

VS60-F
VS60-H

VS60 VIS W/XY CORLTR
CZVSFAO

AH-F803A-MC
FICHE 1 OF 1

FEB 1981
COPYRIGHT © 1980
MADE IN USA



IDENTIFICATION

B 1

SEQ 0001

PRODUCT CODE: AC-F801A-MC
PRODUCT NAME: CZVSFA0 VS60 VIS W/XY CORLTR
DATE: MAR 1980
MAINTAINER: DIAGNOSTIC ENGINEERING

COPYRIGHT (C) 1980
DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASS.

THIS SOFTWARE IS FURNISHED UNDER A LICENSE FOR USE ONLY ON A SINGLE COMPUTER SYSTEM AND MAY BE COPIED ONLY WITH THE INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE, OR ANY OTHER COPIES THEREOF, MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY OTHER PERSON EXCEPT FOR USE ON SUCH SYSTEM AND TO ONE WHO AGREES TO THESE LICENSE TERMS. TITLE TO AND OWNERSHIP OF THE SOFTWARE SHALL AT ALL TIMES REMAIN IN DEC.

THE INFORMATION IN THIS DOCUMENT IS SUBJECT TO CHANGE WITHOUT NOTICE AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT CORPORATION.

DEC ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DEC.

0.0 TABLE OF CONTENTS

- 1.0 ABSTRACT
- 2.0 REQUIREMENTS
 - 2.1 EQUIPMENT
 - 2.2 STORAGE
- 3.0 LOADING PROCEDURE
- 4.0 STARTING PROCEDURE
- 5.0 TEST PATTERN CONTROL
 - 5.1 SWITCH REGISTER CONTROL
 - 5.2 KEYBOARD CHARACTER CONTROL
- 6.0 ERROR REPORTING
- 7.0 MISCELLANEOUS
 - 7.1 VS60 BUS/VECTOR ADDRESS MODIFICATION
 - 7.2 XXDP/APT NOTES
 - 7.3 POWER FAIL
 - 7.4 SINGLE VS60 TESTING
- 8.0 EXECUTION TIME
- 9.0 PROGRAM TEST DESCRIPTIONS
- 10.0 LISTING

1.0 ABSTRACT

THIS PROGRAM IS A VARIATION OF CZVSDC (VS60 VISUAL)
WITH XY POINT CORRELATOR AND WRITING TABLET FUNCTIONS
INCLUDED FOR IN-HOUSE CAD APPLICATIONS.

G.P. MAR '80

THE PROGRAM PROVIDES THE OPERATOR WITH TWENTY TWO VISUAL FRAMES
(TWENTY FOUR IF POINT CORRELATOR ECO'S ARE INSTALLED) TO ADJUST
OR VERIFY THE VISUAL OPERATION OF THE VS60 DISPLAY SYSTEM.
NORMALLY EACH FRAME WILL CYCLE FOR ABOUT 5 SECONDS BEFORE
ADVANCING TO THE NEXT FRAME. EACH VISUAL FRAME CAN BE SELECTED
VIA SWITCH REGISTER OR KEYBOARD SELECTION.

2.0 REQUIREMENTS

2.1 EQUIPMENT

- A. PDP-11 COMPUTER WITH AT LEAST 16K OF MEMORY.
- B. I/O TERMINAL (I.E. ASR33 TTY OR LK40).
- C. VS-60 DISPLAY SYSTEM.
- D. ADDITIONAL VS-60 DISPLAY CONSOLE (OPTIONAL).
- E. WRITING TABLET (IF POINT CORRELATOR ECO INSTALLED).

2.2 STORAGE

THE PROGRAM OCCUPIES THE LOWER 10.5K OF MEMORY BUT
REQUIRES 16K (BUFFER SPACE) TO RUN.

3.0 LOADING PROCEDURE

NORMAL PROCEDURE FOR LOADING A BINARY PROGRAM INTO MEMORY SHOULD
BE FOLLOWED.

4.0 STARTING PROCEDURE

LOAD ADDRESS 200 AND START TO INITIALIZE THE SYSTEM
AND BEGIN TESTING.

5.1 SWITCH REGISTER CONTROL

SWITCH	FUNCTION
SW14=1	LOOP ON CURRENT TEST
SW09=1	STOP SUB-PICTURE MOTION
SW08=1	LOOP ON TEST IN SWR<4:0>
SW07=1	ENABLE KEYBOARD CONTROL (REF.5.2)

5.2 KEYBOARD CONTROL

STARTING THE TEST WITH SR7=1 WILL ENABLE KEYBOARD CONTROL. KEYBOARD CONTROL IS AN AUXILIARY METHOD OF SELECTING THE TEST FRAME, LOOP ON A TEST FRAME, OR STOP-START FRAME MOTION. THE SWITCH REGISTER BITS OVERRIDE THE KEYBOARD CONTROL. THE DIRECTORY FRAME PROVIDES THE OPERATOR WITH THE KEYBOARD LETTER AND SWITCH REGISTER VALUE FOR EACH TEST PATTERN. TO SELECT A TEST PATTERN, SIMPLY DEPRESS THE TEST LETTER ON THE CONSOLE KEYBOARD. DEPRESS THE "RUB-OUT" KEY TO LOOP ON THE CURRENT TEST PATTERN. DEPRESS THE "CR" KEY TO STOP MOTION. UNDEFINED TEST LETTERS WILL DISPLAY THE DIRECTORY FRAME. ALL OTHERS WILL HAVE NO EFFECT OTHER THAN TO RESUME PICTURE MOTION.

6.0 ERROR REPORTING

THE PROGRAM ONLY DISPLAYS VISUAL ERRORS AND DOES NOT REPORT ANY LOGIC ERRORS.

7.0 MISCELLANEOUS

7.1 VS60 BUS/VECTOR/PRIORITY ADDRESS MODIFICATION

MODIFY LOCATION 1242 (\$VECT1) IF BASE VECTOR ADDRESS IS NOT 100320.
MODIFY LOCATION 1246 (\$BASE) IF BASE BUS ADDRESS IS NOT 172000.
MODIFY LOCATION 1252 (\$CDW1) IF BASE TABLET ADDRESS IS NOT 175620.
MODIFY LOCATION 1254 (\$CDW2) IF BASE TABLET VECTOR IS NOT 300.

NOTE: A RESTART IS REQUIRED AFTER THE ABOVE ADDRESS MODIFICATION.

7.2 XXDP/APT NOTES

THE VISUAL TEST IS CHAINABLE UNDER XXDP IF 16K OR GREATER MEMORY IS AVAILABLE. THE VISUAL TEST INCLUDES THE 'APT' SOFTWARE HOOKS, HOWEVER THEY HAVE NOT BEEN TESTED.

7.3 POWER FAIL

A POWER FAILURE WILL CAUSE THE PROGRAM TO BE RESTARTED.

7.4 SINGLE VS60 TESTING

THE VISUAL TEST DOES NOT TEST MULTIPLE VS60'S. THE VISUAL TEST WILL UTILIZE THE SECOND CONSOLE IF CONNECTED. THE "A" AND "U" FRAMES ARE USED TO VERIFY PROPER OPERATION BETWEEN THE TWO DISPLAY CONSOLES.

8.0 EXECUTION TIME

EXECUTION TIME IS APPROX. FOUR MIN. AN 'END OF PASS' IS INDICATED BY A RETURN TO THE DIRECTORY FRAME. NO 'END OF PASS' MESSAGE IS TYPED.

9.0 PROGRAM TEST DESCRIPTIONS

A = 01 Directory Frame

The sub-picture supplies the operator with a List of the Different Visual frames for his inspection.

This frame also includes a list of switch register values and keyboard control letters to select the visual frames. When a non-valid switch register value or keyboard key has been selected, the directory frame will be displayed. IF THE SECOND CONSOLE IS CONNECTED, THE OPERATOR SHOULD VERIFY THE 'THIS IS CONSOLE 0' MESSAGE ON CONSOLE #0 AND THE 'THIS IS CONSOLE 1' MESSAGE ON CONSOLE #1.

The frame is displayed by doing the following:

1. Point to x = 0 y = 1500
2. Enable console 1 intensity
3. Enter "character" mode and display inline text.
4. Display "STOP"
5. Display "JUMP ABSOLUTE" to the start of the frame.

B = 02 Astigmatism and Settling Time Frame

The frame will display points at individual bits at each x and y position register.

A floating one pattern used on each register followed by an accumulation pattern.

Bit 9 of x pos. Bit 9 of y pos.
 Bit 8 of x pos. Bit 9 of y pos.
 etc. etc.

Bits 9 and 8 of x pos. Bits 9 and 8 of y pos.
 etc. etc.

C = 03 Short Term Drift Frame

The frame will display five points. The points will be displayed in each corner and the center of the screen. Each point actually consists of four "Display Point" instructions.

The point is generated by:

1. Positioning the x and y DAC at a coordinate.
2. Intensifying the coordinate ONCE.
3. Do not intensify the point again for five (5) milliseconds.
4. Repeat 2 and 3 three more times.
5. If all the coordinates have not been displayed, update the coordinate and REPEAT 1 thru 4.

The C.P.U. cycle time is a factor in the 5 msec. delay routine. The current delay value (location "DELAY") is valid for a PDP-11/40 CPU type.

D = 04 Minor Axis Gain, Offset and Phase Frame

The frame consists of three square boxes with diagonal bisecting lines. The largest box encompasses the whole main screen viewing area. The second box, whose size is 100., is displayed in the right center area. The third box, whose size is 10., is below the second box. The boxes are drawn counter clockwise from the lower left corner. Upon completion the procedure is reversed and drawn clockwise from the lower left corner. When drawing the clockwise box the "Negative" polarity bit is set to enable adjustment of the "Offset" pot. Each box, upon completion, is segmented by a diagonal line from lower left to upper right and lower right to upper left corner. The frame also draws the same type box in the "menu" area. Because the "menu" is narrower than the high, the result is an rectangle in the menu area. In the lower center area, a series of four vectors 200 units long, are drawn from a common point. In the left center quadrant, ten vectors are drawn using "SHORT VECTOR" mode. Each of the vectors have a

H 1

length of eight units. After drawing the vector a 'RELATIVE POINT' is displayed two units away from the end of the vector. The "Y" coordinate is updated by two units and the "SHORT VECTOR" and "RELATIVE POINT" sequence is repeated. The visual result is a vertical "DOT-DASH" line. Included in the left quadrant is the Intensity Delay sub-picture. Eight vectors are drawn away from a "COMMON POINT" offset by ONE unit. The result will appear to be a square formed by the starting points of the vectors. Each vector has a length of 40 units.

VECTOR #	ORIGINATES AT X	ORIGINATES AT Y
1	0354	1003
2	0354	1004
3	0353	1004
4	0352	1004
5	0352	1003
6	0352	1002
7	0353	1002
8	0354	1002

E = 05 Major Axis Offset and Vector Start Frame

The frame includes the minor axis gain frame plus two additional patterns. The first is used to adjust the vector starting point. The second pattern to adjust the major axis offset. The first pattern is drawn, in the upper quadrant, with the following manNEr:

Vector #	Direction
1	Positive Vertical Reference Vector.
2	Positive Horizontal Vector starting on VECTOR #1
3	Positive Horizontal Vector starting 1 unit RIGHT OF VECTOR #1
4	Negative Horizontal Vector starting on VECTOR #1
5	Negative Horizontal Vector starting 1 unit LEFT OF VECTOR #1
6	Positive Horizontal Vector start at the BOTTOM OF VECTOR #1
7	Negative Vertical Vector starting at the bottom OF VECTOR #1
8	Negative Horizontal Vector starting at the BOTTOM OF VECTOR #1

The second pattern draws, from a common point (x=1000, y=400), four pairs of vectors. The first of each pair is drawn with the "y" axis being the major value. With the second using the "x" as the major. THE THIRD PATTERN CONSISTS OF 10 PAIRS OF SHORT LENGTH VECTORS AND RELATIVE POINT'S DRAWN IN THE LEFT CENTER AREA. AN 8 UNIT SHORT VERTICAL VECTOR IS DRAWN FOLLOWED BY A ONE UNIT RELATIVE POINT.

F = 06 Vector Length Gain, Convergence and Vector Linearity Frame

SEQ 0008

The pattern appears to be a series of horizontal lines being intersected by a diagonal line from upper left to lower right.

The picture is drawn by:

1. Draw an outer reference box
2. Starting from maximum, draw an increasing negative length vector from an increasing "y" origin.
3. Starting from minimum x, draw a decreasing length vector from an increasing "y" origin.
4. Starting in the upper left edge, intensify a point at the intersection of #2 to #3 vector.
5. Starting in the upper left edge, intensify a decending vector that is over #4.
6. From center screen, using "BASIC" Vectors draw two intersecting "x" and "y" lines.

G = 07 Pincushion Frame

Using the "LONG" Vector instruction, display a "CROSS HATCH" visual pattern. The frame can be used to detect distortion in Vectors. From a distance of three feet, all vectors should appear straight with no vector curvature.

H = 10 Octagons AND CIRCLES Frame

The purpose of the frame is to verify the endpoint matching of vectors. FIVE octagons are drawn from the center of the screen. The outer most octagon is drawn by using the "ABSOLUTE VECTOR" instruction from the point x = 530 y = 10.

Vector #	from	x-y	to	x-y
Vector 1	from	530-10	to	1250-10
Vector 2	from	1250-10	to	1770-530
Vector 3	from	1770-530	to	1770-1250
Vector 4	from	1770-1250	to	1250-1770
Vector 5	from	1250-1770	to	530-1770
Vector 6	from	530-1770	to	10-1250
Vector 7	from	10-1250	to	10-530
Vector 8	from	10-530	to	530-10

The FOUR concentric octagons are drawn by using the "LONG VECTOR" display instruction. The sizes are 377, 177, 77, 7 respectively. Two more octagons with a size of 17 units are drawn at x = 300 y = 1000 and x = 1500 y = 1000. These two are drawn using the "BASIC SHORT" vector display instruction. THREE CONCENTRIC CIRCLES ARE DRAWN USING ABSOLUTE VECTOR MODE. EACH CIRCLE CONSISTS OF 45 ABSOLUTE VECTORS. THE THREE CIRCLES HAVE A RADIUS OF 64., 128., AND 256. RESPECTIVELY.

I = 11 Scissoring and Vector Scaling Frame

The frame starts out by displaying a reference box around edge of the screen.

A VECTOR IS DRAWN FROM AN "ON-SCREEN" POSITION TO AN "OFF-SCREEN" POSITION. Another vector is drawn from the end of the previous vector back into the viewing area. This is repeated four times on each screen edge. The vectors should all terminate WITH NO bending or distortion. After all edges have been intersected, draw a large diamond that intersects each edge. The diamond and the vectors crossing the edges are the standard vector length. To verify that vector scale operates properly, draw a square in the center of the screen. By changing the value of the "Vector Scale" register the box should increase in size. the vector scale is changed with the resulting picture being sixteen scaled boxes in the center of the screen.

J = 12 X and Y Dynamic Offset Frame

IN THIS FRAME, A 1000 UNIT BOX IS DRAWN IN THE CENTER OF THE SCREEN. USING THE DISPLAY "OFFSET" INSTRUCTION, THE BOX IS MADE TO SLIDE ACROSS THE SCREEN TO THE RIGHT, THEN TO THE LEFT, TOP, AND BOTTOM EDGES.

AFTER THESE FOUR MOTIONS THE BOX IS RETURNED TO CENTER AND MOVED DIAGONNALLY TO THE UPPER RIGHT, AND LOWER LEFT BY SETTING THE OFFSET REGISTERS DIRECTLY FROM THE CPU (EXTERNAL TO THE DISPLAY FILE).

THE RANGE OF OFFSET USED IS 0 TO 1400 (POS AND NEG) IN BOTH CASES.

K = 13 Character Scale Frame

The frame function is to verify that character scale does change the size. To verify character scale, six characters, (the letters A, B, F, O, T and X) are displayed. each character starts with the largest to the smallest size on a common base line. A horizontal reference is drawn along the base of the characters.

L = 14 Character Quality and Character Rotate Frame

In this frame the message "The quick brown fox jumped over the lazy dogs" is displayed over the entire screen. By displaying the full screen of characters, the quality and distortion of the characters may be checked. Also included in the frame are rotated CHARACTERS. The rotated characters are displayed in the menu area.

M = 15 Character Set, Superscript, Subscript and Italic Frame

The frame displays all the displayable characters, special, italic, superscript and subscript. The first line consists of upper case letter (codes 100-137) and italic uppercase letters. The second line contains lower case letters (codes 140-177) and italic lower case letters. The third line contains numbers and punctuation (codes 40-77) and italic numbers and punctuation. The fourth line contains the special characters and italic special characters. These four lines are repeated in the lower half of the screen. Near the center of the screen a horizontal reference line is displayed.

The largest character scale is enabled and the letter "E" is displayed. This should appear on the base reference line. The code "super-script on" is enabled, followed by another "E".

The procedure is repeated three times with the result being four letter's of "E" with each having a reduced size and an ascending y position. To verify the "superscript-off" function, the code "super-script off" followed by an ASCII "E" is sent. The procedure is repeated three times with the character increasing in size and descending in the y position.

The last "E" should be on the base reference line. The same procedure is repeated using the "subscript-on" and "subscript-off" codes except the characters should first descend with reducing in size followed by ascending and increasing in character size.

N = 16 Sync Speed and Character Terminate Frame

The patterns serves two FUNCTIONS. The first is to test character terminate. A diamond is displayed in the center of the screen with a message about the "SYNC" speed. The message is terminated by the value of '177' (a full dot matrix character).

The code #177 is loaded into the character terminate register and character terminate (character string escape) function is enabled.

The diamond is displayed using the "BASIC Vector" instruction. The message is displayed by entering "character" mode and doing a "display JSR" to the ASCII string. The text should be displayed and a "display POP and RESTORE" should occur after the code #177 is displayed. If "character terminate" fails to cause a 'POP', a DIFFERENT message will be displayed reporting THE FACT.

The second purpose is to verify a visual change in the picture intensity when using NO SYNC, 40 cps sync and 30 cps sync.

The displayed message will indicate the different sync speeds.

When no sync is enabled the frame will appear bright and will have no flicker. When a sync speed of 40 is enabled, the frame will become dim. Upon selection of a sync speed of 30, the frame should appear to flicker. In each case, the frame appears different for each sync speed.

O = 17 Dash Lines and Blink Frame

This is a frame dedicated to the different line types and the ability to generate a blinking element. The type of line followed by two vectors of the same line type are displayed. The first is without blink enabled and the second is displayed with blink enabled. Visually the type of line is displayed followed by a non blinking line of the type followed by a blinking line of the type. This frame also used a "Display jump relative to loop" on the frame.

P = 20 Vector Length. (Spray) Frame

The frame consists of "ABSOLUTE" vectors drawn from point 00 to another x,y point and a return vector to point 0,0. The first vector is drawn from point 0,0 to the maximum x and a y position of 1. Then a INVISIBLE vector to 0,0 is drawn. The third vector is drawn from point 0,0 to the maximum x and a y position of 3. This is repeated until the maximum y position has been displayed(45 DEG.). At that point the sequence is reversed IN that the x is the adjusted end point. The vector is drawn from point 0,0 to a value of x and the maximum value of y. A reference x and y vector is drawn at the right and top edge of the main screen. Each vector should terminate on the reference line. Even spacing should exist between the end of each vector. EVERY OTHER VECTOR WILL BE DISPLAYED.

Q = 21 Horizontal Phosphor Frame

In this frame, a reference box around the main screen perimeter is displayed. A band of intensified vectors are drawn to enable the operator to inspect phosphor surface. The band uses the "BASIC Vector" instruction by going the full value of y (path 2), delta x of 2 units (path 0), negative full value of y (path 6) and a delta x of 2 units. This is repeated 50 times. The origin point of the band is updated via the "Point" instruction. The number of times the band is displayed before moving to the next position is controlled by the number loaded into the "TEMPA".

R = 22 Vertical Phosphor Frame

In this frame, a reference box around the main screen and menu perimeter is displayed. A band of intensified vectors are drawn thru the main screen and the menu screen to enable the operator to inspect the phosphor surface. The band uses "BASIC Vector" instruction by going the full value of x (path 0), delta y of 2 units (path 2), negative full value of x (path 4), and a delta y of 2 units. This is repeated 50 times. THE PROCESS IS THEN REPEATED AGAIN IN THE MENU AREA EXCEPT USING THE MAXIMUM X MENU LENGTH (177).

The origin point of the band is updated via the "Point" instruction. The number of times the band is displayed before moving to the next position is controlled by the number loaded into the "TEMPA".

S = 23 Short Vector and Relative Point Frame

With this frame the operator can verify the correct selection of Relative point and short vectors. Four octagons are drawn in the four quadrants of the screen. Each octagon consists of an outer octagon drawn using the "short vector" instruction. Within each major octagon should be eight points at the intersecting vectors OF THE MAJOR OCTAGON. The "Relative point" instruction is used to display these points. A THIRD OCTAGON IS DISPLAYED USING THE "SHORT VECTOR" INSTRUCTION.

T = 24 GRAPHPLOT INCREMENT REGISTER TEST USING GRAPHPLOT X AND GRAPHPLOT Y

THE GRAPHPLOT INCREMENT REGISTER IS VERIFIED WITH A "SINE WAVE" PATTERN. TWO CYCLES OF A SINE WAVE ARE DISPLAYED IN GRAPHPLOT Y AND GRAPHPLOT X MODES. THE AMOUNT OF INCREMENT BETWEEN POINTS IS A FUNCTION OF THE GRAPHPLOT INCREMENT REGISTER. AT THE END OF THE DISPLAY FILE IS A "DISPLAY STOP". UPON DETECTING THE DSTOP, A COUNTER IS DECREMENTED. UPON EXHAUSTION OF THE COUNTER, THE GRAPHPLOT INCREMENT REGISTER IS CHANGED. THE RESULT IS THE SINE WAVES WILL APPEAR TO EXPAND TO THE RIGHT, FOR GRAPHPLOT Y, AND TO THE TOP, FOR GRAPHPLOT X. ONLY THE LOWER THREE BITS OF THE INCREMENT REGISTER ARE VERIFIED WITH THIS PATTERN.

U = 25 Intensity Level and Lightpen Frame

The frame provides the operator with a method to visually check the eight different intensity levels. Points, Vectors and Characters are drawn using the different intensity levels. The frame also includes handling of "Light-pen" flags and "Light-Pen switches". An octagon is displayed in the upper right corner. Inside the octagon contain the X and Y axis values for the last "Light-Pen Hit". The state of the "Light-Pen switch" is also displayed within the octagon. In the lower right area a matrix of dots is used for a static test of the "Light-Pen field of View". The intensified dots are spaced four units apart. When the dots are detected by the "Light-Pen", the dot which a hit has occurred on will not be displayed. Below the dot matrix is an octal readout reporting the hit count total. The center of the frame is bisected by a Horizontal Reference Line (Y=700). Nine vertical reference lines are drawn at 200 unit increments. The vertical lines are drawn below the Horizontal Reference Line are used to verify correct "X" pen hit position. The lower left section contains vertical spacing test. Three parallel vectors are drawn with decreasing vertical spacing between the lines. The lower center area consists of a Variable Line Length Test. Twenty horizontal lines with increasing X length are drawn from a common X position. Both sections are used to test light pen selectivity. IF THE SECOND CONSOLE IS CONNECTED, VERIFY INDEPENDANT OPERATION OF THE X/Y AND PEN SWITCH READOUT FOR EACH CONSOLE. THE "FIELD OF VIEW" AND THE "HIT-COUNT" ARE THE ONLY DEPENDANT ELEMENTS.

V = 26 KEYBOARD CHARACTER ECHO LOOP (OPERATOR SELECT ONLY)

SEQ 0014

***** TYPE <V> OR SWR = 426 TO RUN THIS FRAME *****

THE FRAME PROVIDES A KEYBOARD TO VS60 SCREEN CHARACTER LOOP TO VERIFY PROPER OPERATION OF THE CONSOLE KEYBOARD. A MAXIMUM OF 1024 CHARACTERS CAN BE DISPLAYED BY THIS LOOP. THE OPERATOR MAY ESCAPE THE LOOP, BY DEPRESSING THE "CTRL" AND "C" KEYS, TO RETURN TO THE DIRECTORY FRAME. UPON DETECTION OF A KEYBOARD CHARACTER, THE CHARACTER'S OCTAL VALUE AND THE CHARACTER ARE DISPLAYED ON THE SCREEN.

THE "SHIFT-OUT" CODE CAN BE ENTERED BY THE OPERATOR, HOWEVER THE PROGRAM WILL NOT USE ANY KEYBOARD CODES GREATER THAN 37 OCTAL. UPON ENTERING A "SHIFT-OUT" MODE, THE CHARACTER DISPLAYED FROM THE CURRENT CHARACTER POSITION TO THE END OF THE LINE WILL APPEAR TO BE AN UPSIDE DOWN "Y" CHARACTER. IN THE "SHIFT-OUT" MODE, THE CHARACTER DISPLAYED HAS THE VALUE OF ZERO.

W = 27 DYNAMIC EXTERNAL STOP FRAME

***** THIS FRAME VERIFIES VS60 ECO'S VT48 #7, *****
 ***** M7054 #4 AND M7058 #5 HAVE BEEN INSTALLED *****

THIS FRAME VERIFIES PROPER OPERATION OF THE EXTERNAL DISPLAY STOP LOGIC. A FRAME CONTAINING MOST OF THE VS60 INSTRUCTIONS IS DISPLAYED. WHILE THE VS60 IS DISPLAYING THE FRAME, THE -11 CPU IS RANDOMLY GENERATION A EXTERNAL DISPLAY STOP SIGNAL (EDSS) TO THE VS60. AFTER AN "EDSS" HAS BEEN SENT, THE -11 WILL VERIFY THE DISPLAY PROGRAM COUNTER REGISTER TO BE WITHIN AN EXPECTED RANGE. THE GENERATION OF AN "EDSS" SHOULD CAUSE AN EXTERNAL STOP INTERRUPT. UPON DETECTING AN "EDSS" INTERRUPT, A COUNTER IS DECREMENTED. IF THE COUNTER DOES NOT GO TO 0, THE PROGRAM WILL ISSUE A "RESUME" TO THE VS60. IF THE COUNTER BECOMES 0, THE PROGRAM WILL GO TO THE "END OF PASS" AND RESTART THE PROGRAM. SEVEN DIFFERENT ERROR CONDITIONS WILL BE VISUALLY REPORTED WITH THIS SUB-TEST:

ERROR #	REASON
-----	-----
0	NO EXTERNAL STOP INTERRUPT
1	UNEXPECTED INTERRUPT TO VECTOR +4
2	UNEXPECTED INTERRUPT TO VECTOR +10
3	UNEXPECTED INTERRUPT TO VECTOR +14
4	D.P.C. OUT OF RANGE (TOO LOW)
5	D.P.C. OUT OF RANGE (TOO HIGH)
6	EXTERNAL STOP INTERRUPT BUT NO EXTERNAL STOP FLAG OR DISPLAY STOP FLAG.

THIS FRAME PROVIDES AN OVERALL CORRELATOR ALIGNMENT VERIFICATION. VARIOUS ALIGNMENT DATA ARE PROVIDED IN THE MENU AREA.

A 6 X 5 DOT ARRAY IS DISPLAYED ON THE SCREEN. THE CORRELATOR IS DRIVEN TO THE VICINITY OF EACH POINT AND ALLOWED TO ACCUMULATE "HITS". EACH POINT WILL YIELD UP TO 100 (DECIMAL) HITS. EACH POINT IS TURNED OFF IF IT HAS BEEN HIT AT LEAST ONCE (1%). THOSE POINTS REMAINING ILLUMINATED AT THE END OF THE FRAME CYCLE WERE NOT VISIBLE TO THE CORRELATOR, AND INDICATE THAT SOME ADJUSTMENT IS REQUIRED.

AT THE END OF EACH TEST CYCLE, AN AVERAGE VISIBILITY PERCENTAGE IS DISPLAYED, AND "ALIGNMENT ACCEPTABLE" OR "ADJUSTMENT REQUIRED" AS APPROPRIATE.

IF ADJUSTMENT IS DEEMED NECESSARY, TYPE <Y> (OR RESTART WITH SWR = 431) TO START THE ADJUSTMENT FRAME.

Y = 31 POINT CORRELATOR ADJUSTMENT FRAME(S) (MANUAL).

SEQ 0016

***** TYPE <Y> OR SWR = 431 TO RUN THIS FRAME *****

THESE FRAMES ARE USED TO PERFORM POINT CORRELATOR
ADJUSTMENTS IN TWO PARTS AS FOLLOWS:

PART 1 -- APPERTURE SIZE AND X/Y OFFSET.
A 100 SQUARE DOT ARRAY IS DISPLAYED AT CENTER SCREEN.
THE CORRELATOR POINTS TO THE CENTER OF THE ARRAY AND
THOSE POINTS VISIBLE TO THE CORRELATOR ARE TURNED OFF.
THIS RESULTS IN A DARKENED AREA WHICH REPRESENTS THE
ACTUAL APPERTURE SIZE, SHAPE, AND POSITION.
APPERTURE SPECS (EXPECTED VS ACTUAL SIZE) ARE DISPLAYED
IN THE MENU AREA. ANY CENTERING ERROR IS DISPLAYED AS
X/Y OFFSET. ADJUST SIZE AND OFFSET (ON A023 BOARD) IN
ACCORDANCE WITH THE INSTRUCTIONS PROVIDED ON THE SCREEN.
TYPE <X> OR <Y> TO PROCEED TO PART 2.

PART 2 -- X/Y SCALE ADJUSTMENT AND FINE-TUNING.
THE APPERTURE VIEWING ARRAY IS DUPLICATED AT
LEFT-CENTER AND RIGHT-CENTER FOR X AXIS ADJUSTMENT, OR
BOTTOM-CENTER AND TOP-CENTER FOR Y AXIS ADJUSTMENT.
THE CENTERING ERROR AT EACH POSITION IS CALCULATED.
THESE ERRORS ARE THEN EVALUATED IN TERMS OF THEIR
'OFFSET' AND 'SCALE' COMPONENTS, AND DISPLAYED AS SUCH.
ADJUST SCALE AND FINE-TUNE OFFSET IN ACCORDANCE WITH
THE INSTRUCTIONS PROVIDED ON THE SCREEN.

TYPE <X> OR <Y> TO SWAP THE ADJUSTMENT AXES.
TYPE <C> TO RETURN TO PART 1 (CENTER SCREEN), OR
TYPE <CR> TO RUN THE VERIFICATION FRAME (TEST 30) IN
MANUAL MODE, FOR OVERALL VERIFICATION.

NOTE THAT SUCCESSIVE <CR>'S WILL TOGGLE BACK AND FORTH
BETWEEN THE ADJUST AND VERIFY FRAMES. THERE IS NO
AUTOMATIC EXIT FROM EITHER. WHEN YOU'RE SATISFIED
WITH THE CORRELATOR PERFORMANCE, TYPE <^C> TO RESTART
THE MAIN PROGRAM.

