 50
90 LENG=INT((C-8)/3):IF (LENG<1) OR (LENG>5) THEN 50
100 X=INT((9*RND(0)+1)*10^(LENG-1))
110 Y=INT((9*RND(0)+1)*10^(LENG-1))
120 PRINT CHR$(125):POSITION 16,8:PRINT X
130 POSITION 14,9:PRINT "+ ";Y
140 POSITION 14,10:PRINT "_ ";:FOR Q=1 TO LENG:PRINT "_";:NEXT Q
150 P=INT(10*RND(0)+1)
170 FOR Q=1 TO 10:IF (Q=P) THEN A=X+Y:GOTO 190
180 A=INT((18*RND(0)+2)*10^(LENG-1))
190 POSITION 15*(Q>5)+10,2*(Q-5*(Q>5))+11
200 A$=STR$(A):FOR Z=1 TO LEN(A$):PRINT CHR$(ASC(A$(Z,Z))+128):;NEXT Z
210 NEXT Q
250 IF PEEK(632)=15 THEN 250
260 JUNK=USR(PENBUF):C=PEEK(208):R=(PEEK(209)-11)/2
270 IF (INT(R)<>R) OR (R<1) OR (R>5) THEN 250
275 POSITION 12,5
280 IF (C>20) THEN R=R+5
290 IF (R=P) THEN PRINT "CORRECT!!!":GOTO 100
300 PRINT "NOT QUITE..."
310 FOR Z=1 TO 200:NEXT Z
320 POSITION 12,5:PRINT " " "":GOTO 250
9000 REM "LOADER" LOADS "PENBUF" INTO PAGE 6
9010 RESTORE 9020:FOR X=1536 TO 1652:READ P:POKE X,P:NEXT X:RETURN
9020 DATA 104,162,0,160,6,132,204,173,12,212,201,10,144,3,56,233,228
9030 DATA 24,105,154,160,117,132,203,32,86,6,165,203,24,101,205,133
9040 DATA 203,173,13,212,56,233,16,32,86,6,232,228,205,240,9,165
9050 DATA 20,197,20,240,252,76,7,6,165,205,74,168,177,203,133,207
9060 DATA 169,117,133,203,177,203,133,206,165,206,74,74,133,208,165,207
9070 DATA 74,74,133,209,96,134,206,164,206,192,0,240,20,136,209,203
9080 DATA 200,176,14,136,133,206,177,203,200,145,203,136,165,206,56,176
9090 DATA 232,145,203,96

```

NOTE: The characters in the quotes in line 140 should be occupied by "CTRL N" characters. It is not shown since the printer which produced this listing has no such character in its character set.

Figure 16 Programming Example 4: A Teaching Aid

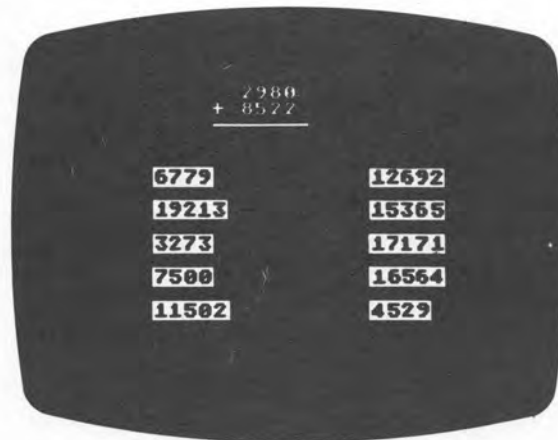


Figure 17 Television Screen Display for Programming Example 4

The program will ask you how many digits to use for an arithmetic problem. Then it will display a pair of numbers, each with the selected number of digits, as an addition problem. There are ten choices of answers to this problem but only one of the answers is correct.

Use the pen to point to the correct answer. If you select the wrong one, the computer will tell you it is not correct. When you point to the correct answer, a new problem will be presented.

By examining the program listing, you will notice that the light pen is used as an input for two separate parts of the program, once at lines 50-60 and again at lines 250-260.

