



Display Oriented Input Device

For the APPLE II

OPERATOR'S MANUAL

Copyright (C) 1979 by SOFTAPE. All rights reserved.  
This document, or the software supplied with it, may  
not be reproduced in any form or by any means in  
whole or in part without the prior written consent of  
the copyright owners.

SOFTAPE  
10432 Burbank Blvd.  
North Hollywood, Ca. 91601  
(213) 985-5763

## TABLE OF CONTENTS

SECTION	TITLE	PAGE
I.	Introduction.....	1
II.	Installation and Calibration.....	2
III.	Software Development Packages.....	3
IV.	Prefix Append Procedure.....	7
V.	INTEGER BASIC Editor Package.....	8
VI.	System Memory Map.....	10

## INTRODUCTION

The **Bright Pen** has been designed as an alternative to keyboard or game paddle input for the APPLE II computer. Data input for many computer applications can be greatly simplified by displaying several options on the video monitor and selecting one by touching it with the **Bright Pen**. This is especially true when programming the computer for users not familiar with computer data entry. Have you ever tried devising an input scheme for a board game like Chess or Checkers. Whatever your idea was, it probably wasn't as easy or fast as pointing out the piece to move and then the place to move it! For these types of input requirements the **Bright Pen** is a perfect solution. This package does not include **Bright Pen Super Chess Version 99.9**, but does include everything you need to develop your own INTEGER BASIC programs utilizing this versatile input device. Many practical examples are given to help you understand how to code BASIC programs to use the pen. The supplied software includes a tutorial menu selection demonstration, two assembly language driver packages, a **Bright Pen** editor for INTEGER BASIC programs and even a calibration program to set up your display. This package is definitely aimed at the computer programmer and easy program development was a major consideration in its design. If you are not a programmer and what you really wanted was **Bright Pen** applications programs, you will be glad to know that **SOFTAPE** plans to offer many programs either explicitly for the **Bright Pen** or with options for **Bright Pen** input.

## INSTALLATION AND CALIBRATION

1. Turn off the computer and carefully remove the lid.
2. Install the **Bright Pen** in the game (paddle) connector with the cable pointed toward the back of the computer.
3. Route the cable out the slot provided in the back and replace the lid.
4. Turn on the computer and load the Calibration program.

## Disk users:

1. Boot the disk drive
2. >LOAD <RETURN> (from tape)
3. >SAVE CALIBRATION <RETURN>
4. >RUN CALIBRATION <RETURN>

Once saved to disk steps 2 and 3 are omitted.

## Tape users:

1. <RESET>
2. \*<CTRL B> <RETURN>
3. >LOAD <RETURN>
4. >RUN <RETURN>

Now carefully follow the instruction given on the screen by the Calibration Program. Its purpose is twofold. First it will determine if your video display is of sufficient quality to produce accurate and reliable **Bright Pen** operation. Second it will aid you in finding the proper balance between brightness and contrast for your particular display. After your display is adjusted, make a note of the setting for future use.

## SOFTWARE DEVELOPMENT PACKAGES

Two assembly language driver packages are supplied for **Bright Pen** software development:

## MENU SELECTION PREFIX

This driver allows up to 32 points to be set up on the TEXT page and identifies the point selected by the pen. Only points set up may be selected.

## PEN LOCATOR PREFIX

This driver determines the GRAPHIC location of the pen anywhere on the screen and returns the horizontal (0-39) and vertical (0-47) position. It may also be used for TEXT page operation if the vertical value is divided by two.

Although the drivers are written in 6502 machine code, for ease of use they are disguised as INTEGER BASIC programs and can be LOAded and SAVEd as such. When executed, they RUN to their battle stations and await the CALL of duty. They were designed to append (attach themselves) to your application program resulting in a single INTEGER BASIC **Bright Pen** program. Each is independent of the other but may be used together if desired.

When a Prefix program is RUN without being appended to a user program a \*\*\* BAD BRANCH ERR message will be generated. This is normal and should not cause alarm. When properly appended to a program with a line number zero this message will not be displayed and execution will link correctly.



## MENU SELECTION PREFIX:

These routines are described in detail in the Menu Selection Demo program and that program should be studied before reading further. An outline of functions will be given here for reference and examples of operation are abundant in the demo program.

CALL 3072      Clear the list of points set for selection. This sets up the system for a new set of inputs.

POKE 0,HT : POKE 1,VT : CALL 3079

Set up a point for selection at line number VT and Character number HT. This CALL is executed for each and every point for selection. The selection points are numbered in order of setup starting with one.

CALL 3108      Wait for the user to select a point then return the number of the selected point in location 0. The statement ID=PEEK(0) will get this number.

