

PDP-4 PROGRAM LIBRARY

NUMBER: DEC - 4 - 45 - M

(7-78-m)

NAME: 370 Light Pen Diagnostic Program

AUTHOR: C. Stein - DEC

DATE: April 29, 1964

SPECS: VE

NEEDED: 340 Display

ABSTRACT: Light Pen Diagnostic is a utility program designed to test the 370 Light pen operation with a 340 display. The test and intensity settings are selected by the AC switches. The program starts in location 22g. All error detection is visual.

CHAPTER I

Console Operating Procedure

The tables below describe the operation procedure to be used when running this diagnostic:

Table 1-1 Loading Procedure:

1. Place tape 370 DIAGNOSTIC in reader
2. Start in 7770g or 17770, depending on the size of the computer.

Table 1-2 Switches:

<u>Switch</u>	<u>Meaning</u>
1	Sensitivity and resume iot test
3	Light pen follow test
5	Field of view test
7-9	X coordinate of lower left corner of field of view box
11-13	Y coordinate of lower left corner of field of view box
15-17	Intensity value for pen follow and field of view box

All error detection is visual.

CHAPTER II

Test Description

- 2-1 Sensitivity and Resume Iot Tests: This tests the light pen interrupt and the display resume iot (DRS.) at all eight intensity levels. Eight horizontal vectors are drawn parallel to one another starting at $x = 700g$, $y = 100g$, and at minimum (0) intensity with the last vector at maximum (7) intensity starting at $x = 700$, $y = 1000g$. When the light pen is placed at any point to the right of center of any line, the vector will be truncated. This demonstrates that the pen is "seeing" light. If the pen is placed to the left of center, the line appears in full. This shows that the display is resuming after a pen interrupt.
- 2-2 Light Pen Follow Test: This test indicates the speed of response of the 370 light pen. A cross is drawn in the center of the screen. The operator places the pen on the cross and then can move the pen. Two lines are drawn perpendicular to one another from the center of the screen to the present pen position. Thus as the pen is moved, a different line is drawn. The cross is drawn in such a way, that if the pen approaches the edge of the screen, the cross will stop.

2-3 Field of View Test: This test determines the light pen field of view or the area "seen" by the pen at any position or intensity on the screen. A box approximately one half inch on a side is displayed on the screen. The position and intensity of the box is determined by the user. Every point within the box is displayed. When the pen is placed over the box, a 4x enlargement of the points "seen" is drawn in the opposite half of the screen. Above the box a digital readout is displayed of the number of points seen by the pen in decimal. To use this test, set AC switches 0 and 5 up and place two numbers in AC switches 7-9 and 11-13. The number placed in AC switches 7-9 and 11-13 are the x and y coordinates of the lower left corner of the solid box, divided by 200₈. Thus if the number 5 were placed in switches 11-13 and 6 placed in 7-9, the coordinate of the lower left corner of the box would be:

$$X = 6 \times 200 = 1400_8$$

$$Y = 5 \times 200 = 1200_8$$

The purpose of allowing the user to position the box is to allow him to test the field of view at any point on the screen.

Next place a desired intensity setting (0=minimum, 7=maximum) in AC switches 15-17. This allows the user to test the sensitivity of the pen to all or any intensity level.

Now the pen may be placed over the small box, and the 4x enlargement and digital readout will be seen. To eliminate the digital readout, set AC switches 0 down. To stop the test, set AC switches 0 and 5 down.

bthsid, dac temp2
 lac (jmp 1 outgo)
 dac stpcod
 dac corhit
 lac (jmp gotcha)
 dac lphit
 dzm lpct

outpt, jms outgo
 jms outgo
 lac noswit
 sna
 jmp .+6
 law nobfxx
 lot 606
 iof
 lot 601
 jmp .-1
 lac buf+1
 add (1)
 sad bufdon
 skp
 jmp incre
 lac temp1
 dac buf+1
 lac temp2
 dac buf 5
 lac buf 6
 add (4)
 dac buf 6
 lac buf+2
 add (1)
 sad bufd1
 jmp done
 dac buf+2
 jmp outpt