Initial setup takes place in lines 20-50. It establishes colors, sets the number of median reads to 15, and turns off the cursor. The initial read of the light pen registers requires that the Y value be a 12. This means you are pointing to one of the characters in the row representing how many digits are to be used for the test. This is done by line 80.

Lines 100-140 calculate and print the digits for the arithmetic problem. There will be 10 possible answers to the problem. Line 150 selects which of the 10 positions will be occupied by the correct answer. A random number is selected for each of the 10 positions, but line 170 bypasses the random number and places the correct answer there instead.

Line 200 prints each answer in inverse video (with dark characters on a bright background) so that the pen will be able to respond to the

light emitted by the characters during its selection attempt. Lines 250-320 test the pen input, looking for the correct response, print an appropriate message to the user, then return either to request another attempt (to line 250) or to calculate a new problem (line 100). The incorrect response loop includes a do-nothing delay loop at line 310. This device effectively highlights particular characters because they retain their luminance against the dark background.

The program concludes with the PENBUF loader at lines 9000-9090. Appendix G of this manual explains how to save some time and space by using the PENBUF loader from the 8K PALLET demo program.

ATARI Light Pen Operator's Manual

SECTION 9 - PERIODIC MAINTENANCE

The light pen has a window in the tip that allows light from the television screen to strike a phototransistor. Dust can accumulate in this area. This area will have to be cleaned with a soft cloth occasionally to maintain the sensitivity of the pen to incoming light. Do not use paper towels for this purpose because they will scratch the window. In addition, you should take the following precautions to maintain your pen:

- o Keep your pen in a cool, dry area.
- o Do NOT use detergents or cleaners on the pen.
- o Do NOT use rough or sharp objects to clean the window.





APPENDIX A - BASICS OF LIGHT PEN OPERATION

The ATARI Personal Computer System produces 60 "frames" (pictures) each second on the television screen. Each of these 60 individual still pictures is produced by a beam of electrons sweeping across a light-sensitive phosphor on the inner surface of the screen. When this beam hits the surface, that section of the surface which is being swept lights up. (Only when the beam is there does it light, and the light dies down very quickly once the beam passes.) Refer to Figure A.1 for a simplified sketch of this picture-producing activity.

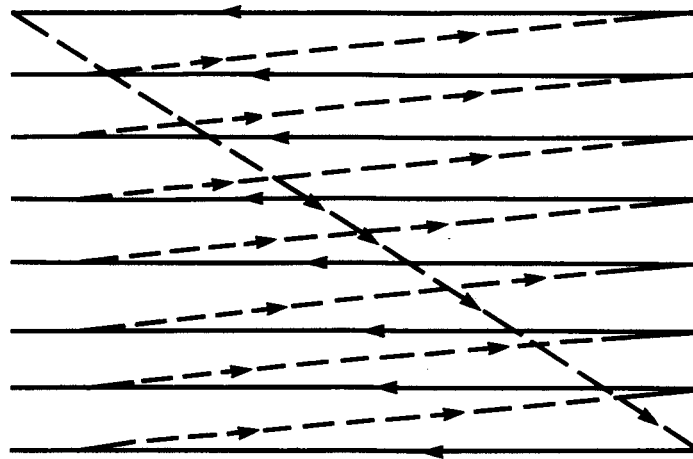


Figure A.1 Television Screen Simulation

Only a very few lines are shown here for discussion purposes. (There are approximately 262 lines per screen scan in each 1/60 second.) The solid lines represent the time when the trace beam is visible (actively producing light on the screen). The dotted lines represent the retrace period when the beam is not visible, while it is returning to the left edge to begin a new scan line or to the upper-left corner to begin the scanning of the whole screen again.

The beam begins at the upper-left corner and goes left to right, retraces, moves down a little bit (1/262 of the total height of the screen) repeats that process until it gets to the lower right corner, then retraces to the upper-left corner and starts all over again.