ALSO NOTE THAT GRAPHICS CO-ORDINATE DATA ARE DISPLAYED
AS OCTAL VALUES. ALL OTHER DATA ARE DECIMAL.

***** TYPE <Z> OR SWR = 432 TO RUN THIS FRAME *****

THIS FRAME IS USED TO VERIFY THAT THE WRITING TABLET INTERFACE (DL11XX) FUNCTIONS CORRECTLY. THERE ARE NO ERROR INDICATIONS OF ANY KIND. THE FRAME IS BY-PASSED IF THERE IS NO TABLET INTERFACE INSTALLED.

A SET OF NESTED BOXES AND A TRACKING BALL (CURSOR) IS DISPLAYED ON THE SCREEN. THE BALL IS SLAVED TO THE TABLET PEN X/Y CO-ORDINATES, AND PROVIDES VISUAL FEEDBACK OF THE ACTUAL PEN POSITION.

THE TABLET PEN CO-ORDINATES, TIP SWITCH (UP/DOWN), AND PROXIMITY (IN/OUT) ARE DISPLAYED IN THE MENU AREA. THE VECTORS MAKING UP THE BOXES ARE 'HIT SENSITIVE' AS ARE ALL CHARACTERS IN THE MENU AREA.

ON ANY 'PC HIT', THE HIT COUNT IN THE MENU IS UPDATED. IF THE PEN TIP SWITCH IS DOWN (CLOSED) AND THE X CO-ORD IS WITHIN THE MAJOR VIEWING AREA, THE POINT OF IMPACT IS MARKED WITH A SMALL 'X' AND THE HIT CO-ORDINATES ARE DISPLAYED.

AS BEFORE, THERE IS NO EXIT FROM THIS FRAME. TYPE <^C> WHEN YOU'RE FINISHED TO RETURN TO THE DIRECTORY FRAME.

10. LISTING

PROGRAM LISTING FOLLOWS.

17		BASIC DEFINITIONS
27		OPERATIONAL SWITCH SETTINGS
28		TRAP CATCHER
(1)		STARTING ADDRESS(ES)
30		ACT11 HOOKS
32		APT PARAMETER BLOCK
33		COMMON TAGS
(2)		APT MAILBOX-ETABLE
(1)		ERROR POINTER TABLE
38		OPERATOR VARIABLE LOCATIONS
44		VS-60 ADDRESSES AND INTERRUPT VECTORS
74		INITIAL PROGRAM STARTUP ROUTINE
77		INITIALIZE THE COMMON TAGS
137		KEYBOARD SERVICE ROUTINE
179		VS-60 INSTRUCTION SET
291		
292	TEST	LETTER DESCRIPTION
293	----	-----
294		
296	T1	A DIRECTORY FRAME
303	T2	B ASTIGMATISM AND SETTLING TIME
310	T3	C SHORT TERM DRIFT
361	T4	D MINOR AXIS GAIN, OFFSET AND PHASE ADJUSTMENT
370	T5	E MAJOR AXIS OFFSET AND VECTOR START POINT ADJUSTMENT
382	T6	F VECTOR LENGTH GAIN, CONVERGENCE AND VECTOR LINEARITY
458	T7	G PINCUSHION FRAME
501	T10	H OCTAGONS AND CIRCLES
506	T11	I SCISSORING AND VECTOR SCALING
523	T12	J OFFSET X AND OFFSET Y POSITION
617	T13	K CHARACTER SCALE FRAME
627	T14	L CHARACTER QUALITY AND CHARACTER ROTATE IN THE MENU
646	T15	M CHARACTER SET, SUPERScript, SUBSCRIPT AND ITALICS
767	T16	N SYNC SPEED AND CHARACTER TERMINATE TEST
800	T17	O DASH LINES AND BLINK
805	T20	P VECTOR SPRAY (LENGTH) TEST
833	T21	Q HORIZONTAL PHOSPHOR TEST
849	T22	R VERTICAL PHOSPHOR TEST
876	T23	S SHORT VECTOR AND RELATIVE POINT
927	T24	T GRAPHPLOT INCREMENT REGISTER TEST USING GRAPHPLOT X AND Y
936	T25	U INTENSITY LEVEL AND LIGHT PEN TEST
973	T26	V KEYBOARD CHARACTER ECHO LOOP (MANUAL)
1034	T27	W DYNAMIC EXT. DISPLAY STOP
1123	T30	X POINT CORRELATOR ALIGNMENT CHECK
1263	T31	Y POINT CORRELATOR ADJUSTMENT (MANUAL)
1830	T32	Z WRITING TABLET EXERCISER FRAME (MANUAL)
2058		END OF PASS ROUTINE
2060		SUBROUTINE FOR VERT. LIGHT PEN FIELD OF VIEW
2068		SUBROUTINE FOR HORIZ. LIGHT PEN FIELD OF VIEW
2078		LIGHT-PEN INTERRUPT SERVICE
2167		DISPLAY SUB-ROUTINE
2205		UPDATE OCTAL READOUT OF THE X-Y FOR LIGHT PEN HIT
2232		X - Y POSITIONS FOR THE SHORT TERM DRIFT TEST
2264		
2265		
2266		
2267		

2268	
2269	SUB-PICTURES
2270	
2271	DIRECTORY SUB-PICTURE
2358	X AND Y POSITIONS FOR THE SETTLING TEST
2510	MENU 1 SUB-PICTURE
2635	DRAW 10 VERTICAL VECTORS IN THE LEFT CENTER AERA
2651	DRAW THE DELAY INTENSITY SUB-PICTURE IN THE LEFT CENTER AERA
2704	OCTAGONS USING LONG AND ABSOLUTE VECTORS (WIDTHS OF 7,77,177,377 AND 520)
2705	CIRCLES USING ABSOLUTE VECTORS (WIDTHS OF 64., 128., AND 256.)
2952	X AND Y OFFSET SUB-PICTURE
2970	SUPER AND SUBSCRIPT SUB-PICTURE
3005	SUPER AND SUBSCRIPT ASCII STRING
3019	SYNC SPEED SUBPICTURE
3057	DASH LINE SUB-PICTURE
3121	VECTOR LENGTH SUB-PICTURE
3146	HORIZONTAL PHOSPHOR SUB-PICTURE
3164	MAIN VERTICAL PHOSPHOR SUB-PICTURE
3181	MENU VERTICAL PHOSPHOR SUB-PICTURE
3244	SHORT VECTOR AND RELATIVE POINT SUB-PICTURE
3261	GRAPHPLOT INCREMENT SUB-PICTURE
3300	DATA STRING FOR A SINE WAVE
3312	SHORT TERM DRIFT SUB-PICTURE
3331	SCREEN SCISSORING SUB-PICTURE
3406	VECTOR SCALE SUB-PICTURE
3428	VECTOR STARTING SUB-PICTURE
3480	MAJOR AXIS OFFSET SUB-PICTURE
3542	CHARACTER SCALE SUB-PICTURE
3654	ROTATE CHARACTERS SUBPICTURE
3718	
3719	LIGHT-PEN SUBPICTURE
3720	
3723	POSITION THE OCTAGON
3730	DISPLAY ON CONSOLE #0 THE X-Y READOUT VALUE
3748	DISPLAY ON CONSOLE #1 THE X-Y READOUT VALUE
3767	DISPLAY HIT COUNT MESSAGE
3791	HORIZONTAL REF. LINE SECTION
3837	VERTICAL SPACEING SECTION
3856	VARIABLE HORIZ. LINE LENGTH
3877	INTENSITY LEVEL SECTION OF LIGHT PEN TEST
4001	DRAW OUTER REFERENCE BOX
4019	
4020	KEYBOARD CHARACTER ECHO SUB-PICTURE
4040	
4041	DYNAMIC EXT. STOP FRAME
4042	
4270	POINT CORRELATOR DISPLAY FILES.
4645	SCOPE HANDLER ROUTINE


```

(1)      100000      SW15= 100000
(1)      040000      SW14= 40000
(1)      020000      SW13= 20000
(1)      010000      SW12= 10000
(1)      004000      SW11= 4000
(1)      002000      SW10= 2000
(1)      001000      SW09= 1000
(1)      000400      SW08= 400
(1)      C00200      SW07= 200
(1)      000100      SW06= 100
(1)      000040      SW05= 40
(1)      000020      SW04= 20
(1)      0C0010      SW03= 10
(1)      000004      SW02= 4
(1)      000002      SW01= 2
(1)      000001      SW00= 1
(1)      .EQUIV      SW09,SW9
(1)      .EQUIV      SW08,SW8
(1)      .EQUIV      SW07,SW7
(1)      .EQUIV      SW06,SW6
(1)      .EQUIV      SW05,SW5
(1)      .EQUIV      SW04,SW4
(1)      .EQUIV      SW03,SW3
(1)      .EQUIV      SW02,SW2
(1)      .EQUIV      SW01,SW1
(1)      .EQUIV      SW00,SW0

(1)      ;*DATA BIT DEFINITIONS (BIT00 TO BIT15)
(1)      100000      BIT15= 100000
(1)      040000      BIT14= 40000
(1)      020000      BIT13= 20000
(1)      010000      BIT12= 10000
(1)      004000      BIT11= 4000
(1)      002000      BIT10= 2000
(1)      001000      BIT09= 1000
(1)      000400      BIT08= 400
(1)      000200      BIT07= 200
(1)      000100      BIT06= 100
(1)      000040      BIT05= 40
(1)      000020      BIT04= 20
(1)      000010      BIT03= 10
(1)      000004      BIT02= 4
(1)      000002      BIT01= 2
(1)      000001      BIT00= 1
(1)      .EQUIV      BIT09,BIT9
(1)      .EQUIV      BIT08,BIT8
(1)      .EQUIV      BIT07,BIT7
(1)      .EQUIV      BIT06,BIT6
(1)      .EQUIV      BIT05,BIT5
(1)      .EQUIV      BIT04,BIT4
(1)      .EQUIV      BIT03,BIT3
(1)      .EQUIV      BIT02,BIT2
(1)      .EQUIV      BIT01,BIT1
(1)      .EQUIV      BIT00,BIT0

(1)      ;*BASIC "CPU" TRAP VECTOR ADDRESSES
    
```



```

(1) 000004 ERRVEC= 4 ;;TIME OUT AND OTHER ERRORS
(1) 000010 RESVEC= 10 ;;RESERVED AND ILLEGAL INSTRUCTIONS
(1) 000014 TBITVEC=14 ;;"T" BIT
(1) 000014 TRTVEC= 14 ;;TRACE TRAP
(1) 000014 BPTVEC= 14 ;;BREAKPOINT TRAP (BPT)
(1) 000020 IOTVEC= 20 ;;INPUT/OUTPUT TRAP (IOT) **SCOPE**
(1) 000024 PWRVEC= 24 ;;POWER FAIL
(1) 000030 EMTVEC= 30 ;;EMULATOR TRAP (EMT) **ERROR**
(1) 000034 TRAPVEC=34 ;;"TRAP" TRAP
(1) 000060 TKVEC= 60 ;;TTY KEYBOARD VECTOR
(1) 000064 TPVEC= 64 ;;TTY PRINTER VECTOR
(1) 000240 PIRQVEC=240 ;;PROGRAM INTERRUPT REQUEST VECTOR

18
19 172000 ABASE= 172000 ; DISPLAY PC ADDRESS.
20 100320 AVECT1= 100320 ; 1ST OF 4 DISPLAY VECTORS.
21 000200 APRIOR= 200
22 *****
23 175620 ACDW1= 175620 ; WRITING TABLET DL ADDRESS.
24 000310 ACDW2= 310 ; TABLET INTERRUPT VECTOR.
25 *****
26
27 .SBTTL OPERATIONAL SWITCH SETTINGS
(1) *
(1) * SWITCH USE
(1) * -----
(1) * 14 LOOP ON TEST
(1) * 9 STOP SUB-PICTURE MOTION
(1) * 8 LOOP ON TEST IN SWR<7:0>
28 .SBTTL TRAP CATCHER
(1)
(1) 000000 .=0
(1) ;*ALL UNUSED LOCATIONS FROM 4 - 776 CONTAIN A ".+2,HALT"
(1) ;*SEQUENCE TO CATCH ILLEGAL TRAPS AND INTERRUPTS
(1) ;*LOCATION 0 CONTAINS 0 TO CATCH IMPROPERLY LOADED VECTORS
(1)
(1) 000174 .=174
(1) 000174 000000 DISPREG: .WORD 0 ;;SOFTWARE DISPLAY REGISTER
(1) 000176 000000 SWREG: .WORD 0 ;;SOFTWARE SWITCH REGISTER
(1)
(1) .SBTTL STARTING ADDRESS(ES)
(1) 000200 000137 001344 JMP @#BEGIN ;;JUMP TO STARTING ADDRESS OF PROGRAM
    
```



```

30      .SBTTL ACT11 HOOKS
(1)
(2)      ::*****
(1)      ;HOOKS REQUIRED BY ACT11
(1)      000204      $$VPC=.          ;SAVE PC
(1)      000046      .=46
(1) 000046 013756      $ENDAD          ;;1)SET LOC.46 TO ADDRESS OF $ENDAD IN .$EOP
(1)      000052      .=52
(1) 000052 000000      .WORD 0          ;;2)SET LOC.52 TO ZERO
(1)      000204      .=$$VPC          ;; RESTORE PC
(1)      001000      .=1000
31
32      .SBTTL APT PARAMETER BLOCK
(1)
(2)      ::*****
(1)      ;SET LOCATIONS 24 AND 44 AS REQUIRED FOR APT
(2)      ::*****
(1)      001000      .$X=.          ;;SAVE CURRENT LOCATION
(1)      000024      .=24          ;;SET POWER FAIL TO POINT TO START OF PROGRAM
(1) 000024 000200      200          ;;FOR APT START UP
(1)      000044      .=44          ;;POINT TO APT INDIRECT ADDRESS PNTR.
(1) 000044 001000      $APTHDR      ;;POINT TO APT HEADER BLOCK
(1)      001000      .=.$X          ;;RESET LOCATION COUNTER
(2)      ::*****
(1)      ;SETUP APT PARAMETER BLOCK AS DEFINED IN THE APT-PDP11 DIAGNOSTIC
(1)      ;INTERFACE SPEC.
(1)
(1) 001000      $APTHD:
(1) 001000 000000      $HIPTS: .WORD 0          ;;TWO HIGH BITS OF 18 BIT MAILBOX ADDR.
(1) 001002 001172      $MBADR: .WORD $MAIL      ;;ADDRESS OF APT MAILBOX (BITS 0-15)
(1) 001004 000020      $TSTM: .WORD 20          ;;RUN TIM OF LONGEST TEST
(1) 001006 000300      $PASTM: .WORD 300        ;;RUN TIME IN SECS. OF 1ST PASS ON 1 UNIT (QUICK VERIFY)
(1) 001010 000000      $UNITM: .WORD 0          ;;ADDITIONAL RUN TIME (SECS) OF A PASS FOR EACH ADDITIONAL UNIT
(1) 001012 000032      .WORD $ETEND-$MAIL/2 ;;LENGTH MAILBOX-ETABLE(WORDS)
    
```


(2)	001206	000000	\$MSGAD: .WORD	AMSGAD	::MESSAGE ADDRESS
(2)	001210	000000	\$MSGLG: .WORD	AMSGLG	::MESSAGE LENGTH
(2)	001212		\$ETABLE:		::APT ENVIRONMENT TABLE
(2)	001212	000	\$ENV: .BYTE	AENV	::ENVIRONMENT BYTE
(2)	001213	000	\$ENVM: .BYTE	AENVM	::ENVIRONMENT MODE BITS
(2)	001214	000000	\$SWREG: .WORD	ASWREG	::APT SWITCH REGISTER
(2)	001216	000000	\$USWR: .WORD	AUSWR	::USER SWITCHES
(2)	001220	000000	\$CPUOP: .WORD	ACPUOP	::CPU TYPE,OPTIONS
(2)			:*		BITS 15-11=CPU TYPE
(2)			:*		11/04=01,11/05=02,11/20=03,11/40=04,11/45=05
(2)			:*		11/70=06,PDQ=07,Q=10
(2)			:*		BIT 10=REAL TIME CLOCK
(2)			:*		BIT 9=FLOATING POINT PROCESSOR
(2)			:*		BIT 8=MEMORY MANAGEMENT
(2)	001222	000	\$MAMS1: .BYTE	AMAMS1	::HIGH ADDRESS,M.S. BYTE
(2)	001223	000	\$MTYP1: .BYTE	AMTYP1	::MEM. TYPE,BLK#1
(2)			:*		MEM.TYPE BYTE -- (HIGH BYTE)
(2)			:*		900 NSEC CORE=001
(2)			:*		300 NSEC BIPOLAR=002
(2)			:*		500 NSEC MOS=003
(2)	001224	000000	\$MADR1: .WORD	AMADR1	::HIGH ADDRESS,BLK#1
(2)			:*		MEM.LAST ADDR.=3 BYTES,THIS WORD AND LOW OF "TYPE" ABOVE
(2)	001226	000	\$MAMS2: .BYTE	AMAMS2	::HIGH ADDRESS,M.S. BYTE
(2)	001227	000	\$MTYP2: .BYTE	AMTYP2	::MEM. TYPE,BLK#2
(2)	001230	000000	\$MADR2: .WORD	AMADR2	::MEM.LAST ADDRESS,BLK#2
(2)	001232	000	\$MAMS3: .BYTE	AMAMS3	::HIGH ADDRESS,M.S.BYTE
(2)	001233	000	\$MTYP3: .BYTE	AMTYP3	::MEM. TYPE,BLK#3
(2)	001234	000000	\$MADR3: .WORD	AMADR3	::MEM.LAST ADDRESS,BLK#3
(2)	001236	000	\$MAMS4: .BYTE	AMAMS4	::HIGH ADDRESS,M.S.BYTE
(2)	001237	000	\$MTYP4: .BYTE	AMTYP4	::MEM. TYPE,BLK#4
(2)	001240	000000	\$MADR4: .WORD	AMADR4	::MEM.LAST ADDRESS,BLK#4
(2)	001242	100320	\$VECT1: .WORD	AVECT1	::INTERRUPT VECTOR#1,BUS PRIORITY#1
(2)	001244	000000	\$VECT2: .WORD	AVECT2	::INTERRUPT VECTOR#2BUS PRIORITY#2
(2)	001246	172000	\$BASE: .WORD	ABASE	::BASE ADDRESS OF EQUIPMENT UNDER TEST
(2)	001250	000000	\$DEVN: .WORD	ADEVN	::DEVICE MAP
(2)	001252	175620	\$CDW1: .WORD	ACDW1	::CONTROLLER DESCRIPTION WORD#1
(2)	001254	000310	\$CDW2: .WORD	ACDW2	::CONTROLLER DESCRIPTION WORD#2
(2)	001256		\$ETEND:		
(2)			.MEXIT		


```

76 001344 000005 BEGIN: RESET
77 .SBTTL INITIALIZE THE COMMON TAGS
(1) ;;CLEAR THE COMMON TAGS ($CMTAG) AREA
(1) 001346 012706 001100 MOV #CMTAG,R6 ;;FIRST LOCATION TO BE CLEARED
(1) 001352 005026 CLR (R6)+ ;;CLEAR MEMORY LOCATION
(1) 001354 022706 001140 CMP #SWR,R6 ;;DONE?
(1) 001360 001374 BNE -6 ;;LOOP BACK IF NO
(1) 001362 012706 001100 MOV #STACK,SP ;;SETUP THE STACK POINTER
(1) ;;INITIALIZE A FEW VECTORS
(1) 001366 012737 036404 000020 MOV #SCOPE,@IOTVEC ;;IOT VECTOR FOR SCOPE ROUTINE
(1) 001374 012737 000340 000022 MOV #340,@IOTVEC+2 ;;LEVEL 7
(1) 001402 013737 013742 013734 MOV $ENDCT,$EOPCT ;;SETUP END-OF-PROGRAM COUNTER
(1) 001410 012737 001410 001106 MOV #,$SLPADR ;;INITIALIZE THE LOOP ADDRESS FOR SCOPE
(2) ;;SIZE FOR A HARDWARE SWITCH REGISTER. IF NOT FOUND OR IT IS
(2) ;;EQUAL TO A "-1", SETUP FOR A SOFTWARE SWITCH REGISTER.
(2) 001416 013746 000004 MOV @ERRVEC,-(SP) ;;SAVE ERROR VECTOR
(2) 001422 012737 001456 000004 MOV #64$,@ERRVEC ;;SET UP ERROR VECTOR
(2) 001430 012737 177570 001140 MOV #DSWR,SWR ;;SETUP FOR A HARDWARE SWICH REGISTER
(2) 001436 012737 177570 001142 MOV #DDISP,DISPLAY ;;AND A HARDWARE DISPLAY REGISTER
(2) 001444 022777 177777 177466 CMP #-1,@SWR ;;TRY TO REFERENCE HARDWARE SWR
(2) 001452 001012 BNE 66$ ;;BRANCH IF NO TIMEOUT TRAP OCCURRED
(2) ;;AND THE HARDWARE SWR IS NOT = -1
(2) 001454 000403 BR 65$ ;;BRANCH IF NO TIMEOUT
(2) 001456 012716 001464 64$: MOV #65$,(SP) ;;SET UP FOR TRAP RETURN
(2) 001462 000002 RTI
(2) 001464 012737 000176 001140 65$: MOV #SWREG,SWR ;;POINT TO SOFTWARE SWR
(2) 001472 012737 000174 001142 MOV #DISPREG,DISPLAY
(2) 001500 012637 000004 66$: MOV (SP)+,@ERRVEC ;;RESTORE ERROR VECTOR
(1)
(2) 001504 005037 001200 CLR $PASS ;;CLEAR PASS COUNT
(2) 001510 132737 000200 001213 BITB #APTSIZE,$ENVM ;;TEST USER SIZE UNDER APT
(2) 001516 001403 BEQ 67$ ;;YES,USE NON-APT SWITCH
(2) 001520 012737 001214 001140 MOV #SSWREG,SWR ;;NO,USE APT SWITCH REGISTER
(2) 001526 67$:
78 001526 012700 001264 RESTAT: MOV #DPC,R0 ;GET POINTER
79 001532 013701 001246 MOV $BASE,R1 ;GET SUPPLIED ADDRESS
80 001536 010120 1$: MOV R1,(R0)+ ;UPDATE
81 001540 062701 000002 ADD #2,R1 ;THE
82 001544 022700 001316 CMP #DDONE,R0 ;ADDRESSES
83 001550 001372 BNE 1$ ;UNTIL DONE
84 001552 012700 001316 MOV #DDONE,R0 ;GET POINTER
85 001556 013701 001242 MOV $VECT1,R1 ;GET SUPPLIED VECTOR
86 001562 042701 160000 BIC #160000,R1 ;CLEAR PSW BITS
87 001566 010120 2$: MOV R1,(R0)+ ;UPDATE
88 001570 062701 000002 ADD #2,R1 ;THE VECTORS
89 001574 022700 001336 CMP #DDONE+20,R0
90 001600 001372 BNE 2$
91 *****
92 001602 013737 001252 001336 MOV $CDW1,TABL ; SET TABLET ADDRESS...
93 001610 000240 000240 240, 240
94 001614 013737 001254 001340 MOV $CDW2,TBVCT ;...AND VECTOR ADDRESS.
95 001622 013737 001340 001342 MOV TBVCT,TBVCT1
96 001630 062737 000002 001342 ADD #2,TBVCT1
97 *****
98 001636 005037 014634 CLR SWITCH ;HOUSEKEEP
99 001642 005037 014474 CLR HOLD ;
    
```



```

100 001646 005004          CLR      R4          ;
101 001650 005037 014476  CLR      TSAVE      ;
102 001654 013777 001262 177376 MOV     TKBVT1,@TKBVCT ;RESET KRB VECTOR
103 001662 005077 177374  CLR      @TKBVT1
104 001666 005037 002322  CLR      KRBD       ; SET AUTO MODE.
105 001672 105777 177242  TSTB   @SWR        ;TEST FOR 'KRB' CONTROL
106 001676 001410          BEQ     3$          ;BR IF NOT
107 001700 005137 002322  COM     KRBD       ; SET 'KRB' CONTROL MODE
108 001704 012777 002104 177346 MOV     #RETB,@TKBVCT ;SET UP 'KRB' INT
109 001712 012777 000340 177342 MOV     #340,@TKBVT1
110 001720 004737 001730 3$:     JSR     PC,FIXVCT ;LOAD BUS VECTORS
111 001724 000137 002324  JMP     TST1       ;START TESTING
112
113 001730 012777 014556 177360 FIXVCT: MOV     #STOPI,@DDONE ;SET UP VS-60 DONE VECTOR
114 001736 113700 001243  MOVB   $VECT1+1,R0 ;GET BR LEVEL
115 001742 042700 177400  BIC    #177400,R0  ;MASK OFF OTHER BITS
116 001746 010077 177346  MOV     R0,@DDONE1
117 001752 013777 001324 177342 MOV     LPVCT1,@LPVCT ;RESET LIGHT-PEN VECTOR
118 001760 005077 177340  CLR    @LPVCT1
119 001764 013777 001330 177334 MOV     TMEVT1,@TIMEVT ;RESET TIME-OUT/SHIFT OUT VECTOR
120 001772 005077 177332  CLR    @TMEVT1
121 001776 013777 001334 177326 MOV     NAMEV1,@NAMEVT ;RESET NAME MATCH VECTOR
122 002004 005077 177324  CLR    @NAMEV1
123 ;*****
124 002010 013777 001342 177322 MOV     TBVCT1,@TBVCT ; RESET TABLET VECTOR.
125 002016 005077 177320  CLR    @TBVCT1
126 ;*****
127 002022 012737 030060 027340 MOV     #30060,DLT14A ;INIT X READOUT VALUE FOR CONSOLE #0
128 002030 012737 030060 027342 MOV     #30060,DLT14A+2
129 002036 012737 030060 027352 MOV     #30060,DLT14B ;INIT Y READOUT
130 002044 012737 030060 027354 MOV     #30060,DLT14B+2
131 002052 012737 030060 027410 MOV     #30060,DLT14C ;RESET READOUT VALUE FOR CONSOLE #1
132 002060 012737 030060 027412 MOV     #30060,DLT14C+2
133 002066 012737 030060 027422 MOV     #30060,DLT14D
134 002074 012737 030060 027424 MOV     #30060,DLT14D+2
135 002102 000207          RTS     PC          ;EXIT
136
137 .SBTTL  KEYBOARD SERVICE ROUTINE
138 002104 117737 177036 014476 RETB:  MOVB   @STKB,TSAVE ;READ THE CHARACTER
139 002112 042737 177600 014476 BIC    #177600,TSAVE ;MASK TO 7 BITS
140 002120 022737 000003 014476 CMP     #3,TSAVE    ;TEST FOR 'CTRL C'
141 002126 001005          BNE    7$          ;BR IF NOT
142 002130 005777 177132  TST    @DSR
143 002134 100375          BPL    .-4        ; WAIT FOR DISPLAY STOP...
144 002136 000137 001344          JMP    BEGIN      ;...AND RE-START.
145 002142 022737 000015 014476 7$:   CMP     #15,TSAVE ;TEST FOR 'CR'
146 002150 001454          BEQ    5$          ;BR IF
147 002152 005037 014634  CLR    SWITCH     ;CLEAR "SWITCH"
148 002156 162737 000101 014476 SUB     #101,TSAVE ;MAKE 0-77
149 002164 100443          BMI    4$          ;<A
150 002166 022737 000032 014476 1$:   CMP     #32,TSAVE
151 002174 100427          BMI    3$          ;>Z
152 002176 013704 014476  MOV     TSAVE,R4
153 002202 110437 001102  MOVB   R4,$STSTNM ;LOAD TEST #
154 002206 006304          ASL    R4
155 002210 005037 014634  CLR    SWITCH
    
```



```

156 002214 005037 014474 CLR HOLD
157 002220 005777 177042 TST @DSR
158 002224 100375 BPL -.4 ; WAIT FOR DISPLAY STOP.
159 002226 000005 RESET
160 002230 004737 001730 JSR PC, FIXVCT ; RESET DISPLAY VECTORS
161 002234 005764 036570 TST DISPTC(R4) ; TEST IF VALID
162 002240 001001 BNE 2$
163 002242 005004 CLR R4
164 002244 012706 001100 2$: MOV #STACK, SP ; RESET STACK.
165 002250 000174 036570 JMP @DISPTC(R4) ; EXIT TO THAT TEST SELECTED
166
167 002254 022737 000076 014476 3$: CMP #76, TSAVE
168 002262 001013 BNE 6$
169 002264 012737 177777 014474 MOV #-1, HOLD ; RUBOUT
170 002272 000002 RTI ; EXIT
171 002274 005037 014474 4$: CLR HOLD
172 002300 000002 RTI
173 002302 012737 177777 014634 5$: MOV #-1, SWITCH ; SET "STOP MOTION" FLAG
174 002310 000002 RTI
175 002312 162737 000040 014476 6$: SUB #40, TSAVE ; CONVERT LC TO UC
176 002320 000721 BR 1$
177 002322 000000 KRBD: 0
  
```

```

178
179 .SBTTL VS-60 INSTRUCTION SET
180
181 100000 CHAR=100000 ; DISPLAY IN CHARACTER MODE
182 104000 SHORTV=104000 ; SHORT VECTOR
183 110000 LONGV=110000 ; LONG VECTOR MODE
184 114000 POINT=114000 ; POINT MODE
185 120000 GRAPHX=120000 ; GRAPH PLOT X MODE
186 124000 GRAPHY=124000 ; GRAPH PLOT Y MODE
187 120000 BASICV=GRAPHX ; BASIC VECTOR MODE
188 130000 RELATP=130000 ; RELATIVE POINT MODE
189 134000 BASICS=RELATP!4000 ; BASIC SHORT VECTOR MODE
190 144000 ABSVCT=144000 ; ABSOLUTE VECTOR MODE
191
192 010000 OFFST0=10000 ; ENABLE OFFSET OF 0
193 012000 OFFST1=12000 ; ENABLE OFFSET OF 1
194 014000 OFFST2=14000 ; ENABLE OFFSET OF 2
195 016000 OFFST3=16000 ; ENABLE OFFSET OF 3
196
197 002000 INTO=2000 ; ENABLE INTENSITY LEVEL 0
198 002200 INT1=2200 ; 1
199 002400 INT2=2400 ; 2
200 002600 INT3=2600 ; 3
201 003000 INT4=3000 ; 4
202 003200 INT5=3200 ; 5
203 003400 INT6=3400 ; 6
204 003600 INT7=3600 ; LEVEL 7
205
206 000100 LPOFF=100 ; DISABLE LIGHT-PEN INTERRUPT
207 000140 LPON=140
208 000020 BLKOFF=20 ; DISABLE BLINK
209 000030 BLKON=30 ; ENABLE BLINK
210
211 000004 LINE0=4 ; ENABLE LINE TYPE 0
  