## PEN LOCATOR PREFIX:

There are two entry points in this driver package. They are called 'FIND' and 'POLL'. To use them first set their names up as variables and assign the CALL values to them. This may be done in your program as follows:

```
10 FIND=4096 : POLL=4285
```

Now they may be CALLED by name. ( ie. CALL FIND )

'FIND'      Perform a full screen search for the pen. If found, a

'logical 1' (non-zero) value is returned in location 9. If not found, a zero is returned. The horizontal location (0-39) is returned in location 31 and the vertical (0-47) in location 30. If the pen is not found then obviously the positional data is not valid. This CALL leaves the system in graphics mode on exit.

```
100 GR : CALL FIND : REM SEARCH FOR THE PEN
```

```
110 IF PEEK (9) THEN 130 : REM SKIP 120 IF FOUND
```

```
120 PRINT "I THINK I'M ON A WILD GOOSE CHASE":END
```

```
130 H=PEEK(31) : V=PEEK(30) : REM GET THE LOCATION
```

```
140 PRINT "EUREKA!! I'VE FOUND IT AT ";H;",";V : END
```

'POLL'      This routine polls the pen to see if it is pointed at a light or dark part of the screen. If the pen is detecting light a logical true (non-zero) will be returned in location 9. On dark, a zero is returned in location 9.

```
100 CALL POLL : REM IS IT LIGHT OUTSIDE?
```

```
110 IF PEEK(9) THEN PRINT "GOOD MORNING!"
```

```
120 IF NOT PEEK (9) THEN PRINT "GOOD NIGHT..."
```

```
130 FOR TIME = 1 TO 1000 : NEXT TIME : GOTO 100
```

These routines may be combined to achieved more sophisticated routines. The following BASIC subroutine (GOSUB) will wait until light is detected then attempt to find the pen. If the pen is found, its position will be stated. If not found, the procedure will be repeated five times before giving up.

```

1000 FOR TRY=1 TO 5
1010 CALL POLL
1020 IF NOT PEEK (9) THEN 1010
1030 CALL FIND : TEXT
1040 IF NOT PEEK (9) THEN 1080
1050 H=PEEK(31):V=PEEK(30)/2
1060 PRINT "THE PEN IS AT LINE #";V;" AND CHARACTER
POSITION ";H
1070 TRY=5 : REM EXIT LOOP IF FOUND
1080 NEXT TRY
1090 RETURN

```

## APPENDING THE PREFIX TO USER PROGRAM

The append procedure outlined below takes two INTEGER BASIC programs and combines them into one. The term "PREFIX" refers to the assembly language driver packages. The term "USER PROGRAM" is the applications program which uses the Bright Pen drivers. After your program is debugged, follow these steps to attach it to the driver prefix. Then you will have one INTEGER BASIC program which is easy to run.

1. <RESET>
  2. \*3D3G <RETURN> (FOR DISK USERS)  
-OR-  
\*<CTRL B> <RETURN> (FOR TAPE USERS)
  3. >LOAD (FILENAME) <RETURN> (LOAD USER PROGRAM)
  4. >CALL -151 <RETURN> (GOTO MONITOR)
  5. \*CA.CB <RETURN> (DUMP PROGRAM POINTER)
- 00CA - PL PH (VARIABLES FOR STEP 7)
6. \*4C.4D <RETURN> (DUMP HIMEM)
- 004C - HL HH (VARIABLES FOR STEP 11)
7. \*4C: PL PH <RETURN> (PL & PH FROM STEP 5)
  8. \*<CTRL C> <RETURN> (RETURN TO BASIC)
  9. >LOAD (FILENAME) <RETURN> (LOAD PREFIX PROGRAM)
  10. >CALL -151 <RETURN> (GOTO MONITOR)
  11. \*4C: HL HH <RETURN> (HL & HH FROM STEP 6)
  12. \*<CTRL C> <RETURN> (RETURN TO BASIC)
  13. >SAVE (FILENAME) <RETURN> (SAVE TOTAL PROGRAM)

## BRIGHT PEN INTEGER BASIC EDITOR

This software package allows easy and fast editing of INTEGER BASIC programs. Just RUN the editor, LOAD your program, turn the editor on and your're on your way. Editing a line is as easy as positioning the pen and entering a few keystrokes. Editing time is cut in half compared with standard keyboard editing. The editor is an assembly language program but is handled as INTEGER BASIC for storage. To use the editor, first LOAD it then RUN it. It is now in memory waiting to be used. Now LOAD a program to be edited.

Turning the editor on: CALL 4352

This engages the editor and disengages the DOS if present.

Exiting the editor:

Disk users:	<RESET>	Tape users:	<RESET>
	*3DØG <RETURN>		*<CTRL C> <RETURN>

## EDITOR FUNCTIONS:

<CTRL A> Address the cursor to the pen location. If the pen is not found an ERR: PEN message will be displayed.

<CTRL S> Copy from the cursor to the character preceeding the pen location. If you attempt to copy backwards from the cursor an ERR: CANNOT COPY BACKWARDS message will be displayed.

<CTRL Q> Accept the line as edited. The line will be parsed by BASIC and if OK it will be accepted without notice. If incorrect, BASIC will generate an appropriate error message.

<CTRL W> Display the last line edited at the bottom of the screen. Very handy for correcting a line after an error has occured.

Editor commands have been selected for speed and ease of use. It is suggested that you press the CTRL key with the ring finger of your left hand and use your middle and index fingers for the A,S,Q and W keys. This frees your right hand for the pen and your eyes for the screen. Happy Editing!

## SYSTEM MEMORY MAP

## BASE PAGE USAGE:

\$00 - 0C	Various pointers and flags
\$1E - 1F	Vertical and Horizontal pen positions

## MEMORY ASSIGNMENTS:

\$100 - 3FF	Reserved APPLE II systems area
\$400 - 7FF	First TEXT page
\$800 - BFF	Reserved for screen search buffer
\$C00 - CFF	Menu Selection driver code
\$D00 - FFF	Data and free space
\$1000-10FF	Pen Locator driver code
\$1100-11FF	Bright Pen Editor (when in use)
\$1200-????	Free space for BASIC (LOMEM)