Different colors are produced at different locations on the screen because the electron beam is being changed by the computer hardware. As it sweeps, the computer controls its hue and luminance (color and brightness) by reading a selected group (graphics block) of memory locations and determining which color is to be displayed at each specific physical location on the surface of the screen.

## ATARI Light Pen Operator's Manual

---

For each position on the screen, there is a specific memory location assigned in the RAM that controls what is displayed at that location. If you could freeze time, you could look at a counter inside the machine to determine exactly which horizontal and vertical memory location is currently being accessed.

This is where the ATARI Light Pen is useful. Let's take the original picture and add the pen to it.

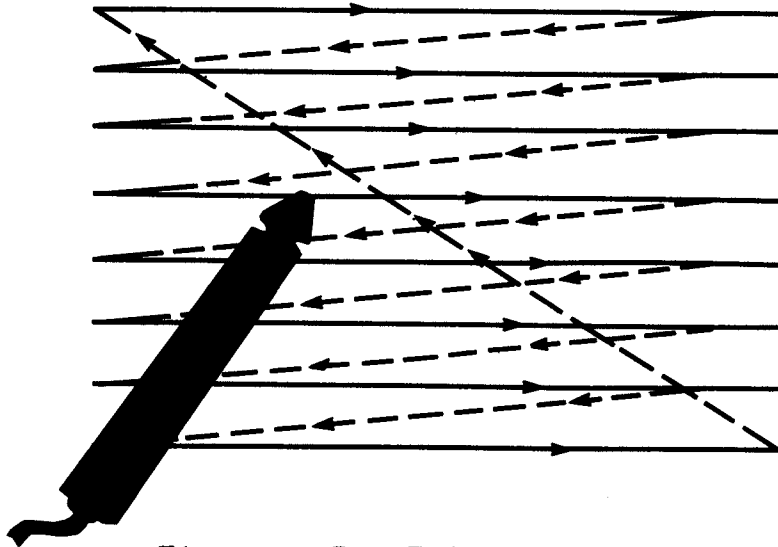


Figure A.2 Television Screen Plus Light Pen

If the pen is pressed to the surface of the screen, it will let in very little outside light. By looking at the scan method shown in Figure A.2 -- left to right, retrace, and so forth -- you will notice that the light pen can cover very few of the lines at one time. Since the screen surface remains lit for only a short time after the beam passes by, the light pen will only see a light pulse for a very small part of the total screen scan time.

When the light pulse is seen by the pen, the computer accesses the memory location that controls the specific area over which the pen happens to be resting.

The computer console contains special hardware that is connected to the light pen through the controller ports. This hardware allows the pen to capture the horizontal and vertical scan addresses when (and if) a light-present pulse is received from the pen. These captured addresses are stored in registers located at decimal 54284 (horizontal-hex \$D40C) and 54285 (vertical-hex \$D40D). These same values can also be accessed at two other memory locations. These are 564 and 565 (horizontal and vertical hex \$234 and \$235).

By peeking at these locations during program operation, you can determine exactly where the pen is resting on the screen. These values are changed 60 times each second if and when the pen is sitting on

an area of the screen bright enough to produce a light-pulse-present output from the pen.

Different light pens have different levels of sensitivity. As you go through the examples given in this manual, you may wish to experiment to determine the minimum level of sensitivity of your pen by varying the hue and luminance levels in certain of the statements.

The values obtained from locations 564 and 565 are taken directly from the scan counter inside the computer. Because of this, they do not match the desired physical screen positions for plotting and drawing or for sensing the position of a specific block on the screen. To use these values accurately, they must be adjusted to match the normal coordinate system; that is, values from zero to some maximum value in both the X and Y directions to relate to the graphics mode being used.