```


212	000005	LINE1=5	:ENABLE LINE TYPE 1
213	000006	LINE2=6	:ENABLE LINE TYPE 2
214	000007	LINE3=7	:ENABLE LINE TYPE 3
215			
216	002000	PATH0=2000	:DIRECTION 0
217	006000	PATH1=6000	:DIRECTION 1
218	012000	PATH2=12000	:DIRECTION 2
219	016000	PATH3=16000	:DIRECTION 3
220	022000	PATH4=22000	: 4
221	026000	PATH5=26000	: 5
222	032000	PATH6=32000	: 6
223	036000	PATH7=36000	: 7
224			
225	160000	DJMP=160000	:DISPLAY ABSOLUTE JUMP
226	161000	DJMPR=DJMP!BIT9	:DISPLAY RELATIVE JUMP
227	162000	DJSR=DJMP!BIT10	:DISPLAY JSR ABSOLUTE
228	163000	DJSRR=DJSR!BIT9	:DISPLAY JSR RELATIVE
229			
230	164000	DNOP=164000	
231	166000	DPOP=DNOP!BIT10	:POP AND RESTORE
232	165000	DPOPNR=DNOP!BIT9	:POP AND NO RESTORE
233	164000	CONSL0=DNOP	:CONSOLE 0
234	164400	CONSL1=DNOP!BIT8	:CONSOLE 1
235			
236	170000	STATSA=170000	
237	173400	DSTOP=173400	
238	170002	DMENU0=STATSA!BIT1	:DISABLE MENU
239	170003	DMENU1=DMENU0!BIT0	
240			
241	000200	LPLITE=200	
242	000300	LPDARK=300	
243	000040	ITAL0=40	:DISABLE ITALIC CHARACTERS
244	000060	ITAL1=60	
245	000004	SYNC30=4	:ENABLE SYNC OF 30 CPS
246	000010	SYNC40=10	:ENABLE SYNC OF 40 CPS
247			
248	174000	STATSB=174000	
249			
250	000100	INCR=100	:ENABLE "GRAPHPLOT INCREMENT REG. CHANGE"
251			
252	154000	STATSC=154000	
253	155000	CHRRTO=STATSC!BIT9	:DISABLE CHAR ROTATE
254	155400	CHRRT1=CHRRTO!BIT8	
255			
256	154200	CHARS0=STATSC!BIT7	:LOAD CHARACTER SCALE TO 1/2
257	154240	CHARS1=CHARS0!BIT5	: 1
258	154300	CHARS2=CHARS0!BIT6	: 1 1/2
259	154340	CHARS3=CHARS0!BIT6!BIT5	: 2
260			
261	154020	VCTRO0=STATSC!BIT4	:LOAD VECTOR SCALE REGISTER
262			
263	176000	STATE=STATSB!BIT10	
264			
265	176002	STRNG0=STATE!BIT1	:DISABLE CHARACTER STRING TERMINATE
266	176003	STRNG1=STRNG0!BIT0	
267			


```

268          176040          EDGE0=STATE!BIT5          ;DISABLE EDGE INTERRUPT
269          176060          EDGE1=EDGE0!BIT4
270          150000          DNAME=150000          ;LOAD DISPLAY NAME REGISTER
271
272          ;MORE EQUATES
273
274          040000          INTX=BIT14          ;INTENSIFY
275          000177          MAXMUX=177          ;MAX. MENU X WIDTH
276          001777          MAXX=1777          ;MAX. X AXIS LENGTH
277          001777          MAXY=1777          ;MAX. Y AXIS LENGTH
278          000777          HALFX=MAXX/2          ;HALF OF MAXIUM LENGTH
279          020000          MINUSX=20000          ;NEGATIVE SIGN BIT
280          020000          MINUSY=20000          ;NEGATIVE SIGN BIT
281          000100          MINSUY=100          ;NEGATIVE SIGN BIT <SHORT VECTOR MODE>
282          000021          SUPON=21          ;SUPER-SCRIPT ENABLE
283          000023          SUPOFF=23          ;SUPER-SCRIPT DISABLE
284          000022          SUBON=22          ;SUB-SCRIPT ENABLE
285          000024          SUBOFF=24          ;SUB-SCRIPT DISABLE
286
287
291          .SBTTL          .SBTTL          TEST          LETTER          DESCRIPTION
292          .SBTTL          TEST          LETTER          DESCRIPTION
293          .SBTTL          ----          -----          -----
294          .SBTTL
295
296          ::*****
297          (3)          ;*TEST 1          A          DIRECTORY FRAME
298          (3)          ::*****
299          (2) 002324 000004          TST1: SCOPE
300          002326 004537 014506          JSR R5,DSPLA          ;EXIT TO DISPLAY A FRAME
301          002332 001000          1000
302          002334 015020          FRMEO          ;USING THE DIR. FRAME
303
304          ::*****
305          (3)          ;*TEST 2          B          ASTIGMATISM AND SETTLING TIME
306          (3)          ::*****
307          (2) 002336 000004          TST2: SCOPE
308          002340 004537 014506          JSR R5,DSPLA          ;DISPLAY DATA LOCATED IN 'BUFFER'
309          002344 020000          20000
310          002346 017254          TABB
    
```



```
310 .....  
(3) : *TEST 3 C SHORT TERM DRIFT  
(3) : .....  
(2) 002350 000004 TST3: SCOPE  
311 002352 012737 000100 014502 MOV #BIT6,TEMPA ;LCAD EXECUTION COUNT  
312 002360 012700 014716 1$: MOV #TABAR,RO ;LOAD TABLE POINTER  
313  
314 002364 012037 024746 2$: MOV (RC)+,STDRA ;LOAD X POSITION DATA  
315 002370 012037 024750 MOV (RO)+,STDRB ;LOAD Y POSITION DATA  
316  
317 002374 100441 BMI 3$  
318 002376 004537 014506 JSR R5,DSPLA ;LOAD X+Y POSITION-DO NOT DISPLAY  
319 002402 000001 1  
320 002404 024742 STDPIC  
321  
322 002406 052737 040000 024746 BIS #INTX,STDRA ;LOAD INTENSIFY ENABLE  
323  
324 002414 004537 014506 JSR R5,DSPLA ;DISPLAY DATA  
325 002420 000001 1  
326 002422 024742 STDPIC  
327 002424 004537 002510 JSR R5,SECDLY ;DELAY FOR 5 MSEC  
328 002430 000005 5  
329 002432 004537 014506 JSR R5,DSPLA ;DISPLAY POINT AGAIN  
330 002436 000001 1  
331 002440 024742 STDPIC  
332 002442 004537 002510 JSR R5,SECDLY ;DELAY FOR 5 MSEC  
333 002446 000005 5  
334  
335 002450 004537 014506 JSR R5,DSPLA ;DISPLAY POINT AGAIN  
336 002454 000001 1  
337 002456 024742 STDPIC  
338 002460 004537 002510 JSR R5,SECDLY ;DELAY FOR 5 MSEC  
339 002464 000005 5  
340  
341 002466 004537 014506 JSR R5,DSPLA ;DISPLAY POINT AGAIN  
342 002472 000001 1  
343 002474 024742 STDPIC  
344 002476 000732 BR 2$  
345  
346 002500 005337 014502 3$: DEC TEMPA ;FINISHED EXECUTION?  
347 002504 001325 BNE 1$  
348 002506 000416 BR TST4 ;:BR OVER SUBROUTINE  
349  
350 002510 012537 002542 SECDLY: MOV (R5)+,11$ ;LOAD TOTAL DELAY COUNT  
351 002514 013737 001256 002540 2$: MOV DELAY,10$ ;LOAD 1 MS.  
352 002522 005337 002540 1$: DEC 10$ ;DELAY  
353 002526 001375 BNE 1$  
354 002530 005337 002542 DEC 11$ ;DEC MSEC COUNT  
355 002534 100367 BPL 2$  
356 002536 000205 RTS R5 ;EXIT  
357 002540 000000 10$: 0  
358 002542 000000 11$: 0  
359
```



```

361
(3)
(3)
(2) 002544 000004
362 002546 012737 000010 014502
363 002554 004537 014506
364 002560 001000
365 002562 017516
366
367 002564 005337 014502
368 002570 001371
369
370
(3)
(3)
(2) 002572 000004
371 002574 012737 001000 014502
372 002602 004537 014506
373 002606 000010
374 002610 017516
375 002612 004537 014506
376 002616 000010
377 002620 025404
378
379 002622 005337 014502
380 002626 001365
381
382
(3)
(3)
(2) 002630 000004
383
384 002632 012700 036656
385 002636 012720 164700
386 002642 012701 017602
387 002646 012120
388 002650 022701 017632
389 002654 001374
390
391 002656 012737 001777 024762
392 002664 012737 000040 024764
393 002672 012737 060040 024770
394
395 002700 012737 000037 014502
396 002706 012701 024760
397 002712 012120
398 002714 022701 024774
399 002720 001374
400 002722 005337 014502
401 002726 001407
402 002730 062737 000040 024764
403 002736 062737 000040 024770
404 002744 000760
405 002746 012737 000000 024762
406 002754 012737 000040 024764
407 002762 012737 041740 024770

*****
*TEST 4 D MINOR AXIS GAIN, OFFSET AND PHASE ADJUSTMENT
*****
TST4: SCOPE
MOV #BIT3,TEMPA ;LOAD EXECUTION COUNT
1$: JSR R5,DSPLA ;DISPLAY SUB-PICTURE
1000
FRME2
DEC TEMPA ;FINISHED EXECUTION ?
BNE 1$ ;BR IF NOT

*****
*TEST 5 E MAJOR AXIS OFFSET AND VECTOR START POINT ADJUSTMENT
*****
TST5: SCOPE
MOV #BIT9,TEMPA
1$: JSR R5,DSPLA ;DISPLAY OFFSET SUB-PICTURE
10
FRME2
JSR R5,DSPLA ;DISPLAY VECTOR START SUB-PICTURE
10
VSTRT
DEC TEMPA ;FINISHED EXECUTION LOOP?
BNE 1$ ;BR IF NOT DONE.

*****
*TEST 6 F VECTOR LENGTH GAIN, CONVERGENCE AND VECTOR LINEARITY
*****
TST6: SCOPE
;GENERATE THE SCREEN PICTURE BUFFER FIRST
MOV #BUFFER,R0 ;LOAD DISPLAY PICTURE POINTER
MOV #CONSL1!BIT7!BIT6,(R0)+ ;ENABLE CONSOLE #1
MOV #PICST0,R1 ;LOAD 'BOX POINTER'
1$: MOV (R1)+,(R0)+ ;GET DATA INTO BUFFER
CMP #PICST1,R1 ;TEST FOR END
BNE 1$ ;BR IF NOT
MOV #MAXX,PICVTA ;LOAD STARTING X POSITION
MOV #40,PICVTB ;LOAD STARTING Y POSITION
MOV #INTX!MINUSX+40,PICVTC ;LOAD INTENFIFY, MINUS DIR AND VALUE
MOV #37,TEMPA ;LOAD A COUNTER
3$: MOV #PICVTL,R1 ;LOAD SUB-PICTURE POINTER
2$: MOV (R1)+,(R0)+ ;LOAD DATA
CMP #PICVTE,R1 ;TEST FOR END
BNE 2$ ;BR IF NOT
DEC TEMPA
BEQ 4$ ;BR IF DONE THIS SIDE
ADD #40,PICVTB ;ADJUST STARTING Y POSITION
ADD #40,PICVTC ;ADJUST VECTOR LENGTH
BR 3$
4$: MOV #0,PICVTA ;LOAD STARTING X POSITION
MOV #40,PICVTB ;LOAD STARTING Y POSITION
MOV #INTX+1740,PICVTC ;LOAD INTENSIFY AND DELTA VALUE

```



```

458 (3)
(3)
(2) 003232 000004
459 003234 012700 036656
460 003240 012720 164700
461 003244 004737 003414
462 003250 012701 000020
463 003254 012720 040000
464 003260 012720 001777
465 003264 012720 000100
466 003270 012720 021777
467 003274 005301
468 003276 001366
469 003300 012720 020001
470 003304 012720 000000
471 003310 012720 040000
472 003314 012720 001777
473 003320 004737 003414
474 003324 012701 000020
475 003330 012720 041777
476 003334 012720 000000
477 003340 012720 021777
478 003344 012720 000100
479 003350 005301
480 003352 001366
481 003354 012720 000000
482 003360 012720 020001
483 003364 012720 041777
484 003370 012720 000000
485 003374 012720 173400
486 003400 012720 160000
487 003404 012710 036656
488 003410 000137 003436
489
490 003414 012720 117000
491 003420 012720 000000
492 003424 012720 000000
493 003430 012720 110000
494 003434 000207
495
496 003436 004537 014506
497 003442 004000
498 003444 036656
499

*****
: *TEST 7 G PINCUSHION FRAME
*****
TST7: SCOPE
MOV #BUFFER,R0 ;LOAD START ADDRESS
MOV #CONSL1!BIT7!BIT6,(R0)+ ;ENABLE CONSOL #1
JSR PC,3$ ;LOAD 0,0 ORGIN
MOV #20,R1 ;SETUP COUNT
1$: MOV #INTX,(R0)+ ;LOAD INT LINE
MOV #MAXY,(R0)+ ; MAX Y
MOV #100,(R0)+ ;LOAD DELTA X
MOV #MINUSX+MAXY,(R0)+ ;LOAD - MAX Y
DEC R1 ;FINISHED ?
BNE 1$ ;BR IF NOT
MOV #MINUSX+1,(R0)+ ;GO BACK 1 UNIT
MOV #0,(R0)+
MOV #INTX,(R0)+
MOV #MAXY,(R0)+ ;PLOT LAST LINE
JSR PC,3$ ;SET ORGIN
MOV #20,R1 ;SETUP COUNT
2$: MOV #INTX+MAXX,(R0)+ ;LOAD DELTA X MAX
MOV #0,(R0)+ ;LOAD DELTA Y = 0
MOV #MINUSX+MAXX,(R0)+ ;RETRACE
MOV #100,(R0)+ ;LOAD DELTA Y OF 100
DEC R1 ;FINISHED ?
BNE 2$ ;BR IF NOT
MOV #0,(R0)+
MOV #MINUSX+1,(R0)+
MOV #INTX+MAXX,(R0)+ ;PLOT LAST LINE
MOV #0,(R0)+
MOV #DSTOP,(R0)+ ;LOAD STOP
MOV #DJMP,(R0)+ ;LOAD JUMP
MOV #BUFFER,(R0)
JMP 4$

3$: MOV #POINT!INT4,(R0)+ ;LOAD POINT
MOV #0,(R0)+ ; AT X
MOV #0,(R0)+ ; AT Y
MOV #LONGV,(R0)+ ;LONG VECTOR
RTS PC ;EXIT

4$: JSR R5,DSPLA ;EXIT TO DISPLAY FRAME
4000
BUFFER ;USING THE CROSS HATCH PATTERN

```



```
501 ::*****  
(3) :*TEST 10 H OCTAGONS AND CIRCLES  
(3) :*****  
(2) 003446 000004 TST10: SCOPE  
502 003450 004537 014506 JSR R5,DSPLA ;DISPLAY TEST  
503 003454 006000 6000 ;FRAME # 3  
504 003456 020546 FRME3  
505  
506 ::*****  
(3) :*TEST 11 I SCISSORING AND VECTOR SCALING  
(3) :*****  
(2) 003460 000004 TST11: SCOPE  
507 003462 012737 000400 014504 MOV #BIT8,TEMPB ;LOAD EXECUTION COUNTER  
508 003470 012737 154037 025344 1$: MOV #VCTR00!17,PICSCA ;RELOAD VECTOR SCALE LENGTH TO 17  
509 003476 012737 000020 014502 MOV #20,TEMPA ;LOAD SCALE COUNTER  
510  
511 003504 004537 014506 2$: JSR R5,DSPLA ;EXIT TO DISPLAY ROUTINE  
512 003510 000001 1 PICSCS ;USING PRESET PICTURE DATA  
513 003512 025002  
514  
515 003514 005337 025344 DEC PICSCA ;REDUCE VECTOR SCALE  
516 003520 005337 014502 DEC TEMPA ;FINISHED ALL SCALES?  
517 003524 001367 BNE 2$ ;BR IF NOT  
518  
519 003526 005337 014504 DEC TEMPB ;FINISHED EXECUTION COUNT  
520 003532 001356 BNE 1$ ;BR IF NOT  
521
```



```

523      ::*****
(3)      :*TEST 12      J      OFFSET X AND OFFSET Y POSITION
(3)      :*****
(2) 003534 000004 TST12: SCOPE
524      :DISPLAY A SQUARE IN THE CENTER SCREEN, THEN
525      :MOVE THE BOX TO THE RIGHT
526 003536 012737 010000 022246      MOV      #BIT12,OFFT1      ;LOAD BASIC X OFFSET VALUE
527 003544 012737 010000 022250      MOV      #BIT12,OFFT2      ;LOAD BASIC Y OFFSET VALUE
528 003552 004537 014506      1$: JSR      R5,DSPLA      ;DISPLAY THAT FRAME
529 003556 000100      100
530 003560 022244      OFFTST
531 003562 005737 014634      TST      SWITCH      ;TEST IF HOLD HERE
532 003566 001371      BNE      1$      ;BR IF YES
533 003570 062737 000001 022246      ADD      #1,OFFT1      ;UPDATE THE X OFFSET
534 003576 022737 011400 022246      CMP      #BIT12!1400,OFFT1 ;TEST IF MORE TO MOVE
535 003604 001362      BNE      1$      ;BR IF NOT
536
537      :MOVE THE BOX TO THE LEFT
538 003606 012737 030000 022246      MOV      #BIT12!MINUSX,OFFT1 ;LOAD THE BASIC X OFFSET
539 003614 012737 010000 022250      MCV      #BIT12,OFFT2      ;LOAD THE BASIC Y OFFSET
540 003622 004537 014506      2$: JSR      R5,DSPLA      ;DISPLAY THE FRAME
541 003626 000100      100
542 003630 022244      OFFTST
543 003632 005737 014634      TST      SWITCH      ;TEST IF HOLD HERE
544 003636 001371      BNE      2$      ;BR IF HOLD
545 003640 062737 000001 022246      ADD      #1,OFFT1      ;UPDATE THE X OFFSET
546 003646 022737 031400 022246      CMP      #BIT12!MINUSX!1400,OFFT1 ;TEST IF MORE
547 003654 001362      BNE      2$      ;BR IF NOT
548
549      :MOVE THE BOX UP
550 003656 012737 010000 022246      MOV      #BIT12,OFFT1      ;LOAD BASIC X OFFSET
551 003664 012737 010000 022250      MOV      #BIT12,OFFT2      ;LOAD BASIC Y OFFSET
552 003672 004537 014506      3$: JSR      R5,DSPLA      ;DISPLAY THAT FRAME
553 003676 000100      100
554 003700 022244      OFFTST
555 003702 005737 014634      TST      SWITCH      ;TEST IF HOLD HERE
556 003706 001371      BNE      3$      ;BR IF YES
557 003710 062737 000001 022250      ADD      #1,OFFT2      ;UPDATE Y OFFSET
558 003716 022737 011400 022250      CMP      #BIT12!1400,OFFT2 ;TEST IF MORE
559 003724 001362      BNE      3$      ;BR IF NOT
560
561      :MOVE THE BOX DOWN
562 003726 012737 030000 022250      MOV      #BIT12!MINUSY,OFFT2 ;LOAD THE BASIC Y OFFSET
563 003734 012737 010000 022246      MOV      #BIT12,OFFT1      ;LOAD THE BASIC X OFFSET
564 003742 004537 014506      4$: JSR      R5,DSPLA      ;DISPLAY THAT FRAME
565 003746 000100      100
566 003750 022244      OFFTST
567 003752 005737 014634      TST      SWITCH      ;TEST IF HOLD HERE
568 003756 001371      BNE      4$      ;BR IF YES
569 003760 062737 000001 022250      ADD      #1,OFFT2      ;UPDATE Y OFFSET
570 003766 022737 031400 022250      CMP      #BIT12!MINUSX!1400,OFFT2 ;TEST IF MORE
571 003774 001362      BNE      4$      ;BR IF NOT

```


573
 574
 575
 576
 577
 578
 579
 580
 581
 582
 583
 584
 585
 586
 587
 588
 589
 590
 591
 592
 593
 594
 595
 596
 597
 598
 599
 600
 601
 602
 603
 604
 605
 606
 607
 608
 609
 610
 611
 612
 613
 614
 615

003776
 003776 005037 022246
 004002 005037 022250
 004006 005077 175266
 004012 005077 175264
 004016 012777 004112 175272
 004024 004537 014506
 004030 000001 022244
 004034 005000
 004036 010077 175236
 004042 010077 175234
 004046 012737 000100 014630
 004054 004537 014540
 004060 005737 014634
 004064 001364
 004066 005200
 004070 020027 021400
 004074 001414
 004076 020027 001400
 004102 001355
 004104 012700 020000
 004110 000752
 004112 010077 175162
 004116 010077 175160
 004122 000137 014556
 004126 005077 175146
 004132 005077 175144
 004136 012777 014556 175152

... NOW DO IT ALL AGAIN SETTING THE OFFSET REGISTERS DIRECTLY.
 ... MOVE THE BOX DIAGONALLY TO THE UPPER RIGHT (POS, POS)
 ... THEN TO THE LOWER LEFT (NEG, NEG).
 ... *** NOTE: WHEN DEALING WITH THE OFFSET REGISTERS DIRECTLY,
 ... IT APPEARS THAT THE NEG DIRECTION (BIT13) GETS CLEARED ON A
 ... DISPLAY "START" FUNCTION. RESUME SEEMS TO BE OK, HOWEVER TO
 ... BE SAFE, "START" WITH + OFFSET, AND RELOAD REGISTERS PRIOR
 ... TO EVERY "RESUME". ??? H - A - C - K ???

```
TST12A:
      CLR      OFFT1      ; ZERO OFFSET INSTRUCTIONS...
      CLR      OFFT2      ; ...IN THE DISPLAY FILE.
      CLR      @XDOFF     ; CLEAR OFFSET REGISTERS.
      CLR      @YDOFF
      MOV      #2$,@DDONE ; CHANGE STOP VECTOR.
      JSR      R5,DSPLA   ; XCT DISPLAY...
      1, OFFTST          ; ...ONCE TO INIT THE OFFSETS.
      CLR      R0         ; VARIABLE OFFSET VALUE => R0.
1$:    MOV      R0,@XDOFF ; SET 1ST/NEXT OFFSET.
      MOV      R0,@YDOFF
      MOV      #100,COUNT ; SET CYCLE COUNT...
      JSR      R5,RESUME  ; ...AND RESUME DISPLAY.
      TST      SWITCH
      BNE     1$         ; BR IF STOP MOTION IS SET.
      INC      R0         ; BUMP OFFSET VALUE.
      CMP     R0,#BIT13!1400 ; ALL DONE ???
      BEQ     3$         ; EXIT IF SO.
      CMP     R0,#1400   ; HALF DONE ???
      BNE     1$         ; NOT YET, LOOP.
      MOV     #BIT13,R0  ; YES, START NEG HALF.
      BR      1$
      2$:    MOV      R0,@XDOFF ; *** ON STOP INTERRUPT...
      MOV      R0,@YDOFF     ; ...RELOAD OFFSETS ***
      JMP     STOPI          ; CONTINUE NORMAL STOP SEQ.
      3$:    CLR      @XDOFF   ; CLEAR OFFSET REGISTERS.
      CLR      @YDOFF
      MOV     #STOPI,@DDONE ; RESET STOP VECTOR.
```



```

617      ;*****
(3)      ;*TEST 13      K      CHARACTER SCALE FRAME
(3)      ;*****
(2) 004144 000004
618 004146 012737 000200 014502 TST13: SCOPE
619      MOV      #BIT7,TEMPA      ;LOAD EXECUTION COUNTER
620 004154 004537 014506 1$: JSR      R5,DSPLA      ;DISPLAY SUBPICTURE DATA
621 004160 000060
622 004162 025712      CHAQU
623      DEC      TEMPA      ;FINISHED EXECUTION?
624 004164 005337 014502      BNE      1$      ;BR IF NOT
625 004170 001371
626
627      ;*****
(3)      ;*TEST 14      L      CHARACTER QUALITY AND CHARACTER ROTATE IN THE MENU
(3)      ;*****
(2) 004172 000004
628 004174 012700 036656 TST14: SCOPE
629 004200 012720 164700      MOV      #BUFFER,R0      ;LOAD DESTINATION POINTER
630 004204 012720 114000      MOV      #CONSL1!BIT7!BIT6,(R0)+ ;ENABLE CONSOLE #1
631 004210 005020      MOV      #POINT,(R0)+      ;LOAD "DPOINT"
632 004212 012720 001763      CLR      (R0)+      ;LOAD X AXIS
633 004216 012701 000015      MOV      #MAXY-14,(R0)+      ;LOAD Y
634 004222 012720 162000 1$: MOV      #15,R1      ;LOAD COUNT
635 004226 012720 026450      MOV      #DJSR,(R0)+      ;LOAD "DJSR" TO BUFFER SPACE
636 004232 005301      DEC      R1      ;LOAD TARGET ADDRESS
637 004234 001372      BNE      1$      ;FINISHED ?
638 004236 012720 160000      MOV      #DJMP,(R0)+      ;BR UNTIL DONE
639 004242 012720 026402      MOV      #ROTCHR,(R0)+      ;RETURN ADDRESS TO THE ROTATED CHAR SUB-PIC
640      JSR      R5,DSPLA      ;EXECUTE DISPLAY FILE
641 004246 004537 014506      200
642 004252 000200      BUFFER      ;STARTING AT 'BUFFER' ADDRESS
643 004254 036656
644
    
```



```

646 (3)
647 (2) 004256 000004
648 004260 012700 036656
649 004264 012720 155000
650 004270 012720 164700
651 004274 012720 114000
652 004300 005020
653 004302 012720 001700
654 004306 012720 100000
655 004312 012737 000100 004512
656 004320 004737 004450
657 004324 012737 000140 004512
658 004332 004737 004450
659 004336 012737 000040 004512
660 004344 004737 004450
661 004350 012720 170040
662 004354 004737 004414
663 004360 004737 004554
664 004364 012720 170060
665 004370 004737 004414
666 004374 012720 173400
667 004400 012720 160000
668 004404 012720 036656
669 004410 000137 004572
670
671 004414 112720 000016
672 004420 005002
673 004422 110220
674 004424 005202
675 004426 022702 000017
676 004432 001774
677 004434 022702 000040
678 004440 001370
679 004442 012720 020017
680 004446 000207
681
682 004450 012720 170040
683 004454 013702 004512
684 004460 004737 004536
685 004464 004737 004554
686 004470 012720 170060
687 004474 013702 004512
688 004500 004737 004536
689 004504 004737 004514
690 004510 000207
691
692 004512 000000
693
694
695
696 004514 112720 000015
697 004520 112720 000012
698 004524 112720 000012

*****
*TEST 15 M CHARACTER SET, SUPERSCRIPT, SUBSCRIPT AND ITALICS
*****
TST15: SCOPE
;SET UP THE BUFFER FOR THIS TEST
MOV #BUFFER,R0
MOV #CHRRTO,(R0)+ ;DISABLE CHAR. ROTATE
MOV #CONSL1!BIT7!BIT6,(R0)+ ;ENABLE CONSOLE #1
MOV #POINT,(R0)+ ;LOAD POINT MPDE
CLR (R0)+
MOV #MAXY-77,(R0)+
MOV #CHAR,(R0)+
MOV #100,STCHAR ;LOAD INITIAL CHAR.
JSR PC,LOADBF
MOV #140,STCHAR ;LOAD INITIAL LC CHAR
JSR PC,LOADBF ;LOAD LINE
MOV #40,STCHAR ;LOAD NUMBERS AND PUNCT
JSR PC,LOADBF ;LOAD LINE
MOV #STATSA!ITALO,(R0)+ ;LOAD NORMAL FONT
JSR PC,LOADSP ;LOAD SPECIAL CHARS
JSR PC,SPACE ;INSERT SPACES
MOV #STATSA!ITAL1,(R0)+ ;LOAD ITALICS FONT
JSR PC,LOADSP ;LOAD SPIECAL
MOV #DSTOP,(R0)+ ;LOAD DSTOP
MOV #DJMP,(R0)+
MOV #BUFFER,(R0)+
JMP FILE4A

LOADSP: MOV #16,(R0)+ ;LOAD "SHIFT-OUT" CHARACTER
CLR R2 ;SET INITIAL SHIFT OUT CHAR
1$: MOV R2,(R0)+ ;LOAD CHAR
2$: INC R2
CMP #17,R2 ;TEST FOR "SHIFT-IN" (SI)
BEQ 2$ ;BR IF SI "17"
CMP #40,R2 ;FINISHED ?
BNE 1$ ;BR IF NOT
MOV #20017,(R0)+ ;LOAD SHIFT-IN SPACE
RTS PC ;EXIT

LOADBF: MOV #STATSA!ITALO,(R0)+ ;LOAD NORMAL FONT
MOV STCHAR,R2 ;GET STARTING CHAR
JSR PC,FILLIT ;LOAD THE CHARACTERS
JSR PC,SPACE ;INSERT SPACES
MOV #STATSA!ITAL1,(R0)+ ;LOAD ITALICS FONT
MOV STCHAR,R2 ;GET STARTING CHARACTER
JSR PC,FILLIT ;LOAD THE CHARACTERS
JSR PC,ACRLF ;INSERT CR-LF
RTS PC ;EXIT

STCHAR: 0
;LOAD CR-LF'S TO VERTICALLY SPACE THE STRINGS

ACRLF: MOV #15,(R0)+
MOV #12,(R0)+
MOV #12,(R0)+
  