incre, dac buf+1
 lac buf 5
 add (4)
 dac buf 5
 jmp outpt

```
done,      las
           and (400000)
           sne
           jmp lftsid-6
           lac buf+5
           and (1777)
           dac call+2
           lac (1300)
           dac call+3
call,      lac lpet
           jms outnox
           0
           0
           dac nobfxx
           dac tempi
           lac (400000)
           dac i tempi
           isz tempi
           lac (3000)
           dac i tempi
           law .
           dac noswit
           las

           and (10000)
           sza
           jmp hole 2
           dzm noswit
           jmp i hole

lftsid,   lac buf+5
           jmp bthsid

hobuf,   34110
         20000
         202000
         13000
         30137
         20000
         342000
         306310
         621462
         3000
         777777
```

```
gotcha,    isz lpct
           iot 704
           law buf 4
           jmp outgo 2
nobfxx,    nobfxx+300/

blt,       0
           dac 10
           law buf-1
           dac 11
           lac i 10
           sad (777777)
           jmp i blt
           dac i 11
           jmp blt+4

follow,    0
           lac fx
           dac xpt
           lac fy
           dac ypt
           las
           and (7)
           xor param
           dac buf
           lac (3000)
           dac buf+4
           lac ypt
           xor fywd
           dac buf 1

           lac (jmp gety)
           dac stpcod
           lac (jmp 1 outgo)
           dac lphit
           lac (hit)
           dac corhit
           lac xpt
           add (50)
           and (2000)
           sza
           jmp backup
           lac xpt
           add (50)
           xor fxwd
           dac buf+2
           lac (600277)
           dac buf+3
           jms outgo
           iot 716
```


rtr
rtr
rtr
rtr
and (1776)
dac x1

lac xpt
tad (-47)
spa
jmp moveup
xor fxwd
dac buf+2
lac (600077)
dac buf+3
jms outgo
lot 716

rtr
rtr
rtr
rtr
and (1776)
add x1

gety,
rar
and (1777)
dac xpt
lac (jmp sho)
dac stpcod
lac xpt
xor fxwd
dac buf+2
lac ypt
add (50)
and (2000)
sra
jmp bacyup
lac ypt
add (50)
xor fywd
dac buf+1
lac (737400)
dac buf+3
jms outgo

lot 716
ral
and (1776)
dac y1
lac ypt
tad (-47)
spa
jmp movyup

```
xor fywd  
dac buf+1  
lac (637400)  
dac buf+3  
jms outgo  
iot 716  
ral  
and (1776)  
add y1  
rcr  
dac ypt
```

```
show,    law buf 2  
         dec 10  
         lac (34114)  
         dec buf  
         lac fy  
         xor fywd  
         dac buf 1  
         lac fx  
         xor fxwd  
         dac buf 2  
         lac fx  
         cma  
         tad (1  
         tad xpt  
         dzm sign  
         spa  
         jms absval  
         dac mag  
         and (777600)  
         sza  
         jmp modify
```

```
lastx,   lac (200000)  
         xor mag  
         xor sign  
         dac i 10  
         lac fy  
         cma  
         add (1)  
         tad ypt  
         dzm sign  
         spa  
         jms yabs  
         dac mag  
         and (777600)  
         sza  
         jmp modifx
```

lasty, lac mag
 rtl
 rtl
 rtl
 rtl
 and (77400)
 xor sign
 xor (600000)
 dac i 10
 lac (3000)
 dac i 10
 lac (jmp done1)
 dac stpcod
 dac lphit
 dac corhit
 jms outgo

abaval, 0
 cma
 dac temp2
 lac (200)
 dac sign
 lac temp2
 jmp i abaval

yabs, 0
 cma
 dac temp2
 lac (100000)
 dac sign
 lac temp2
 jmp i yabs

done1, las
 and (40000)
 sxa
 jmp follow+5
 jmp i follow

modify, lac mag
 tad (-176)
 dac mag
 lac (200177)
 xor sign
 dac i 10
 lac mag
 jmp lastx-3

```
modifx,   lac (277400)
           xor sign
           dac i 10
           lac mag
           tad (-176)
           dac mag
           jmp lasty-3

sensst,   0
           law sensbu-1
           jms blt
           lac (jmp gotlp)
           dac iphit
           lac (jmp i outgo)
           dac stpeed
outpt1,   jms outgo
           lac buf
           sad (34117)
           jmp reset
           add (1)
           dac buf
           lac buf+2
           add (100)
           dac buf+2
           jmp outpt1
reset,    lac (34110)
           dac buf
           lac (302100)
           dac buf+2
           las
           and (200000)
           sza
           jmp outpt1
           jms i sensst

gotlp,    lot 712
           and (400000)
           sza
           jmp outpt1+1
           lot 504
           jmp outgo 3

backup,   lac xpt
           tad (-27)
           dac xpt
           jmp gety
```

moveup, lac xpt
tad (30
dac xpt
jmp gety

bacyup, lac ypt
tad (-27
dac ypt
jmp show

movyup, lac ypt
add (30
dac ypt
jmp show

sendu, 34110
20700
302100
600177
3000
777777

outgo, 0
law buf
lot 606
ion
jmp .