Location 564 will be found to contain values ranging from 74 to 227 or from 0 to 5 because of the scanning memory used for the screen. Location 565 will contain values between 16 and 111 for the same reason. To adjust these values to begin at zero and continue to a desired maximum, the following sequence can be used:

```

X = (PEEK(564))+154
IF (PEEK(564)<10 THEN X=X-128
Y = (PEEK(565))-16
    
```

This gives X values from 0-159 and Y values from 0-95. If you examine Table 9.1 of the ATARI BASIC Reference Manual, you will notice that these values correspond exactly to the block positions for GRAPHICS mode 23. This same number range will also be compatible with GRAPHICS mode 0 if you just divide each number by 4 (GRAPHICS 0 has a 40 by 24 display rather than 160 by 96, so division by 4 for any number obtained will relate to the same physical segment of the screen).



APPENDIX B - ATARI LIGHT PEN CHARACTERISTICS

The ATARI Light Pen has several traits that must be noted by advanced programmers in order to use it effectively.

The computer continuously reads pen values regardless of whether or not the tip switch is pressed. This switch is left for the programmer to interpret.

When the pen is plugged into controller jack 1, the switch may be checked by doing a PEEK of location 632 (decimal). For the other controller jacks, the locations are as follows:

Jack 2	PEEK (633)
Jack 3	PEEK (634)
Jack 4	PEEK (635).

If there is insufficient light on the pen, the position registers (54284-5) remain unchanged. This creates a condition that must be checked by the programmer. For example, a user might make a menu selection with the pen and when that function is finished, the menu is displayed again. If the user now accidentally presses the tip switch while the pen is not aimed at the monitor, the same X, Y coordinate will be read and the previously performed function will be repeated. Room light does not effect a change in pen values.

Pen sensitivities vary quite a bit. To determine the sensitivity of your particular ATARI Light Pen, enter the demo program listed as Programming Example 1. Try your pen at different background luminance levels (line 20). There will be a point at which your pen does not seem to respond. This is the limit of your pen. Also, some hues must have higher intensities in order to be read, and some may be read at lower intensities. For example, the pen is more sensitive to blue than to green and more sensitive to green than to red. At the middle of the screen, the pen tends to be more sensitive than near the corners.

Pen accuracy is directly proportional to screen intensity. You will find that at higher intensities the pen values don't drift very much, and pen noise is dramatically reduced. Also, if you are plotting points continuously and using a dark color, as the pen sits in one place, the surrounding pixels fill in. This darkens the section of screen below the pen, which causes the pen to become less accurate, drifting even further. If extreme accuracy is needed, the solution is to take only one read and then wait until the pen is lifted before plotting the pixel. Be sure to set PENBUF's number of read commands high to obtain the best resolution.

If the pen is used to do plotting, the luminances of the plotted points and the background should be matched as closely as possible. Otherwise, the pen values will drift toward the brighter pixels. It also would be very difficult to plot points adjacent to very bright pixels.



APPENDIX C - TROUBLESHOOTING

What to Do if Your ATARI Light Pen Does Not Respond as Expected

1. Check that the program was entered correctly.
2. Make sure the pen connector is properly seated in the controller jack.
3. Make sure there is no dust or dirt in the tip of the light pen. (See Periodic Maintenance, Section 9.)
4. Check the luminance level used in your SETCOLOR statement. If it is too low, the pen may fail to respond. Try a higher level setting.
5. Check the brightness setting of your television set. If it is too low, the pen may fail to respond.
6. Check to see if the pen switch is working correctly. With the switch on (pen pressed to the screen) and the pen plugged into jack #1, type in PRINT PEEK (632). The value returned should be 14. Lift the pen and repeat the PRINT PEEK (632). The value returned should be 15. If these do not match, you may have a defective switch or jack.
7. If you have problems in step 6, test for a defective jack by plugging the pen into jack #2 instead. Repeat step 6 but use location 633 instead. If you have no problems, the jack is faulty; if the problem still exists (#6 only), it is the light pen switch.