```



```

699 004530 112720 000012          MOVB  #12,(R0)+
700 004534 000207                  RTS    PC                    ;EXIT
701
702                                ;FILL IN WITH AN INCREMENTING CHARACTERS
703
704 004536 012703 000040          FILLIT: MOV  #40,R3
705 004542 110220                  1$:  MOVB  R2,(R0)+
706 004544 005202                  INC    R2
707 004546 005303                  DEC    R3
708 004550 001374                  BNE   1$
709 004552 000207                  RTS    PC
710
711                                ;LOAD "SPACE" CHAR TO SEPERATE CHAR STRINGS
712
713 004554 012703 000010          SPACE: MOV  #10,R3
714 004560 112720 000040          1$:  MOVB  #40,(R0)+        ;LOAD A SPACE
715 004564 005303                  DEC    R3
716 004566 001374                  BNE   1$                    ;BR IF NOT DONE
717 004570 000207                  RTS    PC                    ;EXIT
718
719                                ;NOW ACTUAL DISPLAY THE CHARACTER SET FRAME NOT ROTATED
720
721 004572 012737 000600 014502    FILE4A: MOV  #600,TEMPA      ;LOAD A COUNTER
722 004600 012737 155000 036656    MOV  #CHRRTO,BUFFER      ;DISABLE ROTATE
723 004606 012737 001700 036666    1$:  MOV  #MAXY-77,BUFFER+10 ;LOAD STARTING Y POINT
724 004614 004537 014506          JSR  R5,DSPLA             ;DISPLAY IN UPPER HALF OF SCREEN
725 004620 000001                  1
726 004622 036656          BUFFER
727
728 004624 012737 000400 036666    MOV  #400,BUFFER+10      ;LOAD STARTING Y POINT IN THE LOWER HALF
729 004632 004537 014506          JSR  R5,DSPLA             ;DISPLAY IN LOWER HALF OF SCREEN
730 004636 000001                  1
731 004640 036656          BUFFER
732
733 004642 004537 014506          JSR  R5,DSPLA             ;DISPLAY SUPER AND SUBSCRIPT IN THE MIDDLE
734 004646 000001                  1
735 004650 022302          SUPPIC
736
737 004652 005337 014502          DEC  TEMPA                ;FINISHED ?
738 004656 001353                  BNE   1$                    ;BR IF NOT
739 004660 005737 014634          TST  SWITCH                ;TEST IF "FREEZE"
740 004664 001342                  BNE   FILE4A               ;BR IF YES
741
742                                ;NOW DISPLAY THE CHARACTER SET FRAME ROTATED
743
744 004666 012737 000600 014502    2$:  MOV  #600,TEMPA      ;LOAD DELAY COUNTER FOR THIS HALF
745 004674 012737 155400 036656    MOV  #CHRRT1,BUFFER      ;ENABLE CHAR. ROTATE
746 004702 005037 036666          CLR  BUFFER+10            ;RESET Y ORGIN
747 004706 012737 000100 036664    3$:  MOV  #100,BUFFER+6     ;LOAD X ORGIN
748 004714 004537 014506          JSR  R5,DSPLA             ;DISPLAY FRAME
749 004720 000001                  1
750 004722 036656          BUFFER
751
752 004724 012737 001400 036664    MOV  #1400,BUFFER+6      ;REPOSITION THE X ORGIN
753 004732 004537 014506          JSR  R5,DSPLA             ;DISPLAY FRAME AT NEW ORGIN
754 004736 000001                  1
  
```


CZVSFA VS60 VISUAL WITH TABLET XY CORRELATOR
CZVSFA.P11 26-MAR-80 11:46 T15

MACY11 30G(1063) ^{D 4} 26-MAR-80 11:46 PAGE 10-2
M CHARACTER SET, SUPERSCRIPT, SUBSCRIPT AND ITALICS

SEQ 0042

755 004740 036656
756
757 004742 004537 014506
758 004746 000001
759 004750 022330
760
761 004752 005337 014502
762 004756 001353
763 004760 005737 014634
764 004764 001340
765

BUFFER

JSR R5,DSPLA
1
SUPCO

;DISPLAY TEXT

DEC TEMPA
BNE 3\$
TST SWITCH
BNE 2\$

;FINISHED DELAY ?
;BR IF NOT
;TEST IF FREEZE
;BR IF YES


```
767 :*****  
(3) :*TEST 16 N SYNC SPEED AND CHARACTER TERMINATE TEST  
(3) :*****  
(2) 004766 000004 TST16: SCOPE  
768 047516 ;ASCII VALUE FOR 'NO'  
769 030064 : " " "40"  
770 030063 : " " "30"  
771  
772 004770 012777 000377 174316 MOV #377,@VSTERM ;LOAD TERMINATE REG.  
773 004776 012737 047516 022530 MOV #NO,SYNSPD ;LOAD SYNC ASCII VALUE  
774 005004 012737 170000 022436 MOV #STATSA,SYNPIC ;LOAD NO SYNC ENABLE  
775 005012 004537 014506 1$: JSR R5,DSPLA ;DISPLAY THAT FRAME WITH 'NO' SYNC  
776 005016 010000 10000  
777 005020 022436 SYNPIC  
778 005022 005737 014634 TST SWITCH ;TEST IF HOLD SET  
779 005026 001371 BNE 1$ ;BR IF HOLD  
780  
781 005030 012777 000377 174256 MOV #377,@VSTERM ;LOAD TERMINATE REG.  
782 005036 012737 030064 022530 MOV #S40,SYNSPD ;LOAD SYNC ASCII VALUE  
783 005044 012737 170010 022436 MOV #STATSA!SYNC40,SYNPIC ;LOAD SYNC ENABLE TO 40  
784 005052 004537 014506 2$: JSR R5,DSPLA ;DISPLAY THAT FRAME WITH '40' SYNC  
785 005056 000200 200  
786 005060 022436 SYNPIC  
787 005062 005737 014634 TST SWITCH ;TEST IF HOLD SET  
788 005066 001371 BNE 2$ ;BR IF HOLD  
789  
790 005070 012777 000377 174216 MOV #377,@VSTERM ;LOAD TERMINATE REG.  
791 005076 012737 030063 022530 MOV #S30,SYNSPD ;LOAD ASCII VALUE OF 30  
792 005104 012737 170004 022436 MOV #STATSA!SYNC30,SYNPIC ;LOAD 30 CPS ENABLE  
793 005112 004537 014506 3$: JSR R5,DSPLA ;DISPLAY THAT FRAME AT '30' SYNC  
794 005116 000200 200  
795 005120 022436 SYNPIC  
796 005122 005737 014634 TST SWITCH ;TEST IF HOLD  
797 005126 001371 BNE 3$ ;BR IF HOLD  
798
```



```

800      ::*****
(3)      :*TEST 17      0      DASH LINES AND BLINK
(3)      :*****
(2) 005130 000004
801 005132 004537 014506
802 005136 020000
803 005140 022660
804
805      :*****
(3)      :*TEST 20      P      VECTOR SPRAY (LENGTH) TEST
(3)      :*****
(2) 005142 000004
806 005144 012700 036656
807 005150 012737 041776 014712
808 005156 012737 000001 014714
809 005164 004737 005270
810 005170 062737 000002 014714
811 005176 023727 014714 001777
812 005204 003767
813 005206 012737 001777 014714
814 005214 000407
815 005216 162737 000002 014712
816 005224 023727 014712 040000
817 005232 002403
818 005234 004737 005270
819 005240 000766
820 005242 012720 173400
821 005246 012720 160000
822 005252 012720 023106
823 005256 004537 014506
824 005262 000200
825 005264 023106
826 005266 000407
827 005270 013720 014712
828 005274 013720 014714
829 005300 005020
830 005302 005020
831 005304 000207

TST17: SCOPE
      JSR      R5,DSPLA      ;EXIT TO DISPLAY A FRAME
      20000
      FRME5      ;USING THE DASH AND BLINK FRAME

TST20: SCOPE
      MOV      #BUFFER,R0      ;LOAD BUFFER POINTER
      MOV      #INTX!MAXX-1,DELTX6 ;LOAD X PRESET VALUE
      MOV      #1,DELT6      ;LOAD Y PRESET VALUE
1$: JSR      PC,SPRAY      ;LOAD INCREASING ANGLE VECTOR
      ADD      #2,DELT6      ;UPDATE Y LENGTH
      CMP      DELTY6,#MAXY ;TEST IF END
      BLE      1$          ;BR IF NOT
      MOV      #MAXY,DELT6 ;RESET MAX Y LENGTH
      BR      4$
3$: SUB      #2,DELTX6      ;REDUCE X LENGHT
      CMP      DELTX6,#INTX ;TEST IF END
      BLT      2$          ;BR IF DONE
4$: JSR      PC,SPRAY      ;LOAD DECREASING ANGLE VECTOR
      BR      3$
2$: MOV      #DSTOP,(R0)+ ;LOAD STOP
      MOV      #DJMP,(R0)+
      MOV      #FRME6,(R0)+ ;RESTART DISPLAY FRAME
      JSR      R5,DSPLA ;DISPALY PICTURE
      200 ;COUNT
      FRME6
      BR      TST21 ;:BR TO NEXT TEST
SPRAY: MOV      DELTX6,(R0)+ ;LOAD X VECTOR LENGTH
      MOV      DELTY6,(R0)+ ;LOAD Y VECTOR LENGTH
      CLR      (R0)+ ;VECTOR BACK TO ORGIN
      CLR      (R0)+
      RTS      PC ;EXIT
    
```



```
833      ;*****
(3)      ;*TEST 21      Q      HORIZONTAL PHOSPHOR TEST
(3)      ;*****
(2) 005306 000004      TST21: SCOPE
834 005310 005037 023162      CLR      DELTX7
835 005314 012737 000004 014502 1$: MOV      #4,TEMPA      ;LOAD DELAY COUNT
836 005322 004537 014506      2$: JSR      R5,DSPLA      ;EXIT TO DISPLAY A FRAME
837 005326 000004      4
838 005330 023160      FRME10      ;USING THE HORIZ FRAME
839 005332 004537 014506      JSR      R5,DSPLA      ;EXIT TO DISPLAY A FRAME
840 005336 000001      1
841 005340 024174      FRM10      ;USING THE PERIMETER BOX
842 005342 005737 014634      TST      SWITCH      ;TEST THE 'MOTION-SWITCH'
843 005346 001362      BNE      1$      ;BR IF FREEZE THE MOVEMENT
844 005350 005337 014502      DEC      TEMPA      ;DELAY DONE ?
845 005354 100362      BPL      2$      ;BR IF NOT
846 005356 005237 023162      INC      DELTX7      ;UPDATE THE X ORIGIN
847 005362 022737 001777 023162      CMP      #1777,DELTX7      ;TEST IF THE END
848 005370 001351      BNE      1$      ;BR IF NOT
849      ;*****
(3)      ;*TEST 22      R      VERTICAL PHOSPHOR TEST
(3)      ;*****
(2) 005372 000004      TST22: SCOPE
850 005374 005037 023722      CLR      DELT11
851 005400 005037 023442      CLR      DELTY7
852 005404 012737 000004 014502 1$: MOV      #4,TEMPA      ;LOAD DELAY COUNT
853 005412 004537 014506      2$: JSR      R5,DSPLA      ;EXIT TO DISPLAY A FRAME
854 005416 000004      4
855 005420 023436      FRME11      ;USING THE VERT FRAME
856 005422 004537 014506      JSR      R5,DSPLA      ;EXIT TO DISPLAY A FRAME
857 005426 000001      1
858 005430 024174      FRM10      ;USING THE PERIMETER BOX
859 005432 004537 014506      JSR      R5,DSPLA      ;DISPLAY THE MENU BOX
860 005436 000001      1
861 005440 024234      FRM11M      ;DISPLAY THE TEST IN THE MENU
862 005442 004537 014506      JSR      R5,DSPLA
863 005446 000004      4
864 005450 023716      FRM11S      ;DISPLAY THE 'MENU' PHOSPHOR PIC.
865 005452 005737 014634      TST      SWITCH      ;TEST THE 'MOTION-SWITCH'
866 005456 001352      BNE      1$      ;BR IF FREEZE THE MOVEMENT
867 005460 005337 014502      DEC      TEMPA      ;DELAY DONE ?
868 005464 100352      BPL      2$      ;BR IF NOT
869 005466 022737 001277 023722      CMP      #1277,DELT11      ;TEST IF AT TOP OF MENU
870 005474 001402      BEQ      3$      ;BR IF YES, DONT ADVANCE THE MENU
871 005476 005237 023722      INC      DELT11      ;UPDATE THE Y MENU ORGIN
872 005502 005237 023442      3$: INC      DELTY7      ;UPDATE THE Y ORIGIN
873 005506 022737 001777 023442      CMP      #1777,DELTY7      ;TEST IF THE END
874 005514 001333      BNE      1$      ;BR IF NOT
```



```

876      ::*****
(3)      :*TEST 23      S      SHORT VECTOR AND RELATIVE POINT
(3)      :*****
(2)      TST23:  SCOPE
877      005516  000004      MOV      #BUFFER,RO      ;SET UP RO
878      005520  012700  036656  MOV      #POINT,(RO)+    ;SET UP INITIAL
879      005524  012720  114000  MOV      #240,(RO)+      ;X POSITION
880      005530  012720  000240  MOV      #MAXY+1/2,(RO)+ ;Y POSITION
881      005534  012720  001000  MOV      #SHORTV,(RO)+   ;LOAD "SHORT VECTOR"
882      005540  012720  104000  JSR      PC,LOADVT        ;LOAD THE DISPLAY PATTERN
883      005544  004737  005576  MOV      #RELATP,(RO)+   ;LOAD "RELATIVE POINT"
884      005550  012720  130000  JSR      PC,LOADVT        ;LOAD THE DISPLAY PATTERN
885      005554  004737  005576  MOV      #DSTOP,(RO)+    ;LOAD "DISPLAY STOP"
886      005560  012720  173400  MOV      #DJMP,(RO)+     ;LOAD "DISPLAY JUMP"
887      005564  012720  160000  MOV      #BUFFER,(RO)+   ;TO THE BUFFER ADDRESS
888      005570  012720  036656  BR       FIL14A          ;BR TO THE FRAME
889      005574  000413
890      005576  012737  000024  014500  LOADVT: MOV      #24,CNTR      ;LOAD A COUNTER
891      005604  012720  040077  1$:    MOV      #INTX+77,(RO)+ ;LOAD A DELTA Y
892      005610  012720  004177  MOV      #4177,(RO)+     ;LOAD A DELTA X,Y
893      005614  005337  014500  DEC      CNTR            ;FINISHED?
894      005620  001371  BNE     1$              ;BR IF NOT
895      005622  000207  RTS     PC              ;EXIT
896
897      :DISPLAY FOUR SHORT VECTOR/RELATIVE POINT OCTAGONS IN DIFFERENT QUADRANTS
898      005624  012737  006000  014502  FIL14A: MOV      #6000,TEMPA ;LOAD COUNTER
899      005632  012737  000200  024300  1$:    MOV      #200,FRM14A    ;LOAD FIRST OCTAGON
900      005640  012737  000200  024302  MOV      #200,FRM14B
901      005646  004537  014506  JSR      R5,DSPLA        ;DISPLAY OCT.
902      005652  000001  1
903      005654  024274  FRME14
904      005656  012737  001400  024300  MOV      #1400,FRM14A    ;LOAD SECOND OCTAGON
905      005664  012737  000200  024302  MOV      #200,FRM14B
906      005672  004537  014506  JSR      R5,DSPLA        ;DISPLAY 2ND OCT.
907      005676  000001  1
908      005700  024274  FRME14
909      005702  012737  001400  024300  MOV      #1400,FRM14A    ;LOAD THIRD OCTAGON
910      005710  012737  001400  024302  MOV      #MAXY-377,FRM14B
911      005716  004537  014506  JSR      R5,DSPLA
912      005722  000001  1
913      005724  024274  FRME14
914      005726  012737  000200  024300  MOV      #200,FRM14A    ;LOAD FOURTH OCTAGON
915      005734  012737  001400  024302  MOV      #MAXY-377,FRM14B
916      005742  004537  014506  JSR      R5,DSPLA        ;DISPLAY 4TH OCT.
917      005746  000001  1
918      005750  024274  FRME14
919      :NOW DISPLAY THE SHORT VECTOR/RELATIVE POINT VERTICAL LINES
920      005752  004537  014506  JSR      R5,DSPLA        ;DISPLAY BAR
921      005756  000001  1
922      005760  036656  BUFFER
923      005762  005337  014502  DEC      TEMPA          ;FINISHED ?
924      005766  001321  BNE     1$              ;BR IF NOT
925

```



```

927 *****
(3) *TEST 24 T GRAPHPLOT INCREMENT REGISTER TEST USING GRAPHPLOT X AND Y
(3) *****
(2) 005770 000004 TST24: SCOPE
928 005772 012737 174100 024432 MOV #STATSB!INCR,GRPINC ;LOAD BASIC INCREMENT VALUE
929 006000 004537 014506 1$: JSR R5,DSPLA ;DISPLAY FRAME
930 006004 002000 2000
931 006006 024406 GRAPH
932 006010 005237 024432 INC GRPINC ;UPDATE INCR. VALUE
933 006014 022737 174110 024432 CMP #STATSB!INCR+10,GRPINC ;TEST IF #10
934 006022 001366 BNE 1$ ;BR IF NOT
935
936 *****
(3) *TEST 25 U INTENSITY LEVEL AND LIGHT PEN TEST
(3) *****
(2) 006024 000004 TST25: SCOPE
937 006026 012777 014052 173266 MOV #RET14,@LPVCT ;LOAD LIGHT PEN VECTOR
938 006034 113777 001243 173262 MOVB $VECT1+1,@LPVCT1
939 006042 042777 177400 173254 BIC #177400,@LPVCT1 ;MASK
940 006050 012737 000010 014466 MOV #10,DSAVE1 ;SET UP COUNT
941 006056 012737 027500 027370 MOV #PENOF0,MSOPEN ;RESET PEN MESSAGE #0
942 006064 012737 027560 027440 MOV #PENOF1,MS1PEN ;RESET PEN MESSAGE #1
943 006072 012700 036656 1$: MOV #BUFFER,R0 ;LOAD START ADDR.
944 006076 012737 000100 014464 MOV #100,DSAVE
945 006104 012720 117600 MOV #POINT!INT7,(R0)+ ;LOAD POINT
946 006110 012720 001400 MOV #1400,(R0)+ ;LOAD X POINT
947 006114 012720 000300 MOV #300,(R0)+ ;LOAD Y POINT
948 006120 004737 014014 JSR PC,LOADUP ;LOAD UP THE BUFFER
949 006124 012720 173400 MOV #DSTOP,(R0)+ ;LOAD DSTOP
950 006130 012720 160000 MOV #DJMP,(R0)+ ;LOAD DJUMP
951 006134 012720 036656 MOV #BUFFER,(R0)+ ;LOAD RETURN ADDRESS
952 006140 005037 014462 CLR HITCNT ;CLEAR HIT COUNT
953 006144 012737 030060 027472 MOV #30060,FRM16B-2 ;PRESET THE HIT COUNT VALUE
954 006152 012737 030060 027470 MOV #30060,FRM16B-4
955
956 006160 005037 014460 2$: CLR VIEW
957 006164 004537 014506 JSR R5,DSPLA ;EXIT TO DISPLAY FRAME
958 006170 000004 4
959 006172 027250 FRME16 ;SUB-PICTURE
960
961 006174 005237 014460 INC VIEW
962 006200 004537 014506 JSR R5,DSPLA ;EXIT TO DISPLAY FRAME
963 006204 000001 1
964 006206 036656 BUFFER
965
966 006210 005337 014464 DEC DSAVE ;FINISHED ?
967 006214 001361 BNE 2$ ;BR IF NOT MINI-LOOP
968 006216 005337 014466 DEC DSAVE1 ;FINISHED ?
969 006222 001323 BNE 1$ ;BR IF NOT
970 006224 013777 001324 173070 MOV LPVCT1,@LPVCT ;RESET VECTOR
971 006232 005077 173066 CLR @LPVCT1
  
```



```
973          ;:*****  
(3)          ;*TEST 26      V      KEYBOARD CHARACTER ECHO LOOP (MANUAL)  
(3)          ;:*****  
(2) 006236 000004  
974 006240 000532  
975          ;  
976          ; THIS FRAME MUST BE MANUALLY SELECTED VIA KEYBOARD <V> ENTRY.  
977          ; THERE IS NO EXIT EXCEPT FOR <^C>.  
978          ;  
979 006242 000004  
980 006244 012737 030060 031742  
981 006252 012737 030060 031744  
982 006260 012700 036656  
983 006264 012701 001000  
984 006270 005020  
985 006272 005301  
986 006274 001375  
987 006276 012720 160000  
988 006302 012720 031610  
989 006306 012737 161010 031750  
990 006314 005037 014460  
991 006320 012700 036656  
992 006324 012701 002000  
993 006330 012777 006364 172722  
994 006336 012777 000200 172716  
995 006344 052777 000100 172572  
996 006352 004537 014506  
997 006356 000001  
998 006360 031610  
999 006362 000736  
1000          ;  
1001          ; COME HERE ON KBD INTERRUPT. DPU IS STILL RUNNING.  
1002          ;  
1003 006364 017703 172556  
1004 006370 042703 177600  
1005 006374 020327 000003  
1006 006400 001003  
1007 006402 012716 001344  
1008 006406 000002  
1009          ;  
1010 006410 005301  
1011 006412 001436  
1012 006414 022703 000016  
1013 006420 001005  
1014 006422 005237 014460  
1015 006426 012737 164000 031750  
1016 006434 005737 014460  
1017 006440 001415  
1018 006442 022703 000017  
1019 006446 001005  
1020 006450 005037 014460  
1021 006454 012737 161010 031750  
1022 006462 122703 000037  
1023 006466 100002  
1024 006470 042703 177740  
1025 006474 110320  
  
TST26: SCOPE  
BR TST27 ;:SKIP KBD ECHO IN AUTO-RUN MODE  
  
TST26A: SCOPE  
MOV #30060,ECODEV-4 ;PRESET READOUT TO 00  
MOV #30060,ECODEV-2 ;PRESET READOUT TO 00  
20$: MOV #BUFFER,R0 ;LOAD BUFFER POINTER  
MOV #512.,R1 ;LOAD CHARACTER COUNT  
1$: CLR (R0)+ ;CLEAR THE BUFFER  
DEC R1 ;FINISHED ?  
BNE 1$ ;BR IF NOT  
MOV #DJMP,(R0)+ ;LOAD JUMP RETURN TO START OF BUFFER  
MOV #ECHOFR,(R0)+ ; THE ECHO FRAME  
MOV #161010,ECHJMP ;PRESET JUMP  
CLR SHIFTO ;CLEAR SHIFT IND.  
MOV #BUFFER,R0 ;LOAD BUFFER POINTER  
MOV #1024.,R1 ;LOAD CHARACTER COUNT  
MOV #10$,@TKBVCT ;LOAD INTR. RETURN  
MOV #200,@TKBVT1 ;LOAD RETURN INTR. LEVEL  
3$: BIS #BIT6,@$TKS ;ENABLE KEYBOARD INTR.  
JSR R5,DSPLA ;DISPLAY FRAME AND BUFFER  
1  
ECHOFR ;ADDRESS OF SUB-PICTURE  
BR 20$ ;BR UPON EXT. STOP INTERRUPT  
  
10$: MOV @$TKB,R3 ;READ KEYBOARD DATA  
BIC #177600,R3 ;MASK TO LOWER 7 BITS  
CMP R3,#3 ; <^C> ??  
BNE 11$  
MOV #BEGIN,(SP) ; RESTART IF SO.  
RTI  
  
11$: DEC R1 ;FINISHED INPUTING MAX. CHARS ?  
BEQ 12$ ;BR IF DONE MAX CHARACTERS INPUT  
CMP #16,R3 ;TEST FOR SHIFT OUT CODE  
BNE 4$ ;BR IF NOT  
INC SHIFTO ;SET SHIFT OUT FLAG  
MOV #DNOP,ECHJMP ;NOP THE BYPASS DISP. JMP  
4$: TST SHIFTO ;TEST IF SHIFT OUT  
BEQ 2$ ;BR IF NOT  
CMP #17,R3 ;TEST FOR SHIFT IN CODE  
BNE 5$ ;BR IF NOT  
CLR SHIFTO ;CLEAR SHIFT OUT FLAG  
MOV #161010,ECHJMP ;LOAD BYPASS DISP. JMP  
5$: CMPB #37,R3 ;TEST IF TOO BIG  
BPL 2$ ;BR IF NOT  
BIC #177740,R3 ;MASK OFF BITS  
2$: MOVB R3,(R0)+ ;LOAD CHARACTER INTO NEXT BUFFER LOC.
```


CZVSFA VS60 VISUAL WITH TABLET XY CORRELATOR
CZVSFA.P11 26-MAR-80 11:46 T26

MACY11 30G(1063) 26-MAR-80 11:46 PAGE 16-1
V KEYBOARD CHARACTER ECHO LOOP (MANUAL)

SEQ 0049

1026 006476 012702 031746
1027 006502 004737 014636
1028 006506 000002
1029 006510 022626 12\$:
1030 006512 013702 001264
1031 006516 052762 000200 000012
1032 006524 000655

MOV #EDEV,R2
JSR PC,KBCHR
RTI
CMP (SP)+,(SP)+
MOV DPC,R2
BIS #BIT7,12(R2)
BR 20\$

;LOAD POINTER TO ASCII CHARACTER VALUE WAS = ''
;CONVERT CHARACTER VALUE TO OCTAL
;RETURN TO WAIT
;ADJUST STACK
;GET DPC ADDRESS
;EXTERNAL STOP TO DISPLAY
;CLEAR THE BUFFER AND START AGAIN


```

1034  (3)  (3)  (2)  006526  000004  001242  001242
1035  006530  013700  160000
1036  006534  042700  006744
1037  006540  012720  000200
1038  006544  012720  007026
1039  006550  012720  000340
1040  006554  012720  007040
1041  006560  012720  000340
1042  006564  012720  007052
1043  006570  012720  000340
1044  006574  012720  000340
1045  006600  012746  000000
(1)  006604  012746  006612
(1)  006610  000002
(1)  006612

1046  1047  006612  012777  032000  172444
1048  006620  012702  000400
1049  006624  012700  001344
1050  006630  112001
1051  006632  042701  177700
1052  006636  022700  036656
1053  006642  001770
1054  006644  005037  014460
1055  006650  005301
1056  006652  001376
1057
1058  006654  052777  000200  172414
1059  006662  012703  010000
1060  006666  017737  172372  001126
1061  006674  012737  032000  001124
1062  006702  023737  001124  001126
1063  006710  101065
1064  006712  012737  032454  001124
1065  006720  023737  001124  001126
1066  006726  103462
1067  006730  005737  014460
1068  006734  001335
1069  006736  005303
1070  006740  001352
1071  006742  000425
1072  006744  105777  172326
1073  006750  100407
1074  006752  005777  172310
1075  006756  100052
1076  006760  012777  032000  172276
1077  006766  000002
1078  006770  005302
1079  006772  001407
1080  006774  012777  000001  172262
1081  007002  052737  000001  014460
1082  007010  000002
1083  007012  022626

*****
*TEST 27      W      DYNAMIC EXT. DISPLAY STOP
*****
TST27:  SCOPE
MOV      $VECT1,R0      ;LOAD VECTOR POINTER
BIC      #160000,R0     ;MASK
MOV      #4$, (R0)+     ;LOAD STOP VECTOR
MOV      #200, (R0)+
MOV      #BAD1, (R0)+   ;LOAD UNEXPT. INTR
MOV      #340, (R0)+
MOV      #BAD2, (R0)+   ;LOAD UNEXPT. INTR
MOV      #340, (R0)+
MOV      #BAD3, (R0)+   ;LOAD UNEXPT. INTR
MOV      #340, (R0)+
MOV      #PRO, -(SP)    ;;PUT NEW PS ON STACK
MOV      #64$, -(SP)   ;;PUT NEW PC ON STACK
RTI      ;;POP NEW PC AND PS

64$:
;START DISPLAY AND DELAY
MOV      #FRME17,@DPC   ;START DISPLAY
MOV      #400,R2        ;LOAD TIMER COUNTER
1$:      MOV      #BEGIN,R0 ;LOAD RANDOM NUMBER POINTER
2$:      MOVB    (R0)+,R1   ;GET A RANDOM NUMBER
        BIC      #177700,R1 ;MASK OFF OTHER BITS
        CMP      #BUFFER,R0 ;TEST IF DONE
        BEQ      1$        ;BR BACK
        CLR      DIDINT    ;CLEAR 'DID INTERRUPT ' FLAG
3$:      DEC      R1        ;DELAY
        BNE      3$

;NOW SET EXT. STOP FLAG
BIS      #BIT7,@DSR1    ;SET EXT. STOP FLAG
MOV      #BIT12,R3      ;LOAD DELAY COUNTER
7$:      MOV      @DPC,$BDDAT ;TEST IF DPC IS OUT OF RANGE
        MOV      #FRME17,$GDDAT ;LOAD LOW LIMIT
        CMP      $GDDAT,$BDDAT ;COMAPRE
        BHI      BAD4      ;BR IF TOO LOW
        MOV      #FRM17F+4,$GDDAT ;LOAD HIGH LIMIT
        CMP      $GDDAT,$BDDAT ;COMAPRE
        BLO      BAD5      ;BR IF TOO HIGH
        TST      DIDINT    ;TEST IF EXT. STOP INTR. OCCURRED
        BNE      2$        ;BR IF YES
        DEC      R3        ;DELAY
        BNE      7$        ;BR AND TEST DPC VALUE
        BR      BAD0      ;NO EXT. STOP INTR. REPORT ERROR
4$:      TSTB    @DSR1      ;TEST IF EXT. STOP FLAG
        BMI      5$        ;BR IF EXT. STOP
        TST      @DSR      ;TEST FOR DISPLAY STOP
        BPL      BAD6      ;BR IF NOT
        MOV      #FRME17,@DPC ;START DPU IF NOT EXT. STOP
        RTI      ;RETURN
5$:      DEC      R2        ;FINISHED ?
        BEQ      6$        ;BR IF DONE
        MOV      #BIT0,@DPC ;RESUME THE DPU IF EXT. STOP AND NOT FILISHED LO
        BIS      #1,DIDINT  ;SET EXT. STOP FLAG DID INTR.
        RTI      ;RETURN
6$:      CMP      (SP)+,(SP)+ ;CLEAN THE STACK
  
```


CZVSFA VS60 VISUAL WITH TABLET XY CORRELATOR
CZVSFA.P11 26-MAR-80 11:46 T27

MACY11 30G(1063) ^{M 4} 26-MAR-80 11:46 PAGE 17-1
W DYNAMIC EXT. DISPLAY STOP

SEQ 0051

1084 007014 000461

BR BADXIT ; ALL DONE, EXIT TEST.


```

1086          ;BR HERE IF AN ERROR OCCURRED
1087
1088 007016 012737 032506 032474 BAD0:  MOV  #WHY0,WHY      ;INDICATE NO EXT. STOP INTERRUPT
1089 007024 000432          BR      BADDON
1090 007026 012737 032542 032474 BAD1:  MOV  #WHY1,WHY      ;INDICATE UNEXPECTED INTR.
1091 007034 022626          CMP  (SP)+,(SP)+
1092 007036 000425          BR      BADDON
1093 007040 012737 032606 032474 BAD2:  MOV  #WHY2,WHY      ;INDICATE UNEXPECTED INTR.
1094 007046 022626          CMP  (SP)+,(SP)+
1095 007050 000420          BR      BADDON
1096 007052 012737 032652 032474 BAD3:  MOV  #WHY3,WHY      ;INDICATE UNEXPECTED INTR.
1097 007060 022626          CMP  (SP)+,(SP)+
1098 007062 000413          BR      BADDON
1099 007064 012737 032716 032474 BAD4:  MOV  #WHY4,WHY      ;INDICATE DPC WAS TOO LOW
1100 007072 000407          BR      BADDON
1101 007074 012737 032736 032474 BAD5:  MOV  #WHY5,WHY      ;INDICATE DPC WAS TOO HIGH
1102 007102 000403          BR      BADDON
1103 007104 012737 032760 032474 BAD6:  MOV  #WHY6,WHY      ;INDICATE DONE INTR. BUT NO FLAG
1104
1105
1106 007112 017737 172146 007162 BADDON: MOV  @DPC,PCERR      ;SAVE DPC
1107 007120 017737 172142 007164      MOV  @DSR,SRERR      ;SAVE SR
1108 007126 017737 172144 007166      MOV  @DSR1,SR1ERR    ;SAVE SR1
1109 007134 000240          NOP
1110 007136 000240          NOP
1111 007140 012777 014556 172150      MOV  #STOPI,@DDONE   ;LOAD DISPLAY STOP VECTOR
1112 007146 004537 014506          JSR  R5,DSPLA        ;DISPLAY ERROR MESSAGE
1113 007152 040000          BIT14
1114 007154 032452          FRM17E
1115 007156 000005          RESET
1116 007160          BADXIT:
1117 007160 000403          BR      TST30      ;;
1118
1119 007162 000000          PCERR: 0      ;D.P.C. UPON ERROR
1120 007164 000000          SRERR: 0      ;SR UPON ERROR
1121 007166 000000          SR1ERR: 0     ;SR1 UPON ERROR
  