inter, jms corrat
skp
jmp corhit-1
lot 701
skp
jmp lphit-1
lot 601
skp
jmp stpcod-1
lot 102
lot 202
lot 302
lot 402
ion
jmp i 0

dsi=lot 601
dsp=lot 701

 lot 704
stpcod, 0
 lot 704
corhit, 0
 lot 704
lphit, 0

corrut, p
 dsx
 skp
 jmp . 3
 dsy
 jmp i corrut
 isz corrut
 jmp i corrut

buf, 0
buf 20/

dsx=lot 501
dsy=lot 1001

fy, 1000
fx, 1000
param, 34110
fxwd, 102000
fywd, 220000

#start

display octal and decimal output

outnox, 0

```

// routine to convert octal numbers to decimal and generate
// buffer for display
// calling sequence:
// call, lac a number to be converted
//          jms outnox
//          x x coordinate of left number
//          bit 0=1 if octal rather than dec.
//          y y coordinate of left number
//          dac buff address of buffer
//          return last buffer address in AC
//          leading zeros replaced by blanks.

```

```

dac outmpx
lac i outnox
and (377777)
xor wdx
dac tix
lac i outnox
and (400000)
dac t58x
isz outnox
lac i outnox
xor ydx
dac t2x
isz outnox
lac 10
dac t3x
lac i outnox
dac 10
lac t58x
sxa

```

bothx,

```

jnp octout
lac (tab-1)
dac tabcon
lac paramx
xor i outnox
lac tix
dac i 10
lac t2x
dac i 10
lac 11
dac tix
lac 12
dac t2x
lac tabcon
dac 11
dcm zeroart
-4
dac cvntx

```

```
loopx,      dzu t4x  
            lac 1 11  
            dac t5x  
            lac outmpx  
            tad t5x  
            cps  
            jmp .+3  
            isz t4x  
            jmp .-4  
            dac outmpx  
            lac t5x  
            cma  
            tad outmpx  
            dac outmpx  
            isz outmpx  
            lac t4x  
            sna  
            jms zerois  
            law .  
            dac zerois  
            lac t4x  
            add (base)  
            dac t69x  
            xct 1 t69x  
            jms bltx  
            isz evntx  
            jmp loopx  
            nop  
            -0  
            dac evntx  
            lac (jmp .+4  
            dac .-4  
            dac zerois  
            jmp loopx  
            lac (nop  
            dac .-10  
            lac 10  
            dac t4x  
            lac t3x  
            dac 10  
            lac tix  
            dac 11  
            lac t2x  
            dec 12  
            isz t4x  
            lac t4x  
            isz outnox  
            jmp 1 outnox
```


paramx, 30177
xwdx, 20000
ywdx, 340000
tab, -303237
-23417
-1747
-143
-11
-0

zerox, 307042
221250
331463
237400
104200

onex, 777777
304216
221042
227400
135673
030000

twox, 777777
225252
227317
230000
031460
234210
104000

threex, 777777
225612
167340
225213
237660
135400

fourx, 777777
124000
226302
325243
231463
236000
104200

fivex, 777777
225612
221354
370042
224210
135463
031400
777777

sixx,	307052
	341052
	305400
	170000
	235477
	104200
	TTTTTT
sevenx,	021042
	221210
	301777
	371460
	104210
	100000
	TTTTTT
eightx,	307052
	361250
	331773
	237400
	104200
	TTTTTT
ninex,	225612
	221042
	366363
	334000
	135660
	TTTTTT
blank,	104210
	100000
	TTTTTT

```
zerols,      0
              lac zerowl
              sxa
              jmp zerwit
              lac (12)
              dac t4x
              isz zerols
zerwit,      isz zerols
              isz zerols
              jmp 1 zerols

bitx,        0
              dac 12
              lac 1 12
              sad (-0
              jmp 1 bitx
              dac 1 10
              jmp bitx+2

base,        law zerox-1
              law onex-1
              law twox-1
              law threex-1
              law fourx-1
              law fivex-1
              law sixx-1
              law sevenx-1
              law eightx-1
              law nines-1
              law blank-1

outout,      lac (tabix-1)
              dac taboon
              jmp bothx

tabix,       TTTT7
              TTTT
              TTT
              TT
              T
              -0

start
```