8. Do not attempt to disassemble the pen. Refer any repair to an authorized ATARI Service Center.





APPENDIX D - PENBUF FLOW CHART

APPENDIX D—PENBUF FLOW CHART

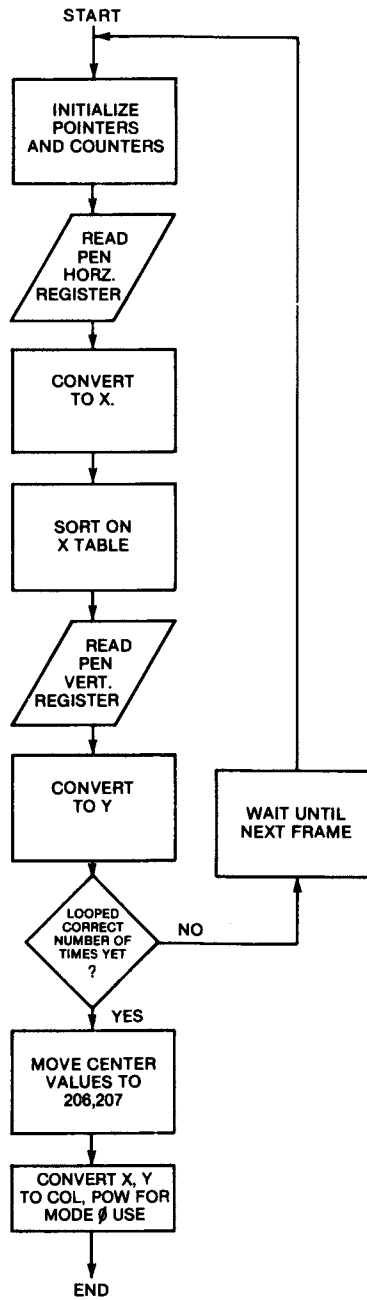


Figure D.1 PENBUF Flow Chart



APPENDIX E - PENBUF ASSEMBLY LANGUAGE LISTING

```

00CB      1000 TBLPTR =    $CB
00CD      1010 FRAMES =    $CD
00CE      1020 XCOORD =    $CE
00CF      1030 YCOORD =    $CF
00D0      1040 COL      =    $D0
00D1      1050 ROW      =    $D1
0000      1060          =    $600
0600 68    1070          PLA                pop off "USR"'s N value
0601 A200  1080          LDX #0            init counter
0603 A006  1090          LDY #6            store table hi byte in pointer hi byte
0605 84CC  1100          STY TBLPTR+1
          1110 ;
          1120 ;load x value & convert
          1130 ;
0607 AD0CD4 1140 LOAD   LDA $D40C        get light pen X coord
060A C90A   1150          CMP ##A
060C 9003   1160          BCC HERE
          1170 ;x>10, so subtract 228
060E 38    1180          SEC
060F E9E4   1190          SBC ##E4
0611 18    1200 HERE    CLC
          1210 ;add 154 to pen value
0612 699A   1220          ADC ##9A
          1230 ;call sort routine
0614 A075   1240          LDY #TABLES-$600 lo byte of X table address
0616 84CB   1250          STY TBLPTR
0618 205606 1260          JSR SORT
          1270 ;
          1280 ;find and store address of YTABLE
          1290 ;
061B A5CB   1300          LDA TBLPTR
061D 18    1310          CLC
061E 65CD   1320          ADC FRAMES      add table size to XTABLE address to find
0620 85CB   1330          STA TBLPTR      YTABLE address
          1340 ;
          1350 ;load y value & convert
          1360 ;
0622 AD0DD4 1370          LDA $D40D        get light pen Y coord
0625 38    1380          SEC
0626 E910   1390          SBC ##10        subtract 16
          1400 ;call sort routine
0628 205606 1410          JSR SORT
          1420 ;
          1430 ;repeat no. times specified in "FRAMES"
          1440 ;
062B E8    1450          INX
062C E4CD   1460          CPX FRAMES
062E F009   1470          BEQ FINISH
0630 A514   1480          LDA #14            load in which frame
0632 C514   1490 WAIT   CMP #14            compare org frame with current frame
0634 F0FC   1500          BEQ WAIT            loop until new frame

```

# ATARI Light Pen Operator's Manual

---

```

0636 4C0706 1510      JMP  LOAD      do another load
                1520 ;
                1530 ;X,Y buffering done - move medians (center values)
                1540 ;to locations "XCOORD" and "YCOORD"
                1550 ;
0639 A5CD   1560 FINISH LDA  FRAMES      load size of table
063B 4A     1565      LSR  A           halve size of table (find center)
063C A8     1570      TAY           load median offset
063D B1CB   1580      LDA  (TBLPTR),Y load median
063F 85CF   1590      STA  YCOORD
0641 A975   1600      LDA  #TABLES-#600
0643 85CB   1610      STA  TBLPTR
0645 B1CB   1620      LDA  (TBLPTR),Y load x median
0647 85CE   1630      STA  XCOORD
                1640 ;
                1650 ;convert X,Y coords to mode 0 coords
                1660 ;
0649 A5CE   1670      LDA  XCOORD      load X median
064B 4A     1680      LSR  A           divide by 2
064C 4A     1690      LSR  A           divide by 2
064D 85D0   1700      STA  COL         store column
064F A5CF   1710      LDA  YCOORD      load Y median