```



```

1123      ;*****
(3)      ;*TEST 30      X      POINT CORRELATOR ALIGNMENT CHECK
(3)      ;*****
(2) 007170 000004 ;TST30: SCOPE
1124      ;
1125      ; OVERALL ALIGNMENT VERIFICATION.
1126      ; DISPLAY A 6 X 5 DOT ARRAY ON THE SCREEN.  STEER THE POINT
1127      ; CORRELATOR TO A POINT AND EXECUTE THE DISPLAY FILE FOR
1128      ; 10. CYCLES (SHOULD SEE 10 HITS).  THEN STEER THE CORRELATOR
1129      ; TO EACH OF 9 ADJACENT POINTS (+ AND - 2, AND BACK TO CENTER)
1130      ; AND GO 'ROUND AGAIN.  REPEAT ALL FOR EACH OF 30 POINTS,
1131      ; 5 OF WHICH ARE IN THE MENU AREA.
1132      ; WE SHOULD ACCUMULATE UP TO 100. HITS PER POINT.  EACH POINT
1133      ; IS THEN TURNED OFF IF IT HAS TAKEN AT LEAST 1 HIT.
1134      ; AT THE END, "ALIGNMENT ACCEPTABLE" IS DISPLAYED IF ALL POINTS
1135      ; ARE EXTINGUISHED, OTHERWISE IT SAYS "ADJUSTMENT REQUIRED".
1136      ; IN EITHER CASE, A VISIBILITY AVERAGE BASED ON THE TOTAL
1137      ; NUMBER OF HITS TAKEN (OUT OF A POSSIBLE 3000. DECIMAL),
1138      ; IS CALCULATED AND DISPLAYED.
1139      ; THE APPERTURE SIZE DEFAULTS TO 0 (MIN), AND IN MANUAL MODE,
1140      ; IS VARIABLE VIA KEYBOARD ENTRY.
1141      ;
1142      ; OK = DNOP      ; SWITCHES FOR THE TEST RESULTS...
1143      ; NOK = DSKP2
1144      ;
1145 007172 005037 007712      CLR      VMODE      ; AUTO-SEQUENCE ENTRY.
1146 007176 012737 166000 033650  MOV      #DPOP,INSTR1 ; NO INSTRUCTIONS.
1147 007204 112737 000060 012136  MOVVB   #'0,ASIZE    ; APPERTURE SIZE '0'.
1148 007212 000406      BR      T30.X
1149 007214 012737 177777 007712  T30.M:  MOV      #-1,VMODE ; MANUAL ENTRY (FROM ADJ FRAME).
1150 007222 012737 164000 033650  MOV      #DNOP,INSTR1 ; ENABLE INSTRUCTIONS.
1151      ;
1152 007230 012777 014556 172060  T30.X:  MOV      #STOPI,@DDONE ; SET DISPLAY STOP VECTOR.
1153 007236 012777 007664 172056  MOV      #T30.H,@LPVCT ; SET LP HIT VECTOR.
1154 007244 012777 000200 172052  MOV      #PR4,@LPVCT1
1155 007252 105037 012137      CLR      ABORT      ; CLEAR "ABORT" FLAG.
1156 007256 004737 012232      JSR      PC,SETM1    ; SET UP MENU CALLS.
1157      ;
1158 007262 004737 011706      1$:      JSR      PC,SETAPP   ; SET APPERTURE PARAMETERS....
1159 007266 013737 012000 033022  MOV      APPERX,PIX1 ; ...AND SET OP-CODE.
1160 007274 012737 166000 033446  MOV      #DPOP,GONOGO ; TURN OFF GO/NOGO MESSAGE.
1161 007302 012700 033052      MOV      #PIX1P,R0
1162 007306 012701 000037      MOV      #31.,R1
1163 007312 052720 040000      2$:      BIS      #INTX,(R0)+ ; INTENSIFY ALL POINTS.
1164 007316 005720      TST      (R0)+      ; BUMP PAST THE Y CO-ORD.
1165 007320 005301      DEC      R1
1166 007322 001373      BNE     2$          ; LOOP 'TIL DONE.
1167 007324 012737 177777 012144  MOV      #-1,HITK
1168 007332 004737 012142      JSR      PC,HITUP   ; CLEAR TOTAL HIT COUNTER.
1169 007336 112737 000036 007714  MOVVB   #30.,BADCNT ; SET BAD POINT COUNTER.
1170 007344 012737 164000 033024  MOV      #DNOP,PIX1+2 ; DISABLE MENU POINT SET.
1171 007352 012704 033056      MOV      #PIX1P+4,R4 ; EXPECTED DPC => R4.
1172 007356 012701 007720      MOV      #PCXY,R1  ; X POINTER => R1.
1173 007362 012137 033030      3$:      MOV      (R1)+,PIX1X ; SET 1ST/NEXT X.
1174 007366 004537 012200      13$:     JSR      R5,OCTAL
1175 007372 033030 035420      PIX1X, VT.X      ; UPDATE X READOUT.
  
```



```

1176 007376 012702 007720
1177 007402 012237 033032
1178 007406 004537 012200
1179 007412 033032 035434
1180 007416 105037 007715
1181 007422 012703 007734
1182 007426 062337 033030
1183 007432 062337 033032
1184 007436 004537 014506
1185 007442 000012 033022
1186
1187 007446 123737 012136 035462
1188 007454 001302
1189 007456 105737 012137
1190 007462 001152
1191
1192 007464 021327 177777
1193 007470 001356
1194 007472 105737 007715
1195 007476 001405
1196 007500 042764 040000 177774
1197 007506 105337 007714
1198 007512 062704 000004
1199 007516 005712
1200 007520 100330
1201 007522 023727 033024 170003
1202 007530 001413
1203 007532 005711
1204 007534 100312
1205 007536 012737 170003 033024
1206 007544 062704 000004
1207 007550 012737 000300 033030
1208 007556 000703
1209
1210 007560 005037 007716
1211 007564 013700 012144
1212 007570 001405
1213 007572 005237 007716
1214 007576 162700 000036
1215 007602 100372
1216 007604 004537 012336
1217 007610 007716 033636
1218 007614 012700 164000
1219 007620 105737 007714
1220 007624 001402
1221 007626 012700 161002
1222 007632 010037 033446
1223 007636 004537 014506
1224 007642 003720 033022
1225 007646 005737 007712
1226 007652 001402
1227 007654 000137 007262
1228 007660 000137 013712

4$: MOV #PCXY,R2 ; Y POINTER => R2.
MOV (R2)+,PIX1X+2 ; SET NEXT Y.
JSR R5,OCTAL
PIX1X+2, VT.Y ; UPDATE Y READOUT.
CLRB HITP ; CLEAR HITS/POINT COUNT.
MOV #OFFS,R3 ; OFFSET POINTER => R3.
5$: ADD (R3)+,PIX1X ; ADD CORRELATOR...
ADD (R3)+,PIX1X+2 ; ...POINT OFFSETS.
JSR R5,DSPLA ; XCT DISPLAY...
10., PIX1 ; ...FOR 10. CYCLES.

CMPB ASIZE,A.SIZ ; NOW, IN MANUAL MODE...
BNE 1$ ; ...RESTART IF APPER CHANGES...
TSTB ABORT
BNE T31 ; ...OR GO TO ADJUST ON 'ABORT'.

CMP (R3), #-1 ; OTHERWISE...
BNE 5$ ; ...LOOP FOR 10 OFFSETS/POINT.
TSTB HITP ; NOW, DID WE HIT THIS POINT ??
BEQ 6$ ; NO, LEAVE IT ON.
BIC #INTX,-4(R4) ; YES, TURN IT OFF...
DECB BADCNT ; ...AND BUMP THE 'BAD' COUNTER.
6$: ADD #4,R4 ; BUMP EXPECTED DPC IN R4.
TST (R2)
BPL 4$ ; LOOP THRU 5 Y'S.
CMP PIX1+2,#DMENU1 ; ARE WE DOING THE MENU SET ???
BEQ 7$ ; WE'RE ALL DONE IF SO.
TST (R1) ; OTHERWISE...
BPL 3$ ; ...LOOP THRU 5 X'S...
MOV #DMENU1,PIX1+2 ; ...THEN SET-UP FOR 5 IN MENU.
ADD #4,R4 ; ADJUST EXP DPC.
MOV #300,PIX1X ; X = 300 IN MENU...
BR 13$ ; ...AND GO 'ROUND FOR 5 Y'S.

7$: CLR AVG ; NOW CALCULATE HIT AVERAGE.
MOV HITK,RO
8$: BEQ 9$
INC AVG ; 1 % ...
SUB #30.,RO ; ...FOR EVERY 30. HITS.
BPL 8$
9$: JSR R5,DECMAL ; CONVERT THE AVERAGE.
AVG, VPX
MOV #OK,RO ; ALL DONE, ASSUME WE'RE OK...
TSTB BADCNT
BEQ 10$ ; ...AND SKIP NEXT IF SO.
MOV #NOK,RO ; BAD NEWS !!!
10$: MOV RO,GONOGO ; DISPLAY TEST RESULTS.
JSR R5,DSPLA
2000., PIX1
TST VMODE
BEQ +6
JMP 1$ ; LOOP FOREVER IF MANUAL...
JMP $EOP ; ...END PASS IF AUTO-SEQUENCE.

```



```
1230  
1231  
1232  
1233 007664 017700 171374  
1234 007670 020004  
1235 007672 001004  
1236 007674 105237 007715  
1237 007700 004737 012142  
1238 007704 012716 014540  
1239 007710 000002  
1240  
1241 007712 000000  
1242 007714 036  
1243 007715 000  
1244 007716 000000  
1245  
1246 007720 000002  
1247 007722 000400  
1248 007724 001000  
1249 007726 001400  
1250 007730 001775 177777  
1251  
1252 007734 000000 000000  
1253 007740 177776 177776  
1254 007744 000002 000000  
1255 007750 000002 000000  
1256 007754 000000 000002  
1257 007760 000000 000002  
1258 007764 177776 000000  
1259 007770 177776 000000  
1260 007774 000000 177776  
1261 010000 000002 000000 177777
```

COME HERE ON ANY CORRELATOR HIT.

```
T30.H: MOV @DPC,R0 ; ON CORRELATOR HIT...  
CMP RO,R4 ;...CHECK DPC.  
BNE 1$ ; BR IF IT'S WRONG...  
INCB HITP ;...OTHERWISE, BUMP HIT COUNT...  
JSR PC,HITUP ;...AND TOTAL (3000. = 5670B).  
1$: MOV #RESUME,(SP) ; FIX RETURN PC...  
RTI ;...AND RESUME DISPLAY.
```

```
VMODE: 0 ; VERIFY MODE (NZ IF ENTRY FROM ADJ).  
BADCNT: .BYTE 30. ; BAD POINT COUNTER.  
HITP: .BYTE 0 ; HITS/POINT COUNTER.  
AVG: 0 ; AVERAGE (CORRELATOR RATING).
```

```
PCXY: 2 ; MAJOR AREA X/Y POINT TABLE.  
400  
1000  
1400  
1775, -1 ; TERMINATE.
```

```
OFFS: 0, 0 ; AND OFFSET TABLE.  
-2, -2  
+2, 0  
+2, 0  
0, +2  
0, +2  
-2, 0  
-2, 0  
0, -2  
+2, 0, -1 ; IERMINATE.
```


1263
(3)
(3)
(2) 010006 000004
1264
1265
1266
1267
1268
1269
1270
1271
1272
1273
1274
1275
1276
1277
1278
1279
1280
1281
1282
1283
1284
1285
1286
1287
1288
1289 072000
1290 011756
1291 036656
1292 050634
1293 062612
1295
1296 010010 012777 012012 171242
1297 010016 012777 000200 171236
1298 010024 052777 000100 171112
1299 010032 012746 000000
(1) 010036 012746 010044
(1) 010042 000002
(1) 010044
1300 010044 012700 036656
1301 010050 012702 000031
1302 010054 012701 000062
1303 010060 012720 040001
1304 010064 012720 040200
1305 010070 005301
1306 010072 001374
1307 010074 012701 000062
1308 010100 012720 040001
1309 010104 012720 060200
1310 010110 005301
1311 010112 001374
1312 010114 005302
1313 010116 001356

```
*****
*TEST 31      Y      POINT CORRELATOR ADJUSTMENT (MANUAL)
*****
TST31: SCOPE
:
: APPERTURE SIZE, OFFSET, AND SCALE ADJUSTMENTS.
:
: PART 1 -- DISPLAY A REFERENCE BOX OVERLAYED BY A 50 SQUARE
: DOT ARRAY. BOX DIMENSIONS REFLECT THE EXPECTED APPERTURE
: SIZE ACCORDING TO SPEC. THE POINT CORRELATOR IS POINTED
: TO THE CENTER OF THE ARRAY. THOSE POINTS WHICH TAKE HITS
: ARE TURNED OFF, RESULTING IN A DARK AREA WHICH REPRESENTS
: THE ACTUAL APPERTURE SIZE, SHAPE, AND POSITION.
: ANY CENTERING ERROR IS INTERPRETED AS X/Y OFFSET.
: ADJUST SIZE TO SPEC, AND OFFSETS TO CENTER THE WINDOW (NULL
: THE OFFSET ERRORS). TYPE <X> OR <Y> TO PROCEED TO...
:
: ...PART 2 -- REPLICATE THE TEST PATTERN AND DISPLAY AT
: LEFT-CENTER AND RIGHT-CENTER FOR X AXIS ADJUSTMENT, OR
: BOTTOM-CENTER AND TOP-CENTER FOR Y AXIS ADJUSTMENT.
: CALCULATE THE CENTERING ERROR AT EACH POSITION, AND
: EVALUATE AND DISPLAY THE OFFSET AND SCALE COMPONENTS.
: ADJUST X (OR Y) SCALE AND OFFSET TO NULL THE ERRORS.
: TYPE <X> OR <Y> TO SWAP THE AXES.
: TYPE <C> TO RETURN TO CENTER DISPLAY (PART 1).
: TYPE <CR> TO GO BACK TO TEST 30 (MANUAL) FOR VERIFICATION.
:
: FIRST REDEFINE THE BUFFER AND INSURE ROOM FOR 2 DOT ARRAYS.
:
: BFR16K= 72000      ; END OF BUFFER SPACE UNDER 16K XXDP.
: DOTS= 2551.*2    ; NUMBER OF BYTES REQUIRED FOR 1 ARRAY.
: BUFR1= BUFFER    ; 1ST ARRAY ADDRESS.
: BUFR2= BUFR1+DOTS ; 2ND ARRAY ADDRESS.
: BFREND= BUFR2+DOTS
:
T31:  MOV    #PCKBD,@TKBVCT ; CHANGE KBD VECTOR...
      MOV    #PR4,@TKBVT1
      BIS    #BIT6,@$TKS    ;...AND ENABLE INTERRUPTS.
      MOV    #PRO,-(SP)     ;:PUT NEW PS ON STACK
      MOV    #64$,-(SP)    ;:PUT NEW PC ON STACK
      RTI                    ;:POP NEW PC AND PS
64$:  MOV    #BUFR1,R0      ; FILL DISPLAY BUFFER WITH...
      MOV    #25.,R2       ;...A 50 SQUARE DOT ARRAY.
1$:   MOV    #50.,R1
      MOV    #INTX!1,(R0)+ ; Y+1
2$:   MOV    #INTX!200,(R0)+ ; X+1, 50 TIMES.
      DEC    R1
      BNE   2$
      MOV    #50.,R1
3$:   MOV    #INTX!1,(R0)+ ; Y+1
      MOV    #INTX!20200,(R0)+ ; X-1. 50 TIMES.
      DEC    R1
      BNE   3$
      DEC    R2
      BNE   1$ ; REPEAT ALL 25 TIMES.
```


1314 010120 012720 166000
 1315
 1316 010124 012700 036656
 1317 010130 012701 050634
 1318 010134 012021
 1319 010136 100376
 1320
 1321
 1322
 1323 010140 012737 010726 010716
 1324 010146 012737 164000 033332
 1325 010154 012737 001000 033300
 1326 010162 012737 001000 033302
 1327 010170 004737 012274
 1328 010174 105037 012137
 1329 010200 105037 012140
 1330 010204 000466
 1331
 1332
 1333
 1334
 1335 010206 012737 010776 010716
 1336 010214 012737 166000 035574
 1337 010222 012737 164000 035644
 1338 010230 012737 161004 033332
 1339 010236 012700 010326
 1340 010242 123727 012140 000130
 1341 010250 001402
 1342 010252 012700 010344
 1343 010256 012037 033300
 1344 010262 012037 033302
 1345 010266 012037 033346
 1346 010272 012037 033350
 1347 010276 012037 034710
 1348 010302 012037 035002
 1349 010306 112037 035652
 1350 010312 113737 035652 035676
 1351 010320 112037 035115
 1352 010324 000416
 1353
 1354 010326 000200 001000
 1355 010332 001600 001000
 1356 010336 032060
 1357 010340 031060
 1358 010342 130
 1359 010343 131
 1360 010344 001000 000200
 1361 010350 001000 001600
 1362 010354 033461
 1363 010356 032461
 1364 010360 131
 1365 010361 130

```

MOV #DPOP,(R0)+ ; TERMINATE DISPLAY BUFFER.
; BUFFER SIZE = 2551. WORDS.
MOV #BUFR1,R0
MOV #BUFR2,R1
MOV (R0)+,(R1)+ ; NOW COPY BUFR1 => BUFR2.
BPL .-2

; SET-UP FOR SIZE AND INITIAL OFFSET ADJUSTMENT (PART 1).
T31.A: MOV #PCH1,PCH.SW ; SET INTERRUPT SWITCH.
MOV #DNOP,PIX2SW ; SET DISPLAY FOR SINGLE...
MOV #1000,PIX2X1 ; ...PATTERN AT CENTER SCREEN.
MOV #1000,PIX2X1+2
JSR PC,SETM2 ; INIT MENU CALLS.
CLRB ABORT ; CLEAR "ABORT" FLAG.
CLRB NUAXIS ; NU = 0 = CENTER.
BR T31.C ; GO TO COMMON CODE.

; SET-UP FOR OFFSET AND SCALE FINE TUNING (PART 2).
; "NUAXIS" = X OR Y ON ENTRY.
T31.B: MOV #PCH2,PCH.SW ; CHANGE INTERRUPT SWITCH.
MOV #DPOP,CTR.ER ; CHANGE THE ERROR SWITCHES.
MOV #DNCP,OAS.ER
MOV #DSKP4,PIX2SW ; ENABLE DUAL PATTERN DISPLAY.
MOV #TBLX,R0 ; ASSUME X AXIS.
CMPB NUAXIS,#'X
BEQ 1$ ; BR IF SO...
MOV #TBLX,R0 ; ...OTHERWISE, USE Y AXIS.
1$: MOV (R0)+,PIX2X1 ; SET LO X/Y...
MOV (R0)+,PIX2X1+2 ; ...HI X/Y...
MOV (R0)+,PIX2X2+2
MOV (R0)+,S.POT ; ...SCALE...
MOV (R0)+,O.POT ; ...AND OFFSET POT NUMBERS...
MOVB (R0)+,O.ID
MOVB O.ID,S.ID ; ...AXIS IDENTIFIERS...
MOVB (R0)+,OTHRAX+1 ; ...AND "OTHER" AXIS TEXT.
BR T31.C

TBLX: 200, 1000 ; PATTERN CO-ORDS (X ADJUST).
1600, 1000
.ASCII /04/ ; SCALE POT IS R04.
.ASCII /02/ ; OFFSET IS R02.
.ASCII /X/ ; CURRENT AXIS...
.ASCII /Y/ ; ...OTHER AXIS.
TBLX: 1000, 200 ; SAME SHIT FOR Y ADJUST.
1000, 1600
.ASCII /17/
.ASCII /15/
.ASCII /Y/
.ASCII /X/
  
```



```

1367
1368
1369
1370
1371 010362 012777 010540 170726 T31.C: .ENABL LSB
1372 010370 012777 010640 170724 MOV #10$,@DDONE ; SET STOP VECTOR...
1373 010376 113737 012140 012141 MOV #PCH,@LPVCT ; ...AND HIT VECTOR.
1374 010404 004737 011706 012141 MOVBU NUAXIS,AXIS ; SET CURRENT AXIS FLAG.
1375 010410 013737 012000 033274 1$: JSR PC,SETAPP ; SET APPERTURE PARAMETERS...
1376 010416 004737 011634 MOV APPERX,PIX2 ; ...AND OPCODE...
1377 010422 012777 033274 170634 JSR PC,REFBOX ; ...AND SET REF BOX SIZE.
1378 MOV #PIX2,@DPC ; START/RESTART THE DISPLAY.
1379 ; BUFFERS WILL BE REFRESHED...
1380 ; ...ON-THE-FLY TO MINIMIZE...
1381 010430 012737 000240 010640 2$: MOV #NOP,PCH ; ...DISPLAY FLICKERING.
1382 010436 012700 040000 MOV #INTX,R0 ; DISALLOW PC HITS WHILE WE...
1383 010442 012701 036656 MOV #BUFR1,R1
1384 010446 012702 050634 MOV #BUFR2,R2
1385 010452 050021 BIS RO,(R1)+ ; ...REFRESH BUFFER #1...
1386 010454 100376 BPL .-2
1387 010456 105737 012141 TSTB AXIS
1388 010462 001402 BEQ 3$ ; ...AND BUFFER #2 (IF REQ'D)...
1389 010464 050022 BIS RO,(R2)+
1390 010466 100376 BPL .-2
1391 010470 012700 001777 3$: MOV #1777,R0
1392 010474 012701 011252 MOV #X1LO,R1
1393 010500 010021 4$: MOV RO,(R1)+ ; ...AND RESET X/Y COMPARATORS.
1394 010502 005021 CLR (R1)+
1395 010504 005021 CLR (R1)+
1396 010506 020127 011302 CMP R1,#Y2DF+2
1397 010512 001372 BNE 4$
1398
1399 010514 012737 000144 010542 MOV #100.,CYCL ; NOW RESET THE CYCLE COUNTER...
1400 010522 012737 000401 010640 CNTNU: MOV #401,PCH ; ...AND START TAKING HITS.
1401 010530 005737 010542 TST CYCL ; NOTHING ELSE TO DO...
1402 010534 001375 BNE .-4 ; ...UNTIL CYCLE K EXPIRES.
1403 010536 000734 BR 2$ ; THEN REFRESH THE BUFFERS.
1404
1405 ; COME HERE ON DISPLAY STOP.
1406
1407 010540 005327 10$: DEC (PC)+ ; BUMP THE CYCLE COUNT.
1408 010542 000000 CYCL: 0
1409 010544 105737 012137 TSTB ABORT
1410 010550 001403 BEQ 11$
1411 010552 012716 007214 MOV #T30.M,(SP) ; "ABORT", XCT VERIFY FRAME.
1412 010556 000002 RTI
1413 010560 123737 012136 035462 11$: CMPB ASIZE,A.SIZ
1414 010566 001403 BEQ 12$
1415 010570 012716 010404 MOV #1$, (SP) ; CHANGE APPERTURE SETTING.
1416 010574 000002 RTI
1417 010576 123737 012140 012141 12$: CMPB NUAXIS,AXIS
1418 010604 001004 BNE 13$ ; BR IF AXES ARE CHANGING.
1419 010606 012777 000001 170450 MOV #1,@DPC ; OTHERWISE, RESUME DISPLAY...
1420 010614 000002 RTI ; ...AND DISMISS INTERRUPT.
1421
1422 010616 105737 012140 13$: TSTB NUAXIS
    
```


CZVSFA VS60 VISUAL WITH TABLET XY CORRELATOR
CZVSFA.P11 26-MAR-80 11:46 T31

H 5
MACY11 30G(1063) 26-MAR-80 11:46 PAGE 20-3
Y POINT CORRELATOR ADJUSTMENT (MANUAL)

SEQ 0059

1423 010622 001403
1424 010624 012716 010206
1425 010630 000002
1426 010632 012716 010140
1427 010636 000002
1428

14\$:

BEQ 14\$
MOV #T31.B,(SP) ; RESTART DUAL PATTERN.
RTI
MOV #T31.A,(SP) ; RESTART SINGLE PATTERN.
RTI
.DSABL LSB


```

1430
1431
1432
1433
1434
1435
1436
1437 010640 000401
1438 010642 000426
1439 010644 017737 170420 014464
1440 010652 042737 176000 014464
1441 010660 017737 170406 014466
1442 010666 042737 176000 014466
1443 010674 017700 170364
1444 010700 042760 040000 177776
1445 010706 012737 000240 010640
1446 010714 012716
1447 010716 010726
1448 010720 005277 170340
1449 010724 000002
1450
1451
1452
1453 010726 004737 011072
1454 010732 012700 001000
1455 010736 012701 011252
1456 010742 012702 035612
1457 010746 004737 011302
1458 010752 012700 001000
1459 010756 012701 011260
1460 010762 012702 035634
1461 010766 004737 011302
1462 010772 000137 010522
1463
1464
1465
1466
1467 010776 020027 050634
1468 011002 103011
1469 011004 004737 011072
1470 011010 012700 000200
1471 011014 012701 011252
1472 011020 012702 011376
1473 011024 000410
1474 011026 004737 011100
1475 011032 012700 001600
1476 011036 012701 011266
1477 011042 012702 011402
1478 011046 123727 012141 000130
1479 011054 001402
1480 011056 062701 000006
1481 011062 004737 011302
1482
1483 011066 000137 010522
  
```

```

: PC HIT INTERRUPT SERVICE ENTRY.
: HIT SERVICE DISABLED/ENABLED VIA 'PCH'.
: SAVE PC, X, AND Y, AND DISALLOW FURTHER HITS 'TIL WE'VE
: FINISHED PROCESSING THIS ONE. ALLOW DISPLAY TO RUN TO
: MINIMIZE FLICKERING.
:
PCH: 401 ; SKIP NEXT (TAKE HITS).
BR PCH.SW+2 ; IGNORE HITS WHEN (PCH) = NOP.
MOV @XPOS, DSAVE ; SAVE VTX...
BIC #^C1777, DSAVE
MOV @YPOS, DSAVE1 ; ...AND Y FOR DIMEN ROUTINE.
BIC #^C1777, DSAVE1
MOV @DPC, R0 ; SAVE DPC IN R0...
BIC #INTX, -2(R0) ; ...TURN OFF THIS POINT.
MOV #NOP, PCH ; DISALLOW FURTHER HITS.
MOV (PC)+, (SP) ; CHANGE STACKED PC TO...
PCH.SW: PCH1 ; ...CONTINUE AT PCH1 OR PCH2.
INC @DPC ; RESUME DISPLAY...
RTI ; ...AND DISMISS INTERRUPT.
:
: PART 1 -- INTERPRET X/Y CENTERING ERROR AS PURE OFFSET.
:
PCH1: JSR PC, DIMEN1 ; DISPLAY APPERT DIMENSIONS.
MOV #1000, R0 ; SET EXPECTED X CENTER...
MOV #X1LO, R1 ; ...ACTUAL X DATA POINTER...
MOV #X.VAL, R2 ; ...AND TEXT POINTER.
JSR PC, CTRERR ; CALCULATE AND DISPLAY X OFFSET
MOV #1000, R0
MOV #Y1LO, R1
MOV #Y.VAL, R2
JSR PC, CTRERR ; CALCULATE AND DISPLAY Y OFFSET
JMP CNTNU ; CONTINUE CYCLING.
:
: PART 2 -- DUAL PATTERN. LO AND HI ERRORS ARE EVALUATED
: TO EXTRACT THE OFFSET AND SCALE COMPONENTS.
:
PCH2: CMP R0, #BUFR2 ; TEST DPC RANGE.
BHS 1$ ; BR IF WE'RE IN THE 2ND HALF.
JSR PC, DIMEN1 ; SET LO BOX DIMENSIONS...
MOV #200, R0 ; ...EXPECTED LO CENTER...
MOV #X1LO, R1 ; ...DATA POINTER...
MOV #LOMAG, R2 ; ...AND ERROR BUFFER POINTER.
BR 2$
1$: JSR PC, DIMEN2 ; SET HI BOX DIMENSIONS...
MOV #1600, R0 ; ...EXPECTED HI CENTER...
MOV #X2LO, R1 ; ...DATA POINTER...
MOV #HIMAG, R2 ; ...AND ERROR BUFFER POINTER.
2$: CMPB AXIS, #'X
BEQ 3$ ; SKIP IF WE'RE DOING X...
ADD #6, R1 ; ...OTHERWISE, POINT TO Y DATA.
3$: JSR PC, CTRERR ; EVALUATE CENTERING ERRORS...
; ...IN TERMS OF OFFSET AND SCALE
; AND CONTINUE CYCLING.
JMP CNTNU
  
```



```

1485
1486      ; SUBROUTINE TO CALCULATE ACTUAL APPERTURE DIMENSIONS.
1487      ; ACCUMULATE DIMENSION DATA IN 1 OF 2 DATA BUFFERS.
1488      ; DISPLAY DIMENSIONS OF POINT SET 1.
1489      ; VT X AND Y ARE SAVED IN 'DSAVE' AND 'DSAVE1'.
1490
1491 011072 012700 011252 DIMEN1: MOV #X1LO,RO ; POINT 1 DATA BUFFER.
1492 011076 000402 BR .+6
1493 011100 012700 011266 DIMEN2: MOV #X2LO,RO ; POINT 2 DATA BUFFER.
1494 011104 023710 014464 CMP DSAVE,(RO)
1495 011110 103002 BHIS 1$
1496 011112 013710 014464 MOV DSAVE,(RO) ; SAVE LOWEST X...
1497 011116 023760 014464 000002 1$: CMP DSAVE,2(RO)
1498 011124 101403 BLOS 2$
1499 011126 013760 014464 000002 MOV DSAVE,2(RO) ;...AND HIGHEST X.
1500 011134 016060 000002 000004 2$: MOV 2(RO),4(RO)
1501 011142 161060 000004 SUB (RO),4(RO) ; ACCUMULATE X DIFFERENCE.
1502
1503 011146 062700 000006 ADD #6,RO
1504 011152 023710 014466 CMP DSAVE1,(RO)
1505 011156 103002 BHIS 3$
1506 011160 013710 014466 MOV DSAVE1,(RO) ; SAVE LOWEST Y...
1507 011164 023760 014466 000002 3$: CMP DSAVE1,2(RO)
1508 011172 101403 BLOS 4$
1509 011174 013760 014466 000002 MOV DSAVE1,2(RO) ;...AND HIGHEST Y.
1510 011202 016060 000002 000004 4$: MOV 2(RO),4(RO)
1511 011210 161060 000004 SUB (RO),4(RO) ; ACCUMULATE Y DIFFERENCE.
1512
1513 011214 004537 012336 JSR R5,DECMAL
1514 011220 011256 012514 X1DF, HI2 ; CONVERT TO TEMP.
1515 011224 013737 012520 035532 MOV LO2,X.DIM ; DISPLAY X...
1516 011232 004537 012336 JSR R5,DECMAL
1517 011236 011264 012514 Y1DF, HI2
1518 011242 013737 012520 035540 MOV LO2,Y.DIM ;...AND Y DIMENSIONS.
1519 011250 000207 RTS PC
1520
1521 011252 001777 X1LO: 1777 ; POINT 1 DATA ACCUMULATOR.
1522 011254 000000 X1HI: 0
1523 011256 000000 X1DF: 0
1524 011260 001777 Y1LO: 1777
1525 011262 000000 Y1HI: 0
1526 011264 000000 Y1DF: 0
1527 011266 001777 X2LO: 1777 ; POINT 2 DATA ACCUMULATOR.
1528 011270 000000 X2HI: 0
1529 011272 000000 X2DF: 0
1530 011274 001777 Y2LO: 1777
1531 011276 000000 Y2HI: 0
1532 011300 000000 Y2DF: 0

```



```

1534
1535
1536
1537
1538
1539
1540
1541
1542
1543
1544
1545
1546
1547
1548
1549
1550 011302 012703 011376
1551 011306 020027 001000
1552 011312 001402
1553 011314 010203
1554 011316 005002
1555
1556 011320 016113 000004
1557 011324 006213
1558 011326 061113
1559 011330 160013
1560 011332 010237 011346
1561 011336 001407
1562 011340 004537 012336
1563 011344 011376
1564 011346 035612
1565 011350 005077 177772
1566 011354 000207
1567
1568 011356 005063 000002
1569 011362 005713
1570 011364 100003
1571 011366 005413
1572 011370 005163 000002
1573 011374 000404
1574
1575 011376 000000
1576 011400 000000
1577 011402 000000
1578 011404 000000

: SUBROUTINE TO CALCULATE X OR Y CENTERING ERROR.
: IF AT CENTER SCREEN, DISPLAY ERROR AS X (OR Y) OFFSET.
: IF DUAL PATTERN, EVALUATE AND DISPLAY ERROR IN TERMS OF
: OFFSET AND SCALE.
:
: ON ENTRY:      R0 = EXPECTED CENTER VALUE.
:                R1 = POINTER TO ACTUAL LO-HI-DIF TRIAD.
:                R2 = POINTER TO DESTINATION TEXT ADDRESS,
:                OR
:                POINTER TO LOMAG/HIMAG BUFFERS.
:
: ON EXIT:       LOMAG = ERROR MAGNITUDE (CENTER OR LO PATTERN).
:                HIMAG = ERROR MAGNITUDE (HI PATTERN).
:                LOSGN, HISGN = 0 (POS) OR -1 (NEG) ERROR SIGNS.
:
CTRERR: MOV      #LOMAG,R3      ; ASSUME WE'LL USE THE LO SIDE.
        CMP      R0,#1000
        BEQ      1$           ; BR IF PATTERN IS AT CENTER.
        MOV      R2,R3       ; LOMAG OR HIMAG POINTER => R3.
        CLR      R2          ; R2 = 0 = EVALUATE ERRORS.
:
1$:     MOV      4(R1),(R3)    ; DIFFERENCE.
        ASR      (R3)         ; 1/2 THE DIFFERENCE...
        ADD      (R1),(R3)    ; ...PLUS LO VAL = ACTUAL CENTER.
        SUB      R0,(R3)     ; ACTUAL-EXPECTED = ERROR.
        MOV      R2,2$
        BEQ      3$           ; BR IF EVALUATING.
        JSR      R5,DECIMAL  ; OTHERWISE, DISPLAY...
        LOMAG   X.VAL        ; ...SIGNED ERROR...
        CLR      @2$         ; ...AT X.VAL (OR Y.VAL)...
        RTS     PC          ; ...LO FOUR DIGITS ONLY...
:                ; ...AND EXIT.
:
3$:     CLR      2(R3)        ; ASSUME A POS ERROR VALUE.
        TST     (R3)
        BPL     4$           ; BR IF SO.
        NEG     (R3)         ; ELSE, MAKE ABSOLUTE...
        COM     2(R3)        ; ...CHANGE SIGN...
        BR      EVALU8      ; ...AND CONTINUE.
:
4$:     LOMAG: 0             ; LO ERROR MAGNITUDE (ABS)...
        LOSGN: 0             ; LO ERROR SIGN (0 OR -1).
        HIMAG: 0             ; DITTO HI.
        HISGN: 0

```



```

1580
1581      ; THIS IS THE ERROR EVALUATOR.
1582
1583 011406 013700 011402      EVALUB: MOV      HIMAG,R0
1584 011412 163700 011376      SUB      LOMAG,R0      ; FIRST, MAKE DIFFERENCE...
1585 011416 100001              BPL      .+4           ; ...SKIP IF POS.
1586 011420 005400              NEG      R0
1587 011422 006200              ASR      R0           ; 1/2 ABS DIFF => R0 (VAL 1).
1588 011424 013701 011402      MOV      HIMAG,R1
1589 011430 013702 011404      MOV      HISGN,R2
1590 011434 020137 011376      CMP      R1,LOMAG     ; LARGEST MAG,,SIGN => R1,,R2.
1591 011440 103004              BHIS    4$
1592 011442 013701 011376      MOV      LOMAG,R1
1593 011446 013702 011400      MOV      LOSGN,R2
1594 011452 160001      4$:  SUB      R0,R1      ; LRGST - DIFF => R1 (VAL 2).
1595 011454 010237 011626      MOV      R2,OSGN     ; OFFSET TAKES SIGN OF LARGEST.
1596 011460 005737 011402      TST      HIMAG       ; NOW IF HI ERROR IS NON-ZERO...
1597 011464 001404              BEQ     5$
1598 011466 013737 011404 011632  MOV      HISGN,SSGN  ; ...SCALE TAKES THAT SIGN.
1599 011474 000407              BR      6$
1600 011476 013737 011626 011632  5$:  MOV      OSGN,SSGN  ; OTHERWISE, SCALE SIGN IS...
1601 011504 012702 000006      MOV      #6,R2
1602 011510 074237 011632      XOR      R2,SSGN     ; ...OPPOSITE THE OFFSET SIGN.
1603
1604 011514 023737 011400 011404  6$:  CMP      LOSGN,HISGN ; NOW COMPARE ORIGINAL SIGNS.
1605 011522 001405              BEQ     7$
1606 011524 010037 011624      MOV      R0,OVAL     ; IF UNLIKE, VAL 1 IS OFFSET...
1607 011530 010137 011630      MOV      R1,SVAL     ; ...AND VAL 2 IS SCALE.
1608 011534 000404              BR      8$
1609 011536 010137 011624      7$:  MOV      R1,OVAL     ; IF LIKE, VAL 2 IS OFFSET...
1610 011542 010037 011630      MOV      R0,SVAL     ; ...AND VAL 1 IS SCALE.
1611
1612 011546 005737 011626      8$:  TST      OSGN
1613 011552 100002              BPL      .+6
1614 011554 005437 011624      NEG      OVAL
1615 011560 004537 012336      JSR     R5,DECMAL
1616 011564 011624 035664      OVAL, O.VAL      ; ENCODE OFFSET...
1617 011570 005037 035664      CLR     O.VAL     ; ...LO FOUR DIGITS ONLY.
1618 011574 005737 011632      TST      SSGN
1619 011600 100002              BPL      .+6
1620 011602 005437 011630      NEG      SVAL
1621 011606 004537 012336      JSR     R5,DECMAL
1622 011612 011630 035710      SVAL, S.VAL     ; ENCODE SCALE...
1623 011616 005037 035710      CLR     S.VAL     ; ...LO FOUR DIGITS ONLY.
1624 011622 000207              RTS     PC
1625
1626 011624 000000      OVAL:  0           ; OFFSET VALUE...
1627 011626 000000      OSGN:  0           ; ...AND ASCII SIGN.
1628 011630 000000      SVAL:  0           ; DITTO SCALE.
1629 011632 000000      SSGN:  0

```



```

1631
1632      ; SUBROUTINE TO SET REFERENCE BOX DIMENSIONS.
1633
1634 011634 013701 011766 REFBOX: MOV     SIZX,R1      ; CURRENT SIZE => R1.
1635 011640 010100      MOV     R1,R0
1636 011642 006200      ASR     R0              ; 1/2 SIZE => R0.
1637 011644 012702 001777 MOV     #1777,R2       ; MASK => R2.
1638 011650 012703 033424 MOV     #RBX,R3       ; BOX VECTOR ADDRESS => R3.
1639 011654 040213      BIC     R2,(R3)
1640 011656 050023      BIS     R0,(R3)+      ; NOW, SET 1ST VECTOR...
1641 011660 040213      BIC     R2,(R3)
1642 011662 050123      BIS     R1,(R3)+      ;...2ND...
1643 011664 040213      BIC     R2,(R3)
1644 011666 050123      BIS     R1,(R3)+      ;...3RD...
1645 011670 040213      BIC     R2,(R3)
1646 011672 050123      BIS     R1,(R3)+      ;...4TH...
1647 011674 040213      BIC     R2,(R3)
1648 011676 050123      BIS     R1,(R3)+      ;...5TH...
1649 011700 040213      BIC     R2,(R3)
1650 011702 050023      BIS     R0,(R3)+      ;...AND THE 6TH.
1651 011704 000207      RTS     PC
1652
1653      ; SUBROUTINE TO SET THE CURRENT APPERTURE PARAMETERS.
1654
1655 011706 113700 012136 SETAPP: MOVB   ASIZE,R0      ; SELECTED SIZE (ASCII 0-3).
1656 011712 110037 035462 MOVB   R0,A.SIZ      ; STUFF IN MENU "APPERTURE".
1657 011716 042700 177774 BIC     #^C3,R0
1658 011722 006300      ASL     R0
1659 011724 016037 011756 011766 MOV     SIZTBL(R0),SIZX ; SET CURRENT SIZE...
1660 011732 016037 011770 012000 MOV     APPTBL(R0),APPERX ;...AND THE CORRECT OPCODE.
1661 011740 016037 012002 035506 MOV     ASZTBL(R0),A.SPC ; NOW SET "SPEC" DIMENSIONS...
1662 011746 013737 035506 035514 MOV     A.SPC,A.SPC1 ;...IN THE MENU.
1663 011754 000207      RTS     PC
1664
1665 011756 000004 SIZTBL: 4.      ; SIZE 0      4 X 4
1666 011760 000006      6.      ;             6 X 6
1667 011762 000010      8.      ;             8 X 8
1668 011764 000012      10.     ;             10 X 10
1669 011766 000004 SIZX:  4.      ; ONE OF THE ABOVE (DEFAULT SIZE 0).
1670
1671 011770 175000 APPTBL: APPER0   ; APPERTURE OPCODE TABLE.
1672 011772 175200      APPER1
1673 011774 175400      APPER2
1674 011776 175600      APPER3
1675 012000 175000 APPERX: APPER0   ; ONE OF THE ABOVE (DEFAULT 0).
1676
1677 012002 000064 ASZTBL: .ASCII  /4/<0> ; ASCII SPEC DATA FOR THE MENU.
1678 012004 000066      .ASCII  /6/<0>
1679 012006 000070      .ASCII  /8/<0>
1680 012010 030061      .ASCII  /10/
    
```



```

1682
1683      ; SUBROUTINE TO HANDLE KBD FOR POINT CORRELATOR FUNCTIONS.
1684      ; ACCEPT 0, 1, 2, AND 3 FOR APPERTURE CHANGES,
1685      ; AND <C>, <X>, AND <Y> FOR SWAPPING THE OFFSET/SCALE FRAMES.
1686
1687      PCKBD:
(2) 012012 010046      MOV      RO,-(SP)      ;;PUSH RO ON STACK
1688 012014 017700 167126      MOV      @TKB,RO      ; GET CHARACTER...
1689 012020 042700 177600      BIC      #^C177,RO    ;...AND STRIP IT.
1690 012024 020027 000060      CMP      RO,#'0
1691 012030 103406      BLO      1$
1692 012032 020027 000063      CMP      RO,#'3
1693 012036 101003      BHI      1$
1694 012040 110037 012136      MOVB     RO,ASIZE     ; SAVE NEW SIZE SELECTION.
1695 012044 000432      BR
1696
1697 012046 020027 000103      1$:     CMP      RO,#'C
1698 012052 001002      BNE      2$
1699 012054 105037 012140      CLRB     NUAXIS      ; SET "CENTER" FLAG.
1700 012060 020027 000130      2$:     CMP      RO,#'X
1701 012064 001002      BNE      3$
1702 012066 110037 012140      MOVB     RO,NUAXIS   ; SET "X" FLAG.
1703 012072 020027 000131      3$:     CMP      RO,#'Y
1704 012076 001002      BNE      4$
1705 012100 110037 012140      MOVB     RO,NUAXIS   ; SET "Y" FLAG.
1706 012104 020027 000015      4$:     CMP      RO,#15
1707 012110 001002      BNE      5$
1708 012112 110037 012137      MOVB     RO,ABORT    ; SET "ABORT" FLAG.
1709 012116 020027 000003      5$:     CMP      RO,#3
1710 012122 001003      BNE      6$
1711 012124 012766 001344 000002      MOV      #BEGIN,2(SP) ; RESTART ON <^C>.
1712 012132      6$:     MOV      (SP)+,RO    ;;POP STACK INTO RO
(2) 012132 012600
1713 012134 000002      RTI
1714
1715 012136      060      ASIZE: .BYTE 60      ; SELECTED APPERTURE (ASCII).
1716 012137      000      ABORT: .BYTE 0      ; ABORT FLAG.
1717 012140      000      NUAXIS: .BYTE 0     ; NEW AXIS (CHANGE) FLAG.
1718 012141      000      AXIS: .BYTE 0     ; CURRENT AXIS (0, X, OR Y).

```



```

1720
1721      ; SUBROUTINE TO RECORD AND DISPLAY THE HIT COUNTER.
1722
1723 012142 005227 HITUP: INC (PC)+
1724 012144 000000 HITK: 0 ; HIT COUNTER.
1725 012146 023727 012144 023417 CMP HITK,#9999.
1726 012154 003402 BLE 1$
1727 012156 005037 012144 CLR HITK ; WRAP-AROUND AT 10K.
1728 012162 004537 012336 1$: JSR R5,DECMAL
1729 012166 012144 035564 HITK, HIT.K ; SET IN MENU.
1730 012172 005037 035564 CLR HIT.K ; 4 DIGITS ONLY.
1731 012176 000207 RTS PC
1732
1733      ; SUBROUTINE TO MAKE A STANDARD CALL ON SHOOP'S
1734      ; 4 DIGIT OCTAL TO ASCII ENCODER.
1735      ; CALL: JSR R5,OCTAL
1736      ; SRC ADDRESS
1737      ; DST ADDRESS
1738      ; RETURN
1739
1740      ; OCTAL:
1741 (2) 012200 010246 MOV R2,-(SP) ;:PUSH R2 ON STACK
1742 012202 010346 MOV R3,-(SP) ;:PUSH R3 ON STACK
1743 012204 010446 MOV R4,-(SP) ;:PUSH R4 ON STACK
1744 012206 013503 MOV @R5+,R3 ; VALUE => R3.
1745 012210 012502 MOV (R5)+,R2 ; ADDRESS OF DEST STRING => R2.
1746 012212 062702 000004 ADD #4,R2 ; POINT TO LO DIGIT+1.
1747 012216 004737 014636 JSR PC,KBCHR ; DO IT.
1748 012222 012604 MOV (SP)+,R4 ;:POP STACK INTO R4
1749 012224 012603 MOV (SP)+,R3 ;:POP STACK INTO R3
1750 012226 012602 MOV (SP)+,R2 ;:POP STACK INTO R2
1751 012230 000205 RTS R5
1752
1753      ; SUBROUTINES TO SET MENU PARAMETER CALLS.
1754 012232 012700 166000 SETM1: MOV #DPOP,R0 ; SET-UP FOR FOR ALIGN CHECK.
1755 012236 012701 164000 MOV #DNOP,R1
1756 012242 010137 035406 MOV R1,VTXY ; X/Y ON
1757 012246 010137 035442 MOV R1,APPERT ; APPERTURE ON
1758 012252 010037 035466 MOV R0,SPECS ; SPECS OFF
1759 012256 010137 035544 MOV R1,HITS ; HITS ON
1760 012262 010037 035574 MOV R0,CTR.ER ; ERRORS OFF
1761 012266 010037 035644 MOV R0,OAS.ER
1762 012272 000207 RTS PC
1763
1764 012274 012700 166000 SETM2: MOV #DPOP,R0 ; SET-UP FOR ADJUSTMENTS.
1765 012300 012701 164000 MOV #DNOP,R1
1766 012304 010037 035406 MOV R0,VTXY ; X/Y OFF
1767 012310 010137 035442 MOV R1,APPERT ; APPERTURE ON
1768 012314 010137 035466 MOV R1,SPECS ; SPECS ON
1769 012320 010037 035544 MOV R0,HITS ; HITS OFF
1770 012324 010137 035574 MOV R1,CTR.ER ; CENTER ERROR ON (INITIAL)
1771 012330 010037 035644 MOV R0,OAS.ER ; OFFSET/SCALE OFF (INITIAL)
1772 012334 000207 RTS PC
    
```



```

1774
1775
1776
1777
1778
1779
1780
1781
1782 012336 012746 012450
1783 012342 000402
1784 012344 012746 012504
1785 012350 010046
1786 012352 010146
1787 012354 013500
1788 012356 012501
1789 012360 112711 000040
1790 012364 005700
1791 012366 100003
1792 012370 005400
1793 012372 112711 000055
1794 012376 004537 012420
1795 012402 023420
1796 012404 001750
1797 012406 000144
1798 012410 000012
1799 012412 000001
1800 012414 000176 000004
1801
1802 012420 005201
1803 012422 112711 000060
1804 012426 020015
1805 012430 002403
1806 012432 105211
1807 012434 161500
1808 012436 000773
1809 012440 022527 000001
1810 012444 001365
1811 012446 000205
1812
1813 012450 016501 177776
1814 012454 012700 000004
1815 012460 126127 000001 000060
1816 012466 001006
1817 012470 111161 000001
1818 012474 112721 000040
1819 012500 005300
1820 012502 001366
1821 012504
1822 (2) 012504 012601
1823 012506 012600
1824 012510 005726
1825 012512 000205
1826 012514 030060
1827 012516 030060
1828 012520 030060

:
: CONVERT BINARY TO SIGNED SIX DIGIT ASCII DECIMAL STRING.
: "DECIMAL" -- NULL LEADING ZEROS (WITH SPACES).
: "DECMLZ" -- RETAIN LEADING ZEROS.
: CALL: JSR R5,DECIMAL (OR DECMLZ)
: VALUE ADDRESS
: STRING ADDRESS
:
DECIMAL: MOV #NLZ,-(SP) ; SET TO NULL LEAD ZEROS...
          BR .+6
DECMLZ: MOV #DECXIT,-(SP) ; ...OR NOT.
          MOV R0,-(SP) ;:PUSH R0 ON STACK
          MOV R1,-(SP) ;:PUSH R1 ON STACK
          MOV @R5+,R0 ; VALUE => R0.
          MOV (R5)+,R1 ; DST ADDRESS => R1.
          MOV #',(R1) ; 1ST DIGIT IS SIGN, ASSUME (+).
          TST R0 ; IS IT ???
          BPL 1$ ; YUP.
          NEG R0 ; NOPE, MAKE ABS VALUE...
          MOV #'-,(R1) ; ...AND CHANGE THE SIGN TO (-).
1$: JSR R5,2$ ; CONVERT BINARY TO ASCII
    10000. ; 10^4
    1000. ; 10^3
    100. ; 10^2
    10. ; 10^1
    1. ; 10^0
    JMP @4(SP) ; GOTO 'NLZ' OR 'DECXIT'.

2$: INC R1 ; POINT TO NEXT DIGIT...
    MOV #'0,(R1) ; ...AND SET IT TO 0.
3$: CMP R0,(R5)
    BLT 4$
    INCB (R1) ; ACCUMULATE THIS DIGIT.
    SUB (R5),R0
    BR 3$
4$: CMP (R5)+,#1 ; BUMP INDEX POINTER...
    BNE 2$ ; ...LOOP FOR 5 DIGITS...
    RTS ; ...AND RETURN.

NLZ: MOV -2(R5),R1 ; RECALL ORIGINAL POINTER...
      MOV #4,R0 ; ...AND NULL UP TO 4 LEAD 0'S.
1$: CMPB 1(R1),#'0
    BNE DECXIT ; EXIT ON 1ST NON-ZERO DIGIT.
    MOVB (R1),1(R1) ; MOVE SIGN DOWN 1...
    MOVB #',(R1)+ ; ...AND INSERT NULL (SPACE).
    DEC R0
    BNE 1$ ; LOOP 'TIL DONE.

DECXIT: MOV (SP)+,R1 ;:POP STACK INTO R1
        MOV (SP)+,R0 ;:POP STACK INTO R0
        TST (SP)+
        RTS R5

HI2: .ASCII /00/ ; TEMP ASCII BUFFER.
MID2: .ASCII /00/
LO2: .ASCII /00/

```



```
1830 :*****
(3) :*TEST 32 Z WRITING TABLET EXERCISER FRAME (MANUAL)
(3) :*****
(2) 012522 000004 TST32: SCOPE
1831 :
1832 : DISPLAY A SET OF NESTED BOXES AND A TRACKING BALL ON SCREEN.
1833 : SLAVE THE BALL TO THE TABLET CO-ORDS, AND DISPLAY TAB X/Y.
1834 : ON ANY POINT CORRELATOR HIT, BUMP THE HIT COUNT READOUT.
1835 : IF THE TABLET PEN IS 'DOWN', AND WE'RE NOT IN THE MENU AREA,
1836 : MARK THE HIT POINT (X) AND UPDATE THE HIT X/Y READOUT.
1837 : ALTER APPERTURE SIZE AS BEFORE. TYPE <^C> TO EXIT.
1838 :
1839 164000 PENUP= DNOP ; SWITCHES FOR PEN STATUS...
1840 161003 PENDN= DSKP3
1841 164000 PRXIN= DNOP ; ...AND PROXIMITY STATUS IN MENU.
1842 161002 PRXOUT= DSKP2
1843 :
1844 012524 012746 000000 MOV #PRO,-(SP) ;;PUT NEW PS ON STACK
(1) 012530 012746 012536 MOV #64$,-(SP) ;;PUT NEW PC ON STACK
(1) 012534 000002 RTI ;;POP NEW PC AND PS
(1) 012536
1845 012536 013700 000004 64$: MOV @#ERRVEC,RO ; FIRST, LETS SEE IF THERE'S...
1846 012542 012737 012556 000004 MOV #1$,@#ERRVEC ;...A TABLET INSTALLED.
1847 012550 005777 166562 TST @TABLT
1848 012554 000405 BR T32 ; OK, GO START 'EM UP.
1849 012556 010037 000004 1$: MOV RO,@#ERRVEC ; NO TABLET, RESTORE VECTOR...
1850 012562 012716 001344 MOV #BEGIN,(SP) ;...FIX RETURN PC.
1851 012566 000002 RTI
1852 :
1853 012570 010037 000004 T32: MOV RO,@#ERRVEC ; RESTORE ERROR VECTOR.
1854 012574 012777 012012 166456 MOV #PCKBD,@TKBVCT ; SET KBD VECTOR...
1855 012602 012777 000200 166452 MOV #PR4,@TKBV11
1856 012610 052777 000100 166326 BIS #BIT6,@$TKS ;...AND TURN IT ON.
1857 012616 105037 013252 CLR B TSYNC ; CLEAR TABLET SYNC FLAG...
1858 012622 012737 013252 013264 MOV #TSYNC,TBUFR ;...AND SET BUFFER POINTER.
1859 012630 012777 013302 166502 MOV #TBSRVC,@TBVCT ; SET TABLET VECTOR...
1860 012636 012777 000200 166476 MOV #PR4,@TBVCT1
1861 012644 052777 000100 166464 BIS #BIT6,@TABLT ;...AND TURN IT ON.
1862 012652 012777 013124 166442 MOV #10$,@LPVCT ; SET LP HIT VECTOR.
1863 012660 012777 000200 166436 MOV #PR4,@LPVCT1
1864 012666 012777 013244 166422 MOV #11$,@DDONE ; SET STOP VECTOR (JUST RESUME).
1865 012674 012737 164000 035544 MOV #DNOP,HITS ; ENABLE HIT COUNTER...
1866 012702 012737 164000 035442 MOV #DNOP,APPERT ;...AND APPERTURE SIZE IN MENU.
1867 012710 012737 177777 012144 MOV #-1,HITK
1868 012716 004737 012142 JSR PC,HITUP ; CLEAR HIT COUNTER.
1869 012722 012777 035720 166334 MOV #PIX3,@DPC ; START THE DISPLAY....
1870 :...AND FALL THRU.
```



```

1872
1873      ; NOW UPDATE EVERYTHING ON THE FLY.
1874
1875 012730 005737 013300      1$:  TST      TBRDY      ; TABLET DATA READY ??
1876 012734 001461              BEQ      5$          ; NO, SKIP IT.
1877 012736 004737 013430      JSR      PC,GETXY    ; YES, GET TABLET X,Y => R2,R3
1878 012742 023727 036364 164000  CMP      PRX.SW,#PRXIN ; ARE WE IN PROXIMITY ??
1879 012750 001053              BNE      5$          ; NO, DATA'S INVALID, SKIP IT.
1880
1881 012752 020327 001777              CMP      R3,#1777    ; YES, TRACK THE BALL.
1882 012756 003402              BLE      2$          ;
1883 012760 012703 001777              MOV      #1777,R3    ; LIMIT Y TO 1777.
1884 012764 020227 002000      2$:  CMP      R2,#2000  ; IS X IN THE MENU ??
1885 012770 002007              BGE      3$          ;
1886 012772 012737 164000 035722  MOV      #DNOP,PIX3+2 ; NO, TRACK IN MAJOR AREA.
1887 013000 012737 161002 036174  MOV      #DSKP2,TBMENU+2
1888 013006 000420              BR       4$          ;
1889 013010 012737 164000 036174  3$:  MOV      #DNOP,TBMENU+2 ; YES, TRACK IN MINOR AREA.
1890 013016 012737 161002 035722  MOV      #DSKP2,PIX3+2
1891 013024 162702 002000      SUB      #2000,R2   ; ADJUST X FOR MENU FIELD...
1892 013030 010200              MOV      R2,R0
1893 013032 006302              ASL      R2
1894 013034 060002              ADD      R0,R2      ;...SCALE X3.
1895 013036 020227 000250              CMP      R2,#250
1896 013042 003402              BLE      4$          ;
1897 013044 012702 000250              MOV      #250,R2    ;*** LIMIT MENU X TO 250 ***
1898 013050 010237 036054      4$:  MOV      R2,TBX      ; SET X CO-ORD...
1899 013054 010337 036056              MOV      R3,TBX+2   ;...AND Y CO-ORD...
1900 013060 004537 012200              JSR      R5,OCTAL   ;...AND UPDATE THE READOUT.
1901 013064 036054 036230              TBX, TAB.X
1902 013070 004537 012200              JSR      R5,OCTAL
1903 013074 036056 036246              TBX+2, TAB.Y
1904
1905 013100 123737 012136 035462  5$:  CMPB     ASIZE,A.SIZ ; APPERTURE SIZE CHANGED ??
1906 013106 001710              BEQ      1$          ; NO, GO 'ROUND.
1907 013110 004737 011706              JSR      PC,SETAPP  ; YES, UPDATE IT...
1908 013114 013737 012000 036050  MOV      APPERX,TBALL
1909 013122 000702              BR       1$          ;...AND GO 'ROUND.

```



```

1911
1912           ; COME HERE ON ANY PC HIT.
1913
1914 013124 023727 036364 164000 10$:  CMP    PRX.SW,#PRXIN  ; ARE WE IN PROXIMITY ??
1915 013132 001044                BNE    11$          ; IF NOT, JUST RESUME AND EXIT.
1916
1917 013134 017737 166130 014464      MOV    @XPOS,DSAVE  ; OTHERWISE, SAVE VTX...
1918 013142 042737 176000 014464      BIC    #^C1777,DSAVE
1919 013150 017737 166116 014466      MOV    @YPOS,DSAVE1 ; ...AND VTY...
1920 013156 042737 176000 014466      BIC    #^C1777,DSAVE1
1921 013164 004737 012142                JSR    PC,HITUP    ; ...AND UPDATE HIT COUNT.
1922
1923 013170 032777 000100 166100      BIT    #BIT6,@DSR1
1924 013176 001022                BNE    11$          ; RESUME AND EXIT IF MENU ON...
1925 013200 023727 036332 161003      CMP    PEN.SW,#PENDN
1926 013206 001016                BNE    11$          ; ...OR PEN SWITCH NOT DOWN.
1927
1928 013210 004537 012200                JSR    R5,OCTAL    ; OTHERWISE, UPDATE HIT X...
1929 013214 014464 036264      DSAVE, HIT.X
1930 013220 004537 012200                JSR    R5,OCTAL    ; ...AND HIT Y...
1931 013224 014466 036302      DSAVE1, HIT.Y
1932 013230 013737 014464 036020      MOV    DSAVE,PIX3X ; ...AND MARK THE SPOT.
1933 013236 013737 014466 036022      MOV    DSAVE1,PIX3X+2
1934
1935 013244 005277 166014                11$:  INC    @DPC        ; RESUME THE DISPLAY...
1936 013250 000002                RTI                ; ...AND DISMISS.
  