0651 4A     1720      LSR  A           divide by 2
0652 4A     1730      LSR  A           divide by 2
0653 85D1   1740      STA  ROW         store at row
0655 60     1750      RTS           back to basic
                1760 ;
                1770 ;perform insert sort of acc into table at address
                1780 ;specified in TBLPTR
                1790 ;
0656 86CE   1800 SORT  STX  XCOORD      transfer x reg to y reg using XCOORD as
0658 A4CE   1810      LDY  XCOORD      temporary storage
065A C000   1820 INSLOOP CPY  #0         hit bottom?
065C F014   1830      BEQ  INSERT      if so, insert
065E 88     1840      DEY           move pointer down
065F D1CB   1850      CMP  (TBLPTR),Y compare to next entry on table
0661 C8     1860      INY           restore pointer
0662 B00E   1870      BCS  INSERT      if A>=M, new entry's slot found
0664 88     1880      DEY           move index to next entry
                1890 ;move Y table entry to Y+1
0665 85CE   1900      STA  XCOORD      temporarily save acc
0667 B1CB   1910      LDA  (TBLPTR),Y
0669 C8     1920      INY           move pointer forward
066A 91CB   1930      STA  (TBLPTR),Y
066C 88     1940      DEY           restore pointer
066D A5CE   1950      LDA  XCOORD      restore acc
066F 38     1960      SEC
0670 B0EB   1970      BCS  INSLOOP      loop again
                1980 ;new value's proper place found - drop it in
0672 91CB   1990 INSERT STA  (TBLPTR),Y
0674 60     2000      RTS           from sort
0675      2010 TABLES =      *

0675      2020      .END

```

APPENDIX F - DESCRIPTION OF PENBUF OPERATION

The PENBUF assembly language subroutine performs two primary operations. It converts the register values to a usable range, as shown in Programming Example 1, and finds the statistical median of a user-specified number of successive pen reads.

The median is defined as the number in a data set that is less than 50 percent of the data and greater than the other 50 percent of the data. In other words, it is the number that is exactly in the center of a group of numbers (with respect to value). A median is a perfect choice for our purposes since values which are totally off target will not affect the result.

Example:

```
97, 101, 112...Median = 101
0, 101, 112...Median = 101
```

In the example above note that the 0, although far off target, does not affect the median. A statistical average would be inappropriate since values far off target would move the perceived value off target also. This is the downfall of an averaging technique. The light pen should sense the central value that occurs most often to discover the most likely true position of the pen.

This median value is found by sorting the data and reading the center value, which is precisely what PENBUF is designed to do. The read section of PENBUF reads the coordinates; the convert section converts those values to correct X and Y coordinates as described in Programming Example 1. Then, by the technique described below, the sort section uses an "insert-sort" to position the values as they are received.

When PENBUF completes the reading and sorting of the data, it moves the medians (center values of the tables) to locations 206 and 207 (decimal) and then performs a Graphics Mode 0 conversion of these values by simply dividing both values by 4. The results are stored in locations 208 and 209.