```


1938
 1939
 1940
 1941
 1942
 1943
 1944 013252 000
 1945 013253 061
 1946 013254 060
 1947 013255 065
 1948 013256 061
 1949 013257 062
 1950 013260 060
 1951 013261 065
 1952 013262 061
 1953 013263 062
 1954
 1955 013264 013252
 1956 013266 000000
 1957 013270 000000 000000
 1958 013274 000000 000000
 1959 013300 000000
 1960
 1961 013302
 (2) 013302 010046
 1962 013304 010146
 1963 013306 013701 013264
 1964 013312 013700 001336
 1965 013316 016000 000002
 1966 013322 042700 177600
 1967 013326 105737 013252
 1968 013332 001005
 1969 013334 005037 013300
 1970 013340 020027 000015
 1971 013344 001026
 1972
 1973 013346 020027 000015
 1974 013352 001020
 1975 013354 012701 013253
 1976 013360 112137 013266
 1977 013364 012137 013270
 1978 013370 012137 013272
 1979 013374 012137 013274
 1980 013400 012137 013276
 1981 013404 005237 013300
 1982 013410 012701 013252
 1983 013414 110021
 1984 013416 010137 013264
 1985 013422
 (2) 013422 012601
 1986 013424 012600
 1987 013426 000002

TABLET INTERRUPT HANDLER.
 ACCOMODATE EITHER THE OLD OR THE NEW DATA FORMATS.

			OLD	NEW
			---	---
TSYNC:	.BYTE	0	<15>	<15>
	.BYTE	'1	ASCII PEN UP/DN	PEN<0>, AND LO 2 BITS OF X/Y
	.BYTE	'0	ASCII DECML X	X<13:8> IN <5:0>
	.BYTE	'5	"	X<7:2> IN <5:0>
	.BYTE	'1	"	Y<13:8> IN <5:0>
	.BYTE	'2	"	Y<7:2> IN <5:0>
	.BYTE	'0	ASCII DECML Y	UNUSED
	.BYTE	'5	"	"
	.BYTE	'1	"	"
	.BYTE	'2	"	"

TBUFR:	TSYNC	:	BUFFER POINTER.	
TBSTAT:	0	:	STATUS	STATUS
TABX:	0,0	:	ASCII X	OCTAL X AND Y
TABY:	0,0	:	ASCII Y	UNUSED
TBRDY:	0	:	DATA READY FLAG.	

TBSRVC:

```

MOV R0,-(SP)      ;;PUSH R0 ON STACK
MOV R1,-(SP)      ;;PUSH R1 ON STACK
MOV TBUFR,R1      ; POINTER => R1.
MOV TABLT,R0
MOV 2(R0),R0      ; CHAR => R0.
BIC #^C177,R0     ; STRIP OFF CRAP.
TSTB TSYNC
BNE 1$           ; BR IF WE'RE SYNCED-UP.
CLR TBRDY        ; OTHERWISE, CLEAR READY...
CMP R0,#15      ;...AND...
BNE 4$           ;...DISMISS UNTIL 1ST <CR>.

1$:  CMP R0,#15
     BNE 3$
2$:  MOV #TSYNC+1,R1 ; ON <CR>, DOUBLE BUFFER...
     MOVB (R1)+,TBSTAT ;...STATUS...
     MOV (R1)+,TABX
     MOV (R1)+,TABX+2 ;...X CO-ORD...
     MOV (R1)+,TABY
     MOV (R1)+,TABY+2 ;...AND Y CO-ORD...
     INC TBRDY        ;...AND SET NEW DATA READY FLAG.
     MOV #TSYNC,R1   ; NOW, RESET BUFFER POINTER.
3$:  MOVB R0,(R1)+   ; STORE DATA CHAR...
     MOV R1,TBUFR    ;...AND SAVE NEW POINTER.

4$:  MOV (SP)+,R1     ;;POP STACK INTO R1
     MOV (SP)+,R0     ;;POP STACK INTO R0
     RTI              ;...AND EXIT.
  
```



```

1989
1990
1991
1992
1993
1994 013430 005037 013300
1995 013434 123727 013266 000102
1996 013442 001004
1997 013444 012737 161002 036364
1998 013452 000207
1999
2000 013454 012737 164000 036364 1$:
2001 013462 012700 161003
2002 013466 033727 013266 000001
2003 013474 001402
2004 013476 012700 164000
2005 013502 010037 036332 2$:
2006 013506 013703 013270
2007 013512 004737 013524
2008 013516 010302
2009 013520 013703 013272
2010 013524 000303 3$:
2011 013526 042703 140300
2012 013532 106303
2013 013534 106303
2014 013536 006203
2015 013540 006203
2016 013542 000207

:
: SUBROUTINE TO PACK TABLET DATA INTO GRAPHICS CO-ORDS.
: SET PEN AND PROXIMITY INDICATORS.
: RETURN X IN R2, AND Y IN R3 IF GOOD DATA (IN PROXIMITY).
:
GETXY: CLR TBRDY ; CLEAR READY FLAG.
NEWFM: CMPB TBSTAT,#102 ; OUT OF PROXIMITY ??
BNE 1$ ; NO.
MOV #PRXOUT,PRX.SW ; SET 'OUT-OF-PROXIMITY'...
RTS PC ;...AND RETURN.

1$: MOV #PRXIN,PRX.SW ; SET 'IN PROXIMITY'.
MOV #PENDN,R0 ; SET 'PEN DOWN'...
BIT TBSTAT,#1 ;...IF BIT 0 = 0...
BEQ 2$
MOV #PENUP,R0 ;...OTHERWISE, 'PEN UP'.
2$: MOV R0,PEN.SW
MOV TABX,R3 ; GET RAW X DATA.
JSR PC,3$
MOV R3,R2 ; PACKED X => R2.
MOV TABX+2,R3 ; GET RAW Y.
3$: SWAB R3 ; REVERSE THE BYTES, AND...
BIC #140300,R3 ;...DISCARD HI 2 BITS OF EACH.
ASLB R3
ASLB R3 ; REPOSITION LO BYTE TO <7:2>.
ASR R3
ASR R3 ; REPOSITION WORD TO <11:0>.
RTS PC

```



```
2018  
2019  
2020  
2021  
2022  
2023  
2024 013544 023727 013270 031066  
2025 013552 001404  
2026 013554 023727 013274 031066  
2027 013562 001004  
2028 013564 012737 161002 036364  
2029 013572 000207  
2030  
2031 013574 012737 164000 036364  
2032 013602 012700 161003  
2033 013606 123737 013266 030447  
2034 013614 001402  
2035 013616 012700 164000  
2036 013622 010037 036332  
2037 013626 012700 013270  
2038 013632 004737 013640  
2039 013636 010302  
2040 013640 005003  
2041 013642 012746 000004  
2042 013646 112001  
2043 013650 042701 177600  
2044 013654 162701 000060  
2045 013660 006303  
2046 013662 010346  
2047 013664 006303  
2048 013666 006303  
2049 013670 062603  
2050 013672 060103  
2051 013674 042703 170000  
2052 013700 005316  
2053 013702 001361  
2054 013704 006203  
2055 013706 005726  
2056 013710 000207
```

```
;  
; SUBROUTINE TO CONVERT TABLET DATA TO OCTAL GRAPHICS CO-ORDS.  
; RETURN X IN R2, AND Y IN R3 IF GOOD DATA (PROXIMITY OK).  
; *** OLD STYLE ASCII DECIMAL DATA FORMAT -- NO LONGER USED ***  
; *** RETAINED HERE FOR HISTORICAL REASONS ONLY *****  
;  
OLDFMT: CMP TABX,#'62 ; X IN PROXIMITY ??  
BEQ 1$ ; NO.  
CMP TABY,#'62 ; YES, HOW 'BOUT Y ??  
BNE 2$ ; BR IF BOTH OK.  
1$: MOV #PRXOUT,PRX.SW ; SET 'OUT-OF-PROXIMITY'...  
RTS PC ;...AND RETURN.  
  
2$: MOV #PRXIN,PRX.SW ; SET 'IN PROXIMITY'.  
MOV #PENDN,R0 ; SET 'PEN DOWN'...  
CMPB TBSTAT,'1'  
BEQ 3$  
MOV #PENUP,R0 ;...OR 'PEN UP'.  
3$: MOV R0,PEN.SW  
MOV #TABX,R0 ; CONVERT DECIMAL TABLET X/Y...  
JSR PC,4$ ;...TO OCTAL IN R3.  
MOV R3,R2 ; X CO-ORD => R2 AND FALL THRU.  
4$: CLR R3 ; ACCUMULATE IN R3.  
MOV #4,-(SP) ; SET A LOOP COUNT.  
5$: MOVB (R0)+,R1 ; 1ST/NEXT DIGIT => R1.  
BIC #^C177,R1  
SUB #'0,R1 ; STRIP ASCII.  
ASL R3 ; ACC X 2...  
MOV R3,-(SP) ;...AND SAVE FOR A MOMENT.  
ASL R3 ; X 4  
ASL R3 ; X 8  
ADD (SP)+,R3 ; + (X2) = X 10  
ADD R1,R3 ; ADD IN THIS DIGIT.  
BIC #^C7777,R3 ; CAN'T EXCEED 12 BITS.  
DEC (SP)  
BNE 5$ ; LOOP FOR 4 DIGITS.  
ASR R3 ; *** ADJUST FINAL CO-ORD ***  
TST (SP)+  
RTS PC ; FIX STACK
```



```

2058 .SBTTL END OF PASS ROUTINE
(1)
(2)
(1)
(1)
(1)
(1)
(1)
(1) 013712
(1) 013712 000004
(1) 013714 005037 001102
(1) 013720 005237 001200
(1) 013724 042737 100000 001200
(1) 013732 005327
(1) 013734 000001
(1) 013736 003013
(1) 013740 012737
(1) 013742 000001
(1) 013744 013734
(1) 013746 013700 000042
(1) 013752 001405
(1) 013754 000005
(1) 013756 004710
(1) 013760 000240
(1) 013762 000240
(1) 013764 000240
(1) 013766
(1) 013766 000137
(1) 013770 001526

$EOP:
CLR $STNM ;;ZERO THE TEST NUMBER
INC $PASS ;;INCREMENT THE PASS NUMBER
BIC #100000,$PASS ;;DON'T ALLOW A NEG. NUMBER
DEC (PC)+ ;;LOOP?
$EOPCT: .WORD 1
BGT $DOAGN ;;YES
MOV (PC)+,@(PC)+ ;;RESTORE COUNTER
$ENDCT: .WORD 1
$GET42: MOV @#42,R0 ;;GET MONITOR ADDRESS
BEQ $DOAGN ;;BRANCH IF NO MONITOR
RESET ;;CLEAR THE WORLD
$ENDAD: JSR PC,(R0) ;;GO TO MONITOR
NOP ;;SAVE ROOM
NOP ;;FOR
NOP ;;ACT11
$DOAGN:
JMP @(PC)+ ;;RETURN
$RTNAD: .WORD RESTAT

2059 .SBTTL SUBROUTINE FOR VERT. LIGHT PEN FIELD OF VIEW
2060
2061 013772 012701 000030 LOADAC: MOV #24.,R1 ;LOAD COUNT
2062 013776 012720 130000 MOV #RELATP,(R0)+ ;LOAD RELATIVE POINT
2063 014002 012720 040004 1$: MOV #INTX+4,(R0)+ ;LOAD INTEN BIT
2064 014006 005301 DEC R1 ;FINISHED ?
2065 014010 001374 BNE 1$ ;BR IF NOT
2066 014012 000207 RTS PC ;EXIT

2067
2068 .SBTTL SUBROUTINE FOR HORIZ. LIGHT PEN FIELD OF VIEW
2069 014014 012737 000030 014500 LOADUP: MOV #24.,CNTR ;LOAD COUNT
2070 014022 004737 013772 1$: JSR PC,LOADAC ;LOAD ACCROSS
2071 014026 012720 110000 MOV #LONGV,(R0)+ ;LOAD LONG VECTOR
2072 014032 012720 000004 MOV #4,(R0)+ ;LOAD VECTOR OVER
2073 014036 012720 020140 MOV #MINUSX+140,(R0)+ ;AND UP
2074 014042 005337 014500 DEC CNTR
2075 014046 001365 BNE 1$ ;BR IF NOT DONE
2076 014050 000207 RTS PC ;EXIT

2077
2078 .SBTTL LIGHT-PEN INTERRUPT SERVICE
2079 014052 017737 165230 001126 RET14: MOV @VSCONS,$BDDAT ;READ CONSOLE STATUS REG
2080 014060 017737 165204 014470 MOV @XPOS,DSAVE2 ;READ X POSITION
2081 014066 017737 165200 014472 MOV @YPOS,DSAVE3 ;READ Y POSITION
2082 014074 042737 176000 014470 BIC #176000,DSAVE2 ;MASK HIGH SIX BITS
2083 014102 042737 176000 014472 BIC #176000,DSAVE3
2084 014110 005037 014456 CLR 40$ ;CLEAR SWITCH FLAG HAPPEN LOC.
2085 014114 032737 000100 001126 BIT #BIT6,$BDDAT ;TEST IF CONSOLE #1 SWITCH FLAG
2086 014122 001405 BEQ 3$ ;BR IF NOT
    
```

2087	014124	012737	027560	027440		MOV	#PENOF1,MS1PEN		:INFORM PEN OF ON #1 SET
2088	014132	005237	014456			INC	40\$:SET SW HAPPENED FLAG
2089	014136	032737	000200	001126	3\$:	BIT	#BIT7,\$BDDAT		:TEST IF CONSOLE #1 SWITCH FLAG
2090	014144	001405				BEQ	4\$:BR IF NOT
2091	014146	012737	027610	027440		MOV	#PENON1,MS1PEN		:INFORM PEN ON #1 SET
2092	014154	005237	014456			INC	40\$:SET SW HAPPENED FLAG
2093	014160	032737	010000	001126	4\$:	BIT	#BIT12,\$BDDAT		:TEST IF CONSOLE #0 SWITCH FLAG
2094	014166	001405				BEQ	5\$:BR IF NOT
2095	014170	012737	027500	027370		MOV	#PENOFF0,MSOPEN		:INFORM PEN OFF #0 SET
2096	014176	005237	014456			INC	40\$:SET SW HAPPENED FLAG
2097	014202	032737	020000	001126	5\$:	BIT	#BIT13,\$BDDAT		:TEST IF CONSOLE #0 SET
2098	014210	001405				BEQ	6\$:BR IF NOT
2099	014212	012737	027530	027370		MOV	#PENON0,MSOPEN		:INFORM PEN ON #0 SET
2100	014220	005237	014456			INC	40\$:SET SW HAPPENED FLAG
2101	014224	005737	014456		6\$:	TST	40\$:TEST IF SWITCH FUNCTION
2102	014230	001003				BNE	12\$:BR IF YES
2103	014232	005737	014460			TST	VIEW		:TEST IF FIELD OF VIEW
2104	014236	001043				BNE	20\$:BR IF YES
2105	014240	032737	040000	001126	12\$:	BIT	#BIT14,\$BDDAT		:TEST IF PEN FLAG #0 SET
2106	014246	001414				BEQ	7\$:BR IF NOT
2107	014250	013703	014470			MOV	DSAVE2,R3		:LOAD R3
2108	014254	012702	027344			MOV	#DLT14A+4,R2		:LOAD ADDRESS
2109	014260	004737	014636			JSR	PC,KBCHR		:LOAD X READOUT
2110	014264	013703	014472			MOV	DSAVE3,R3		:LOAD R3
2111	014270	012702	027356			MOV	#DLT14B+4,R2		:LOAD ADDRESS
2112	014274	004737	014636			JSR	PC,KBCHR		:LOAD Y READOUT
2113	014300	032737	000400	001126	7\$:	BIT	#BIT8,\$BDDAT		:TEST IF PEN #1 FLAG
2114	014306	001414				BEQ	10\$:BR IF NOT
2115	014310	013703	014470			MOV	DSAVE2,R3		:GET X VALUE
2116	014314	012702	027414			MOV	#DLT14C+4,R2		:LOAD POINTER
2117	014320	004737	014636			JSR	PC,KBCHR		:CONVERT TO ASCII
2118	014324	013703	014472			MOV	DSAVE3,R3		:GET Y VALUE
2119	014330	012702	027426			MOV	#DLT14D+4,R2		:LOAD POINTER
2120	014334	004737	014636			JSR	PC,KBCHR		:CONVERT TO ASCII
2121	014340	012716	014540		10\$:	MOV	#RESUME,(SP)		:FIX RETURN PC...
2122	014344	000002				RTI			:...AND RESUME.
2123									
2124									:COME HERE IF LIGHT-PEN HIT DURING THE FIELD OF VIEW SUB-PICTURE
2125									
2126	014346	005237	014462		20\$:	INC	HITCNT		:UPDATE COUNT
2127	014352	013703	014462			MOV	HITCNT,R3		:LOAD COUNT #
2128	014356	012702	027474			MOV	#FRM16B,R2		:LOAD MESSAGE POINTER
2129	014362	004737	014636			JSR	PC,KBCHR		:CONVERT TO ASCII
2130	014366	005001				CLR	R1		
2131	014370	005002				CLR	R2		
2132	014372	013700	014470			MOV	DSAVE2,R0		:GET X AXIS
2133	014376	162700	001400			SUB	#1400,R0		:GET A BASE ADDRESS
2134	014402	006200				ASR	R0		
2135	014404	006200				ASR	R0		
2136	014406	001404				BEQ	30\$		
2137	014410	062701	000070		21\$:	ADD	#70,R1		:UPDATE OFFSET
2138	014414	005300				DEC	R0		
2139	014416	001374				BNE	21\$:BR UNTIL DONE
2140	014420	013700	014472		30\$:	MOV	DSAVE3,R0		:GET Y AXIS
2141	014424	162700	000304			SUB	#304,R0		:MAKE BASE ADDRESS
2142	014430	006200				ASR	R0		

2143	014432	006200				ASR	RO		;SHIFT RIGHT
2144	014434	001404				BEQ	32\$		
2145	014436	062701	000002		31\$:	ADD	#2,R1		
2146	014442	005300				DEC	RO		
2147	014444	001374				BNE	31\$		
2148	014446	042761	040000	036666	32\$:	BIC	#INTX,BUFFER+10(R1)		;CLEAR THE BIT
2149	014454	000731				BR	10\$		
2150									
2151	014456	000000			40\$:		0		
2152									
2153	014460					SHIFTO:			
2154	014460					DIDINT:			
2155	014460	000000				VIEW:	0		
2156	014462	000000				HITCNT:	0		
2157	014464	000000				DSAVE:	0		
2158	014466	000000				DSAVE1:	0		
2159	014470	000000				DSAVE2:	0		
2160	014472	000000				DSAVE3:	0		
2161	014474	000000				HOLD:	0		
2162	014476	000000				TSAVE:	0		
2163	014500	000000				CNTR:	0		
2164	014502	000000				TEMPA:	0		
2165	014504	000000				TEMPB:	0		

```

2167          .SBTTL DISPLAY SUB-ROUTINE
2168          :
2169          : ARG5 ARE LOOP COUNT AND DISPLAY BUFFER ADDRESS
2170          : UPON INTERRUPT , DEC LOOP COUNT
2171          : RESUME DISPLAY IF NOT 0
2172          : RTS R5 IF COMPLETED
2173          :
2174          014506 012537 014630 DSPLA: MOV (R5)+,COUNT ; ITERATION COUNT.
2175          014512 012537 014632          MOV (R5)+,FILE ; DISPLAY FILE ADDRESS.
2176          014516 012746 000000          MOV #PRO,-(SP) ; PUT NEW PS ON STACK
2177          (1) 014522 012746 014530          MOV #64$,-(SP) ; PUT NEW PC ON STACK
2178          (1) 014526 000002          RTI ; POP NEW PC AND PS
2179          (1) 014530
2180          014530 013777 014632 164526 64$: MOV FILE,@DPC ; START DISPLAY
2181          014536 000403          BR .+10
2182          014540 012777 000001 164516 RESUME: MOV #1,@DPC ; RESUME DISPLAY.
2183          014546 000240 000240 1$: 240, 240
2184          014552 000001          WAIT
2185          014554 000774          BR 1$ ; LOOP BACK
2186          :
2187          : RETURN HERE UPON STOP INTERRUPT
2188          :
2189          014556 005337 014630 STOPI: DEC COUNT ; FINISHED LOOPING ?
2190          014562 001403          BEQ 1$ ; BR IF SO.
2191          014564 012716 014540          MOV #RESUME,(SP) ; RESUME...
2192          014570 000002          RTI ; ...OTHERWISE.
2193          :
2194          014572 105777 164342 1$: TSTB @SWR ; KEYBOARD CONTROL ??
2195          014576 100410          BMI 2$ ; EXIT IF SO.
2196          014600 005037 014634          CLR SWITCH
2197          014604 032777 001000 164326 BIT #BIT9,@SWR ; TEST SWITCH BIT 9
2198          014612 001402          BEQ 2$
2199          014614 005137 014634          COM SWITCH ; SET FLAG IF SWR 9 = 1
2200          014620 012716 014626 2$: MOV #3$,(SP) ; FIX RETURN PC...
2201          014624 000002          RTI ; ...AND...
2202          014626 000205          3$: RTS R5 ; ...RETURN TO CALLER.
2203          :
2204          COUNT: 0
2205          FILE: 0
2206          SWITCH: 0
    
```



```

2205 .SBTTL UPDATE OCTAL READOUT OF THE X-Y FOR LIGHT PEN HIT
2206
2207 : CALL: MOV VAL,R3 ; VALUE TO ENCODE.
2208 : MOV ADDR,R2 ; ADDRESS OF LO ORDER CHAR +1.
2209 : JSR PC,KBCHR ; ENCODE 4 DIGIT OCTAL ASCII.
2210
2211 KBCHR: JSR PC,10$ ;LOAD BITS
2212 : MOVB R4,-(R2) ;SAVE BITS
2213 : JSR PC,11$ ;MOVE BITS
2214 : MOVB R4,-(R2) ;SAVE BITS
2215 : JSR PC,11$ ;MOVE BITS
2216 : MOVB R4,-(R2) ;SAVE BITS
2217 : JSR PC,11$
2218 : MOVB R4,-(R2)
2219 : RTS PC
2220 11$: ROR R3
2221 : ROR R3
2222 : ROR R3
2223 10$: MOV R3,R4 ;LOAD R4
2224 : BIC #177770,R4 ;MASK BITS
2225 : ADD #60,R4 ;MAKE A NUMBER
2226 : RTS PC
2227
2228 DELTX6: C
2229 DELTY6: C
2230
  
```

```
2232 .SBTTL X - Y POSITIONS FOR THE SHORT TERM DRIFT TEST
2233
2234 014716 000000 000000 000000 TABA: .WORD 0,0,0,0,0,0
2235 014732 001000 001000 .WORD 1000,1000
2236 014736 000000 001777 000000 .WORD 0,MAXY,0,MAXY,0,MAXY
2237 014752 001000 001000 .WORD 1000,1000
2238 014756 001777 001777 001777 .WORD MAXX,MAXY,MAXX,MAXY,MAXX,MAXY
2239 014772 001000 001000 .WORD 1000,1000
2240 014776 001777 000000 001777 .WORD MAXX,0,MAXX,0,MAXX,0
2241 015012 001000 001000 .WORD 1000,1000
2242 015016 100000 BIT15
2261
2262
```


2274	015020	117000			FRME0: POINT!INT4	
2275	015022	000000			0	
2276	015024	001600			MAXY-177	
2277	015026	164300			CONSLO!BIT7!BIT6	:ENABLE CONSOLE #0
2278	015030	164700			CONSL1!BIT7!BIT6	:ENABLE CONSOLE #1
2279	015032	100000			CHAR	
2280	015034	051526	030066	053040	.ASCII /VS60 VISUAL DISPLAY TEST < CZVSFA >/	
2281	015100	015	012	012	.BYTE 15,12,12	
2282	015103	040	020040	044504	.ASCII / DIRECTORY OF THE TESTS/	
2283	015134	015	012	012	.BYTE 15,12,12	
2284	015137	101	036440	030040	.ASCII /A = 01 = DIRECTORY FRAME/	
2285	015167	015	012		.BYTE 15,12	
2286	015171	102	036440	030040	.ASCII /B = 02 = ASTIGMATISM AND SETTLING/	
2287	015232	015	012		.BYTE 15,12	
2288	015234	020103	020075	031460	.ASCII /C = 03 = SHORT TERM DRIFT/	
2289	015265	015	012		.BYTE 15,12	
2290	015267	104	036440	030040	.ASCII /D = 04 = MINOR AXIS GAIN/	
2291	015317	015	012		.BYTE 15,12	
2292	015321	105	036440	030040	.ASCII /E = 05 = MAJOR AXIS OFFSET/	
2293	015353	015	012		.BYTE 15,12	
2294	015355	106	036440	030040	.ASCII /F = 06 = VECTOR LENGTH GAIN/	
2295	015410	015	012		.BYTE 15,12	
2296	015412	020107	020075	033460	.ASCII /G = 07 = PINCUSHION/	
2297	015435	015	012		.BYTE 15,12	
2298	015437	110	036440	030440	.ASCII /H = 10 = OCTAGONS AND CIRCLES/	
2299	015474	015	012		.BYTE 15,12	
2300	015476	020111	020075	030461	.ASCII /I = 11 = SCISSORING AND VECTOR SCALES/	
2301	015543	015	012		.BYTE 15,12	
2302	015545	112	036440	030440	.ASCII /J = 12 = X AND Y DYNAMIC OFFSET TEST/	
2303	015611	015	012		.BYTE 15,12	
2304	015613	113	036440	030440	.ASCII /K = 13 = CHARACTER SCALE/	
2305	015643	015	012		.BYTE 15,12	
2306	015645	114	036440	030440	.ASCII /L = 14 = CHARACER QUALITY AND CHARACTER ROTATE/	
2307	015723	015	012		.BYTE 15,12	
2308	015725	115	036440	030440	.ASCII /M = 15 = CHARACTER SET, SUPER AND SUBSCRIPT, AND ITALIC CHARACTERS/	
2309	016027	015	012		.BYTE 15,12	
2310	016031	116	036440	030440	.ASCII /N = 16 = SYNC SPEED AND CHARACTER TERMINATE/	
2311	016104	015	012		.BYTE 15,12	
2312	016106	020117	020075	033461	.ASCII /O = 17 = DASH LINES AND BLINK/	
2313	016143	015	012		.BYTE 15,12	
2314	016145	120	036440	031040	.ASCII /P = 20 = VECTOR LENGTH/	
2315	016173	015	012		.BYTE 15,12	
2316	016175	121	036440	031040	.ASCII /Q = 21 = HORIZONTAL PHOSPHOR TEST/	
2317	016236	015	012		.BYTE 15,12	
2318	016240	020122	020075	031062	.ASCII /R = 22 = VERTICAL PHOSPHOR TEST/	
2319	016277	015	012		.BYTE 15,12	
2320	016301	123	036440	031040	.ASCII /S = 23 = SHORT VECTORS AND RELATIVE POINT/	
2321	016352	015	012		.BYTE 15,12	
2322	016354	020124	020075	032062	.ASCII /T = 24 = GRAPHPLOT X AND GRAPHPLOT Y TEST/	
2323	016425	015	012		.BYTE 15,12	
2324	016427	125	036440	031040	.ASCII /U = 25 = INTENSITY LEVEL AND LIGHT PEN TESTS/	
2325	016503	015	012		.BYTE 15,12	
2326	016505	126	036440	031040	.ASCII /V = 26 = KEYBOARD CHARACTER ECHO LOOP/	
2327	016552	015	012		.BYTE 15,12	
2328	016554	020127	020075	033462	.ASCII /W = 27 = DYNAMIC EXTERNAL STOP TEST/	
2329	016617	015	012		.BYTE 15,12	

2330	016621	130	036440	031440	.ASCII /X = 30 = POINT CORRELATOR ALIGNMENT CHECK/
2331	016672	015	012		.BYTE 15,12
2332	016674	020131	020075	030463	.ASCII /Y = 31 = POINT CORRELATOR ADJUSTMENT FRAMES/
2333	016747	015	012		.BYTE 15,12
2334	016751	132	036440	031440	.ASCII /Z = 32 = WRITING TABLET EXERCISER FRAME/
2335	017020	015	012	012	.BYTE 15,12,12
2336	017023	040	051040	041125	.ASCII / RUBOUT (DELETE) TO LOOP ON CURRENT PATTERN/
2337	017077	015	012		.BYTE 15,12
2338	017101	040	041440	020122	.ASCII / CR TO STOP FRAME MOTION, SPACE TO RESUME/
2339	017153	015	000012		.ASCIZ <15><12>
2340		017155			.=.-1
2341		017156			.EVEN
2342	017156	164600			CONSL1!BIT7 ;DISABLE CONSOLE #1
2343	017160	100000			CHAR
2344	017162	020040	044124	051511	.ASCIZ / THIS IS CONSOLE 0/<15><12>
2345		017207			.=.-1
2346		017210			.EVEN
2347	017210	164200			CONSLO!BIT7 ;DISABLE CONSOLE #0
2348	017212	164700			CONSL1!BIT7!BIT6 ;ENABLE CONSOLE #1
2349	017214	100000			CHAR
2350	017216	020040	044124	051511	.ASCIZ / THIS IS CONSOLE 1/<15><12>
2351		017243			.=.-1
2352		017244			.EVEN
2353	017244	164300			CONSLO!BIT7!BIT6 ; ENABLE CONSOLE #0
2354	017246	173400			DSTOP
2355	017250	160000			DJMP
2356	017252	015020			FRME0
2357					
2358					.SBTTL X AND Y POSITIONS FOR THE SETTLING TEST
2359					
2360	017254	164700			TABB: CONSL1!BIT7!BIT6 ;ENABLE CONSOLE #1
2361	017256	114000			POINT
2362	017260	041000			INTX!BIT9
2363	017262	001000			BIT9
2364	017264	040400			INTX!BIT8
2365	017266	000400			BIT8
2366	017270	040200			INTX!BIT7
2367	017272	000200			BIT7
2368	017274	040100			INTX!BIT6
2369	017276	000100			BIT6
2370	017300	040040			INTX!BIT5
2371	017302	000040			BIT5
2372	017304	040020			INTX!BIT4
2373	017306	000020			BIT4
2374	017310	040010			INTX!BIT3
2375	017312	000010			BIT3
2376	017314	040004			INTX!BIT2
2377	017316	000004			BIT2
2378	017320	040002			INTX!BIT1
2379	017322	000002			BIT1
2380	017324	040001			INTX!BIT0
2381	017326	000001			BIT0
2382	017330	040000			INTX
2383	017332	000000			0
2384					
2385	017334	041400			INTX!1400