When PENBUF is done, it returns to BASIC. Now you only have to PEEK locations 206, 207 for values of X, Y, where X must be between 0 and 159 and Y must be between 0 and 95 (Graphics Mode 23), or PEEK locations 208, 209 for X, Y, where X lies between 0 and 39 and Y between 0 and 23 for Graphics Mode 0.

Before PENBUF is called, you must POKE location 205 with the number representing the desired accuracy of the pen. This number determines the number of readings that PENBUF performs and also determines the size of the data tables. The larger the number of reads, the more accurate the readings will be, but the read process will be slower. Thus, if the programmer desires fast response from the pen, this number should be as small as possible. If higher accuracy is needed,

such as in a menu selection, this number should be higher. A value of 9 is usually sufficient for a highly accurate determination of pen position.

The sorting of the X and Y pen coordinates is done by a method known as an insert-sort. A storage table is defined to hold the total number (max=69) of values to be sorted; 69 is the maximum because this is the maximum area of memory available to be allocated for the sort table. An error will occur if you try to use a greater number of readings. The sampling number is stored at decimal 205.

As each of the values is read, the table of previous values is examined one item at a time. If the new value is less than a value previously received, that previous value is moved to an array location with a subscript one number higher than the one previously occupied.

In other words, if you had previously processed six values, these could be represented as VAL(1) through VAL(6). These previous values would already be ordered from lowest value VAL(1) to highest value VAL(6). Now when the new one is read, it is compared to VAL(6). If the new value is lower than VAL(6), then VAL(7) is set equal to VAL(6), having the effect of moving the highest value into the next position in order to make a slot for the new value. Now a comparison is made to VAL(5). If lower, then VAL(5) is moved into the position formerly occupied by VAL(6), and so forth, until the correct slot is found.

This continues for all new values until each one has found a slot. Then the center value in the table is the median of the sort. This will be the value that occurs most often and will be the most likely true position of the pen. Random noise readings will have been pushed out to the outer ends of the table and will not affect the correct location of the median. The advantage of using an insert-sort is that the sort can be done as each data item is read. If you use a batch sort, you will have to wait while the data is sorted following all the readings. The insert-sort requires no wait because it sorts while reading.

The number stored at decimal 205, used to establish the number of reads, should be odd (1, 3, 5, ... 65, 67, 69) to provide a true middle point to the sort table. The diagram below represents a typical sorting sequence for the seventh value in a 7-value insert sort.

	TIME =							STORAGE TABLE FOR SORT	
0	0	1	3	22	55	79	[ ]	Value to sort = 10	
1	0	1	3	22	55	[ ]	79	Move 79--its bigger	
2	0	1	3	22	[ ]	55	79	Move 55--its bigger	
3	0	1	3	[ ]	22	55	79	New value is greater than VAL (3), so....	
4	0	1	3	[10]	22	55	79	Found its slot; put it in.	
VAL():	1	2	3	4	5	6	7		

Figure F.1 TYPICAL INSERT-SORT





APPENDIX G - NOTES ON MEMORY CONSERVATION/PENBUF MEMORY USAGE

In most of the demonstration programs in this manual, the PENBUF routine is appended as a part of the program. If you are trying to conserve memory for a particular program, there are two ways you can do it.

You can use the PENBUF loader contained in the ATARI Light Pen Demonstration Cassette program for the 8K version of PALLET. This occupies fewer bytes than the 16K PENBUF loader used in the other demo programs and yet performs the same function (it also loads faster). The only difference is that the decimal values in the loader have been converted into characters and stored in the BASIC string A\$.

The other way to conserve space is by leaving the computer console on and not disturbing Page 6. As long as the power stays on, the data in the memory areas will remain the same unless specifically changed by the programs running in the machine. The area used by PENBUF is not used by the operating system, so any program segment loaded there will stay undisturbed as long as the power is on.

The loader segment in each program from line 9000 onward places the PENBUF data into hex page 6 in the machine. It will not function anywhere else. If no other programs use this same area, you can load and run the loader routine once. Then you can use the USR calls to that area (decimal 1536) in other programs loaded later without needing to include the PENBUF loader within those other programs. Therefore you will have more room left in your program to use for other purposes.

The PENBUF routine will remain intact as long as the computer remains powered up. But you must remember to reload and run the loader routine before trying to use PENBUF within other programs, and do not POKE any data into locations 1536-1791 because this would disturb the PENBUF routine.

ATARI Light Pen Operator's Manual

---

LOCATION		FUNCTION
HEX	DEC	
\$CB	203	"PENBUF" internal use - any value placed here will be destroyed.
\$CC	204	"PENBUF" internal use - any value placed here will be destroyed.
\$CD	205	The value placed here determines the number of reads (see above).
\$CE	206	*Buffered X coordinate, $0 \leq X \leq 159$ (output)
\$CF	207	*Buffered Y coordinate, $0 \leq Y \leq 95$ (output)
\$D0	208	*Buffered column of pen, $0 \leq \text{COLUMN} \leq 39$ (output)
\$D1	209	*Buffered row of pen, $0 \leq \text{ROW} \leq 23$ (output)
\$600	1536	Start of "PENBUF" routine - enter it here
\$675	1653	End of "PENBUF" routine - start of data tables. No. bytes used in tables is $2x(\text{location } 205 = \$CD)$

\*Degree of buffering depends upon the value in location \$CD=205.

Figure G.1 PENBUF MEMORY USAGE

APPENDIX H - GENERAL PROGRAMMING NOTES

Familiarity with ATARI assembly language is not required to write programs that use the pen. However, ATARI has provided an assembly language routine (PENBUF) which is loaded in each case by the BASIC loader program and is called by the BASIC "USR" function. The following list provides general programming notes for the USR function as it applies to the light pen.

USING THE PEN WITH ATARI BASIC IN GRAPHICS MODE 0

1. Include PENBUF loader in your program.
2. GOSUB 9000 to load it into the right place.
3. Define PENBUF=1536 (as the start of the routine)
4. Sense that the pen switch is pressed; read location 632, 633, 634, or 635, depending on where the pen is installed (controller jack #1, 2, 3, or 4) 14 = pressed, 15 = not pressed.
5. Get X, Y coordinates by: JUNK=USR(PENBUF)

NOTE: X and Y coordinates will not change unless enough light reaches the pen.

6. X is in location 208, Y is in location 209.
- 7A. POSITION the cursor using X and Y and PRINT anything, OR
- 7B. Compare X and Y to other numbers known to represent desired responses.

USING THE PEN WITH ATARI BASIC IN GRAPHICS MODE 23

- 1-5. Same as above.
6. X is in location 206, Y is in location 207.
- 7A. PLOT X, Y as desired OR 7B. Same as above.

USING OTHER PROGRAMMING LANGUAGES WITH THE ATARI LIGHT PEN

1. Use the other language's equivalent commands to load PENBUF into Page 6.
2. Change the value stored at 1536 (hex \$600) to a 234 (hex \$EA). This is a NO-OP (No Operation) machine code instruction. BASIC passes a value onto the stack representing the number of values being passed in a BASIC USR routine. For a normal subroutine call

from languages other than BASIC, no such stack value will occur. Thus the "PLA" instruction must be deleted if you are not using BASIC.

3. Sense a switch closure on the pen by reading locations 632, 633, 634, or 635, depending on whether the pen is plugged into jack #1, 2, 3, or 4. The pen may be used in a jack other than jack #1 by using the following locations:

Location	Jack
632	1
633	2
634	3
635	4

4. When you call PENBUF as a subroutine, the X and Y values will be returned to the locations shown above. It is up to you to determine the correspondence between your plot coordinate system in this other language and the 160 by 96 (GRAPHICS 23) or 40 by 24 (GRAPHICS 0) of ATARI BASIC.