2386	017336	000400	BIT8
2387	017340	041600	INTX!1600
2388	017342	000200	BIT7
2389	017344	041700	INTX!1700
2390	017346	000100	BIT6
2391	017350	041740	INTX!1740
2392	017352	000040	BIT5
2393	017354	041760	INTX!1760
2394	017356	000020	BIT4
2395	017360	041770	INTX!1770
2396	017362	000010	BIT3
2397	017364	041774	INTX!1774
2398	017366	000004	BIT2
2399	017370	041776	INTX!1776
2400	017372	000002	BIT1
2401	017374	041777	INTX!1777
2402	017376	000001	BIT0
2403			
2404	017400	041400	INTX!1400
2405	017402	001400	1400
2406	017404	041600	INTX!1600
2407	017406	001600	1600
2408	017410	041700	INTX!1700
2409	017412	001700	1700
2410	017414	041740	INTX!1740
2411	017416	001740	1740
2412	017420	041760	INTX!1760
2413	017422	001760	1760
2414	017424	041770	INTX!1770
2415	017426	001770	1770
2416	017430	041774	INTX!1774
2417	017432	001774	1774
2418	017434	041776	INTX!1776
2419	017436	001776	1776
2420	017440	041777	INTX!1777
2421	017442	001777	1777
2422			
2423	017444	040400	INTX!BIT8
2424	017446	001400	1400
2425	017450	040200	INTX!BIT7
2426	017452	001600	1600
2427	017454	040100	INTX!BIT6
2428	017456	001700	1700
2429	017460	040040	INTX!BIT5
2430	017462	001740	1740
2431	017464	040020	INTX!BIT4
2432	017466	001760	1760
2433	017470	040010	INTX!BIT3
2434	017472	001770	1770
2435	017474	040004	INTX!BIT2
2436	017476	001774	1774
2437	017500	040002	INTX!BIT1
2438	017502	001776	1776
2439	017504	040001	INTX!BIT0
2440	017506	001777	1777
2441	017510	173400	DSTOP

2442 017512 160000
2443 017514 017254

DJMP
TABB

;RETURN ADDRESS


```
2445
2446 ;FILE 2 <ANALOG TUNE-UP TEST >
2447
2448 017516 164700 FRME2: CONSL1!BIT7!BIT6 ;ENABLE CONSOLE #1
2449 ;INTENSIFY A POINT 3 TIMES IN EACH CORNER
2450 017520 114000 POINT ;LOWER LEFT
2451 017522 040000 INTX
2452 017524 000000 0
2453 017526 040000 INTX
2454 017530 000000 0
2455 017532 040000 INTX
2456 017534 000000 0
2457 017536 041777 INTX!MAXX ;LOWER RIGHT
2458 017540 000000 0
2459 017542 041777 INTX!MAXX
2460 017544 000000 0
2461 017546 041777 INTX!MAXX
2462 017550 000000 0
2463 017552 041777 INTX!MAXX ;UPPER RIGHT
2464 017554 001777 MAXY
2465 017556 041777 INTX!MAXX
2466 017560 001777 MAXY
2467 017562 041777 INTX!MAXX
2468 017564 001777 MAXY
2469 017566 040000 INTX ;UPPER LEFT
2470 017570 001777 MAXY
2471 017572 040000 INTX
2472 017574 001777 MAXY
2473 017576 040000 INTX
2474 017600 001777 MAXY
2475 ;NOW DRAW THE OUTER REF. BOX
2476 017602 114000 PICST0: POINT
2477 017604 000000 0
2478 017606 000000 0
2479 017610 110000 LONGV
2480 017612 041777 INTX!MAXX ; +X, +Y
2481 017614 000000 0
2482 017616 040000 INTX ; +X, +Y
2483 017620 001777 MAXY
2484 017622 061777 INTX!MINUSX!MAXX ; -X, +Y
2485 017624 000000 0
2486 017626 040000 INTX ; +X, -Y
2487 017630 021777 MINUSY!MAXY
2488 ;NOW RE-DO THE BOX WITH NEGATIVE SIGN BITS
2489 017632 060000 PICST1: INTX!MINUSX
2490 017634 001777 MAXY
2491 017636 041777 INTX!MAXX
2492 017640 020000 MINUSY
2493 017642 060000 INTX!MINUSX
2494 017644 021777 MINUSX!MAXY
2495 017646 061777 INTX!MINUSX!MAXX
2496 017650 020000 MINUSY
2497 ;NOW DRAW LOWER LEFT TO UPPER RIGHT DIAG.
2498 017652 041777 INTX!MAXX
2499 017654 001777 MAXY
2500 017656 061777 INTX!MINUSX!MAXX
```

2501	017660	021777	MINUSX!MAXY	
2502			:REPOSITION TO LOWER RIGHT AND DRAW LOWER RIGHT	
2503			: TO UPPER LEFT DIAG.	
2504	017662	001777	MAXX	
2505	017664	000000	0	
2506	017666	061777	INTX!MINUSX!MAXX	
2507	017670	001777	MAXY	
2508	017672	041777	INTX!MAXX	
2509	017674	021777	MINUSX!MAXY	
2510			.SBTTL MENU 1 SUB-PICTURE	
2511			:DRAW REF. BOX IN THE MENU	
2512	017676	170003	DMENU1	:ENABLE MENU
2513	017700	114000	POINT	
2514	017702	000000	0	
2515	017704	000000	0	
2516	017706	110000	LONGV	:DRAW REF. BOX
2517	017710	040177	INTX!177	
2518	017712	000000	0	
2519	017714	040000	INTX	
2520	017716	001777	MAXY	
2521	017720	060177	INTX!MINUSX!177	
2522	017722	000000	0	
2523	017724	040000	INTX	
2524	017726	021777	MINUSX!MAXY	
2525			:NOW REVERSE THE DRAWING PROCEDURE	
2526	017730	060000	INTX!MINUSX	
2527	017732	001777	MAXY	
2528	017734	040177	INTX!177	
2529	017736	020000	MINUSX	
2530	017740	060000	INTX!MINUSX	
2531	017742	021777	MINUSX!MAXY	
2532	017744	060177	INTX!MINUSX!177	
2533	017746	020000	MINUSX	
2534			:NOW DRAW THE DIAG. X IN THE MENU	
2535	017750	040177	INTX!177	:LOWER LEFT, IN MENU, TO UPPER RIGHT
2536	017752	001777	MAXY	
2537	017754	060177	INTX!MINUSX!177	
2538	017756	021777	MINUSX!MAXY	
2539	017760	000177	177	:REPOSITION TO LOWER LEFT OF MENU
2540	017762	000000	0	
2541	017764	060177	INTX!MINUSX!177	:LOWER RIGHT TO UPPER LEFT
2542	017766	001777	MAXY	
2543	017770	040177	INTX!177	
2544	017772	021777	MINUSX!MAXY	
2545	017774	170002	DMENU0	:RETURN TO MAIN SCREEN
2546			:CONTINUE MAIN SCREEN PICTURE	
2547	017776	114000	POINT	
2548	020000	001400	1400	
2549	020002	001000	1000	
2550			:DRAW A 100 UNIT BOX, SAME METHOD AS OUTER REF. BOX	
2551	020004	110000	LONGV	
2552	020006	040144	INTX!100.	: +X, +Y
2553	020010	000000	0	
2554	020012	040000	INTX	: +X, +Y
2555	020014	000144	100.	
2556	020016	060144	INTX!MINUSX!100.	: -X, +Y

2557	020020	000000	0	
2558	020022	040000	INTX	; +X, -Y
2559	020024	020144	MINUSY!100.	
2560	020026	040144	INTX!100.	; +X, -Y
2561	020030	020000	MINUSY	
2562	020032	060000	INTX!MINUSX	; -X, +Y
2563	020034	000144	100.	
2564	020036	060144	INTX!MINUSX!100.	; -X, -Y
2565	020040	020000	MINUSY	
2566	020042	060000	INTX!MINUSX	; -X, -Y
2567	020044	020144	MINUSY!100.	
2568	020046	040144	INTX!100.	
2569	020050	000144	100.	
2570	020052	060144	INTX!MINUSX!100.	
2571	020054	020144	MINUSX!100.	
2572	020056	000144	100.	
2573	020060	000000	0	
2574	020062	060144	INTX!MINUSX!100.	
2575	020064	000144	100.	
2576	020066	040144	INTX!100.	
2577	020070	020144	MINUSX!100.	
2578			;DRAW A 10 UNIT BOX, SAME METHOD AS OUTER BOX	
2579	020072	114000	POINT	
2580	020074	001400	1400	
2581	020076	000700	700	
2582	020100	110000	LONGV	
2583	020102	040012	INTX!10.	; +X, +Y
2584	020104	000000	0	
2585	020106	040000	INTX	; +X, +Y
2586	020110	000012	10.	
2587	020112	060012	INTX!MINUSX!10.	; -X, +Y
2588	020114	000000	0	
2589	020116	040000	INTX	; +X, -Y
2590	020120	020012	MINUSY!10.	
2591	020122	040012	INTX!10.	; +X, -Y
2592	020124	020000	MINUSY	
2593	020126	060000	INTX!MINUSX	; -X, +Y
2594	020130	000012	10.	
2595	020132	060012	INTX!MINUSX!10.	; -X, -Y
2596	020134	020000	MINUSY	
2597	020136	060000	INTX!MINUSX	; -X, -Y
2598	020140	020012	MINUSY!10.	
2599	020142	040012	INTX!10.	
2600	020144	000012	10.	
2601	020146	060012	INTX!MINUSX!10.	
2602	020150	020012	MINUSX!10.	
2603	020152	000012	10.	
2604	020154	000000	0	
2605	020156	060012	INTX!MINUSX!10.	
2606	020160	000012	10.	
2607	020162	040012	INTX!10.	
2608	020164	020012	MINUSX!10.	
2609			;DRAW FOUR VECTORS FROM A "COMMON" POINT WHICH WILL BE THE	
2610			;SUPERIMPOSED UPON BY THE NEXT SUB-PICTURE	
2611	020166	114000	POINT	
2612	020170	001000	1000	

2613	020172	000400	400	
2614	020174	110000	LONGV	
2615	020176	040000	INTX	
2616	020200	000200	200	
2617	020202	114000	POINT	
2618	020204	001000	1000	
2619	020206	000400	400	
2620	020210	110000	LONGV	
2621	020212	040200	INTX!200	
2622	020214	000000	0	
2623	020216	114000	POINT	
2624	020220	001000	1000	
2625	020222	000400	400	
2626	020224	110000	LONGV	
2627	020226	040000	INTX	
2628	020230	020200	MINUSY!200	
2629	020232	114000	POINT	
2630	020234	001000	1000	
2631	020236	000400	400	
2632	020240	110000	LONGV	
2633	020242	060200	INTX!MINUSX!200	
2634	020244	000000	0	
2635			.SBTTL	DRAW 10 VERTICAL VECTORS IN THE LEFT CENTER AERA
2636	020246	114000	POINT	
2637	020250	000200	200	
2638	020252	000740	740	
2650	020254	104000	SHORTV	
(1)	020256	040010	INTX!10	;DRAW A 8. UNIT VERTICAL VECTOR
(1)	020260	130000	RELATP	
(1)	020262	040002	INTX!2	;INTENSIFY A POINT 2 UNITS AWAY
(1)	020264	000002	2	;MOVE THE Y AXIS
(1)	020266	104000	SHORTV	
(1)	020270	060010	INTX!MINUSX!10	;DRAW A 8. UNIT VERTICAL VECTOR
(1)	020272	130000	RELATP	
(1)	020274	040002	INTX!2	;INTENSIFY A POINT 2 UNITS AWAY
(1)	020276	000002	2	;MOVE THE Y AXIS
(1)	020300	104000	SHORTV	
(1)	020302	040010	INTX!10	;DRAW A 8. UNIT VERTICAL VECTOR
(1)	020304	130000	RELATP	
(1)	020306	040002	INTX!2	;INTENSIFY A POINT 2 UNITS AWAY
(1)	020310	000002	2	;MOVE THE Y AXIS
(1)	020312	104000	SHORTV	
(1)	020314	060010	INTX!MINUSX!10	;DRAW A 8. UNIT VERTICAL VECTOR
(1)	020316	130000	RELATP	
(1)	020320	040002	INTX!2	;INTENSIFY A POINT 2 UNITS AWAY
(1)	020322	000002	2	;MOVE THE Y AXIS
(1)	020324	104000	SHORTV	
(1)	020326	040010	INTX!10	;DRAW A 8. UNIT VERTICAL VECTOR
(1)	020330	130000	RELATP	
(1)	020332	040002	INTX!2	;INTENSIFY A POINT 2 UNITS AWAY
(1)	020334	000002	2	;MOVE THE Y AXIS
(1)	020336	104000	SHORTV	
(1)	020340	060010	INTX!MINUSX!10	;DRAW A 8. UNIT VERTICAL VECTOR
(1)	020342	130000	RELATP	
(1)	020344	040002	INTX!2	;INTENSIFY A POINT 2 UNITS AWAY
(1)	020346	000002	2	;MOVE THE Y AXIS


```
(1) 020350 104000 SHORTV
(1) 020352 040010 INTX!10 ;DRAW A 8. UNIT VERTICAL VECTOR
(1) 020354 130000 RELATP
(1) 020356 040002 INTX!2 ;INTENSIFY A POINT 2 UNITS AWAY
(1) 020360 000002 2 ;MOVE THE Y AXIS
(1) 020362 104000 SHORTV
(1) 020364 060010 INTX!MINUSX!10 ;DRAW A 8. UNIT VERTICAL VECTOR
(1) 020366 130000 RELATP
(1) 020370 040002 INTX!2 ;INTENSIFY A POINT 2 UNITS AWAY
(1) 020372 000002 2 ;MOVE THE Y AXIS
(1) 020374 104000 SHORTV
(1) 020376 040010 INTX!10 ;DRAW A 8. UNIT VERTICAL VECTOR
(1) 020400 130000 RELATP
(1) 020402 040002 INTX!2 ;INTENSIFY A POINT 2 UNITS AWAY
(1) 020404 000002 2 ;MOVE THE Y AXIS
(1) 020406 104000 SHORTV
(1) 020410 060010 INTX!MINUSX!10 ;DRAW A 8. UNIT VERTICAL VECTOR
(1) 020412 130000 RELATP
(1) 020414 040002 INTX!2 ;INTENSIFY A POINT 2 UNITS AWAY
(1) 020416 000002 2 ;MOVE THE Y AXIS
2651 .SBTTL DRAW THE DELAY INTENSITY SUB-PICTURE IN THE LEFT CENTER AERA
2652 : DRAW 8 VECTORS USING BASIC VECTOR INSTRUCTION AWAY FROM A COMMON POINT
2653 : BUT OFFSET BY ONE UNIT.
2654 : THE COMMON POINT X=353 AND Y =1003
2655 :
2656 000353 XQ6=353
2657 001003 YQ6=1003
2658 000040 LQ6=40
2659 020420 114000 POINT
2660 020422 000354 XQ6+1
2661 020424 001003 YQ6
2662 020426 120000 BASICV
2663 020430 042040 INTX!PATH0!LQ6
2664 020432 114000 POINT
2665 020434 000354 XQ6+1
2666 020436 001004 YQ6+1 ;VECTOR #1
2667 020440 120000 BASICV
2668 020442 046040 INTX!PATH1!LQ6
2669 020444 114000 POINT
2670 020446 000353 XQ6
2671 020450 001004 YQ6+1 ;VECTOR #2
2672 020452 120000 BASICV
2673 020454 052040 INTX!PATH2!LQ6
2674 020456 114000 POINT
2675 020460 000352 XQ6-1
2676 020462 001004 YQ6+1 ;VECTOR #3
2677 020464 120000 BASICV
2678 020466 056040 INTX!PATH3!LQ6
2679 020470 114000 POINT
2680 020472 000352 XQ6-1
2681 020474 001003 YQ6 ;VECTOR #4
2682 020476 120000 BASICV
2683 020500 062040 INTX!PATH4!LQ6
2684 020502 114000 POINT
2685 020504 000352 XQ6-1
2686 020506 001002 YQ6-1 ;VECTOR #5
```

2687	020510	120000	BASICV	
2688	020512	066040	INTX!PATH5!LQ6	
2689	020514	114000	POINT	
2690	020516	000353	XQ6	
2691	020520	001002	YQ6-1	:VECTOR #6
2692	020522	120000	BASICV	
2693	020524	072040	INTX!PATH6!LQ6	
2694	020526	114000	POINT	
2695	020530	000354	XQ6+1	
2696	020532	001002	YQ6-1	:VECTOR #7
2697	020534	120000	BASICV	
2698	020536	076040	INTX!PATH7!LQ6	
2699				
2700	020540	173400	DSTOP	
2701	020542	160000	DJMP	
2702	020544	017516	FRME2	


```
2704 .SBTTL OCTAGONS USING LONG AND ABSOLUTE VECTORS (WIDTHS OF 7,77,177,377 AND 52
2705 .SBTTL CIRCLES USING ABSOLUTE VECTORS (WIDTHS OF 64., 128., AND 256.)
2706
2707 020546 114000 FRME3: PCINT
2708 020550 000774 774
2709 020552 000764 764
2710 020554 164700 CONSL1!BIT7!BIT6 ;ENABLE CONSOLE #1
2730 020556 110000 LONGV ;OCTOGON BY LENGTH OF 7
(1) 020560 040007 INTX+7
(1) 020562 000000 0
(1) 020564 040007 INTX+7
(1) 020566 000007 7
(1) 020570 040000 INTX
(1) 020572 000007 7
(1) 020574 060007 INTX!MINUSX+7
(1) 020576 000007 7
(1) 020600 060007 INTX!MINUSX+7
(1) 020602 000000 0
(1) 020604 060007 INTX!MINUSX+7
(1) 020606 020007 MINUSX+7
(1) 020610 040000 INTX
(1) 020612 020007 MINUSX+7
(1) 020614 040007 INTX+7
(1) 020616 020007 MINUSX+7
```

;CIRCLE 8 DEG. RADIUS OF 64

2732
2733
2734 020620 114000
2735 020622 001077
2736 020624 000777
2737 020626 144000
2738 020630 041076 001010
2739 020634 041075 001021
2740 020640 041071 001031
2741 020644 041065 001041
2742 020650 041060 001050
2743 020654 041052 001057
2744 020660 041043 001064
2745 020664 041033 001071
2746 020670 041023 001074
2747 020674 041012 001076
2748 020700 041001 001077
2749 020704 040771 001077
2750 020710 040761 001075
2751 020714 040750 001072
2752 020720 040740 001066
2753 020724 040731 001061
2754 020730 040722 001053
2755 020734 040714 001045
2756 020740 040707 001035
2757 020744 040704 001025
2758 020750 040701 001014
2759 020754 040700 001003
2760 020760 040700 000774
2761 020764 040701 000763
2762 020770 040704 000752
2763 020774 040707 000742
2764 021000 040714 000732
2765 021004 040722 000724
2766 021010 040731 000716
2767 021014 040740 000711
2768 021020 040750 000705
2769 021024 040760 000702
2770 021030 040771 000700
2771 021034 041001 000700
2772 021040 041012 000701
2773 021044 041023 000703
2774 021050 041033 000706
2775 021054 041043 000713
2776 021060 041052 000720
2777 021064 041060 000727
2778 021070 041065 000736
2779 021074 041071 000746
2780 021100 041075 000756
2781 021104 041076 000767
2782 021110 041077 000777
2783 021114 164000
2784 021116 164000
2785 021120 164000
2786 021122 164000
2787 021124 164000

POINT
1077
777
ABSVCT
.WORD INTX!1076,1010
.WORD INTX!1075,1021
.WORD INTX!1071,1031
.WORD INTX!1065,1041
.WORD INTX!1060,1050
.WORD INTX!1052,1057
.WORD INTX!1043,1064
.WORD INTX!1033,1071
.WORD INTX!1023,1074
.WORD INTX!1012,1076
.WORD INTX!1001,1077
.WORD INTX!771,1077
.WORD INTX!761,1075
.WORD INTX!750,1072
.WORD INTX!740,1066
.WORD INTX!731,1061
.WORD INTX!722,1053
.WORD INTX!714,1045
.WORD INTX!707,1035
.WORD INTX!704,1025
.WORD INTX!701,1014
.WORD INTX!700,1003
.WORD INTX!700,774
.WORD INTX!701,763
.WORD INTX!704,752
.WORD INTX!707,742
.WORD INTX!714,732
.WORD INTX!722,724
.WORD INTX!731,716
.WORD INTX!740,711
.WORD INTX!750,705
.WORD INTX!760,702
.WORD INTX!771,700
.WORD INTX!1001,700
.WORD INTX!1012,701
.WORD INTX!1023,703
.WORD INTX!1033,706
.WORD INTX!1043,713
.WORD INTX!1052,720
.WORD INTX!1060,727
.WORD INTX!1065,736
.WORD INTX!1071,746
.WORD INTX!1075,756
.WORD INTX!1076,767
.WORD INTX!1077,777
DNOP
DNOP
DNOP
DNOP
DNOP

;ENABLE ABSOLUTE VECTOR MODE

8 8

:CIRCLE 8 DEG. RADIUS OF 128

:DISPLAY IN ABSOLUTE VECTOR MODE

2788					
2789				POINT	
2790	021126	114000		1177	
2791	021130	001177		777	
2792	021132	000777		ABSVCT	
2793	021134	144000		.WORD	INTX!1176,1021
2794	021136	041176	001021	.WORD	INTX!1172,1042
2795	021142	041172	001042	.WORD	INTX!1164,1063
2796	021146	041164	001063	.WORD	INTX!1154,1103
2797	021152	041154	001103	.WORD	INTX!1141,1121
2798	021156	041141	001121	.WORD	INTX!1125,1136
2799	021162	041125	001136	.WORD	INTX!1107,1151
2800	021166	041107	001151	.WORD	INTX!1067,1162
2801	021172	041067	001162	.WORD	INTX!1047,1171
2802	021176	041047	001171	.WORD	INTX!1025,1175
2803	021202	041025	001175	.WORD	INTX!1003,1177
2804	021206	041003	001177	.WORD	INTX!763,1176
2805	021212	040763	001176	.WORD	INTX!741,1173
2806	021216	040741	001173	.WORD	INTX!720,1166
2807	021222	040720	001166	.WORD	INTX!700,1156
2808	021226	040700	001156	.WORD	INTX!661,1144
2809	021232	040661	001144	.WORD	INTX!644,1130
2810	021236	040644	001130	.WORD	INTX!630,1112
2811	021242	040630	001112	.WORD	INTX!617,1073
2812	021246	040617	001073	.WORD	INTX!610,1053
2813	021252	040610	001053	.WORD	INTX!603,1032
2814	021256	040603	001032	.WORD	INTX!600,1010
2815	021262	040600	001010	.WORD	INTX!600,767
2816	021266	040600	000767	.WORD	INTX!603,745
2817	021272	040603	000745	.WORD	INTX!610,724
2818	021276	040610	000724	.WORD	INTX!617,704
2819	021302	040617	000704	.WORD	INTX!630,665
2820	021306	040630	000665	.WORD	INTX!644,647
2821	021312	040644	000647	.WORD	INTX!661,633
2822	021316	040661	000633	.WORD	INTX!700,621
2823	021322	040700	000621	.WORD	INTX!720,611
2824	021326	040720	000611	.WORD	INTX!741,604
2825	021332	040741	000604	.WORD	INTX!763,601
2826	021336	040763	000601	.WORD	INTX!1003,600
2827	021342	041003	000600	.WORD	INTX!1025,602
2828	021346	041025	000602	.WORD	INTX!1047,606
2829	021352	041047	000606	.WORD	INTX!1067,615
2830	021356	041067	000615	.WORD	INTX!1107,626
2831	021362	041107	000626	.WORD	INTX!1125,641
2832	021366	041125	000641	.WORD	INTX!1141,656
2833	021372	041141	000656	.WORD	INTX!1154,674
2834	021376	041154	000674	.WORD	INTX!1164,714
2835	021402	041164	000714	.WORD	INTX!1172,735
2836	021406	041172	000735	.WORD	INTX!1176,756
2837	021412	041176	000756	.WORD	INTX!1177,777
2838	021416	041177	000777	.WORD	
2839	021422	164000		DNOP	
2840	021424	164000		DNOP	
2841	021426	164000		DNOP	
2842	021430	164000		DNOP	
2843					

:CIRCLE 8 DEG. RADIUS OF 256

2844				
2845	021432	114000		POINT
2846	021434	001377		1377
2847	021436	000777		777
2848	021440	144000		ABSVCT
2849	021442	041375	001043	.WORD INTX!1375,1043
2850	021446	041365	001106	.WORD INTX!1365,1106
2851	021452	041351	001147	.WORD INTX!1351,1147
2852	021456	041330	001207	.WORD INTX!1330,1207
2853	021462	041303	001244	.WORD INTX!1303,1244
2854	021466	041252	001275	.WORD INTX!1252,1275
2855	021472	041216	001323	.WORD INTX!1216,1323
2856	021476	041157	001345	.WORD INTX!1157,1345
2857	021502	041116	001362	.WORD INTX!1116,1362
2858	021506	041053	001373	.WORD INTX!1053,1373
2859	021512	041010	001377	.WORD INTX!1010,1377
2860	021516	040745	001376	.WORD INTX!745,1376
2861	021522	040702	001367	.WORD INTX!702,1367
2862	021526	040640	001354	.WORD INTX!640,1354
2863	021532	040600	001335	.WORD INTX!600,1335
2864	021536	040542	001311	.WORD INTX!542,1311
2865	021542	040510	001261	.WORD INTX!510,1261
2866	021546	040461	001226	.WORD INTX!461,1226
2867	021552	040436	001167	.WORD INTX!436,1167
2868	021556	040417	001127	.WORD INTX!417,1127
2869	021562	040406	001064	.WORD INTX!406,1064
2870	021566	040401	001021	.WORD INTX!401,1021
2871	021572	040401	000756	.WORD INTX!401,756
2872	021576	040406	000713	.WORD INTX!406,713
2873	021602	040417	000651	.WORD INTX!417,651
2874	021606	040436	000610	.WORD INTX!436,610
2875	021612	040461	000552	.WORD INTX!461,552
2876	021616	040510	000516	.WORD INTX!510,516
2877	021622	040542	000466	.WORD INTX!542,466
2878	021626	040600	000442	.WORD INTX!600,442
2879	021632	040640	000423	.WORD INTX!640,423
2880	021636	040702	000410	.WORD INTX!702,410
2881	021642	040745	000401	.WORD INTX!745,401
2882	021646	041010	000400	.WORD INTX!1010,400
2883	021652	041053	000404	.WORD INTX!1053,404
2884	021656	041116	000415	.WORD INTX!1116,415
2885	021662	041157	000432	.WORD INTX!1157,432
2886	021666	041216	000454	.WORD INTX!1216,454
2887	021672	041252	000502	.WORD INTX!1252,502
2888	021676	041303	000533	.WORD INTX!1303,533
2889	021702	041330	000570	.WORD INTX!1330,570
2890	021706	041351	000630	.WORD INTX!1351,630
2891	021712	041365	000671	.WORD INTX!1365,671
2892	021716	041374	000734	.WORD INTX!1374,734
2893	021722	041377	000777	.WORD INTX!1377,777
2894	021726	164000		DNOP
2895	021730	164000		DNOP
2896	021732	164000		DNOP
2897	021734	164000		DNOP
2898	021736	114000		POINT
2899	021740	000740		740

:ENABLE ABSOLUTE VECTOR MODE

2900	021742	000640	640	
2901	021744	110000	LONGV	:OCTOGON BY LENGTH OF 77
(1)	021746	040077	INTX+77	
(1)	021750	000000	0	
(1)	021752	040077	INTX+77	
(1)	021754	000077	77	
(1)	021756	040000	INTX	
(1)	021760	000077	77	
(1)	021762	060077	INTX!MINUSX+77	
(1)	021764	000077	77	
(1)	021766	060077	INTX!MINUSX+77	
(1)	021770	000000	0	
(1)	021772	060077	INTX!MINUSX+77	
(1)	021774	020077	MINUSX+77	
(1)	021776	040000	INTX	
(1)	022000	020077	MINUSX+77	
(1)	022002	040077	INTX+77	
(1)	022004	020077	MINUSX+77	
2902	022006	114000	POINT	
2903	022010	000700	700	
2904	022012	000500	500	
2905	022014	110000	LONGV	:OCTOGON BY LENGTH OF 177
(1)	022016	040177	INTX+177	
(1)	022020	000000	0	
(1)	022022	040177	INTX+177	
(1)	022024	000177	177	
(1)	022026	040000	INTX	
(1)	022030	000177	177	
(1)	022032	060177	INTX!MINUSX+177	
(1)	022034	000177	177	
(1)	022036	060177	INTX!MINUSX+177	
(1)	022040	000000	0	
(1)	022042	060177	INTX!MINUSX+177	
(1)	022044	020177	MINUSX+177	
(1)	022046	040000	INTX	
(1)	022050	020177	MINUSX+177	
(1)	022052	040177	INTX+177	
(1)	022054	020177	MINUSX+177	
2906	022056	114000	POINT	
2907	022060	000600	600	
2908	022062	000200	200	
2909	022064	110000	LONGV	:OCTOGON BY LENGTH OF 377
(1)	022066	040377	INTX+377	
(1)	022070	000000	0	
(1)	022072	040377	INTX+377	
(1)	022074	000377	377	
(1)	022076	040000	INTX	
(1)	022100	000377	377	
(1)	022102	060377	INTX!MINUSX+377	
(1)	022104	000377	377	
(1)	022106	060377	INTX!MINUSX+377	
(1)	022110	000000	0	
(1)	022112	060377	INTX!MINUSX+377	
(1)	022114	020377	MINUSX+377	
(1)	022116	040000	INTX	
(1)	022120	020377	MINUSX+377	

```

(1) 022122 040377          INTX+377
(1) 022124 020377          MINUSX+377
2910          ;DRAW ABSOLUTE VECTOR OCTAGON
2911 022126 114000          POINT
2912 022130 000530          530
2913 022132 000010          10
2914 022134 144000          ABSVCT
2915 022136 041250          INTX!1250          ; #1
2916 022140 000010          10
2917 022142 041770          INTX!1770          ; #2
2918 022144 000530          530
2919 022146 041770          INTX!1770          ; #3
2920 022150 001250          1250
2921 022152 041250          INTX!1250          ; #4
2922 022154 001770          1770
2923 022156 040530          INTX!530          ; #5
2924 022160 001770          1770
2925 022162 040010          INTX!10          ; #6
2926 022164 001250          1250
2927 022166 040010          INTX!10          ; #7
2928 022170 000530          530
2929 022172 040530          INTX!530          ; #8
2930 022174 000010          10
2931          ;DRAW A BASIC SHORT VECTOR OCTAGON
2932 022176 114000          POINT
2933 022200 000300          300
2934 022202 001000          1000
2935 022204 134000          BASICS          ;BASIC SHORT VECTOR
2936 022206 073777          73777          ;PATH 6 & 7
2937 022210 063737          63737          ;PATH 4 & 5
2938 022212 053677          53677          ;PATH 2 & 3
2939 022214 043637          43637          ;PATH 0 & 1
2940          ;DRAW ANOTHER IN THE RIGHT CENTER
2941 022216 114000          POINT
2942 022220 001500          1500
2943 022222 001000          1000
2944 022224 134000          BASICS
2945 022226 073737          73737          ;PATH 6 & 5
2946 022230 043777          43777          ;PATH 0 & 7
2947 022232 053637          53637          ;PATH 2 & 1
2948 022234 063677          63677          ;PATH 4 & 3
2949 022236 173400          DSTOP
2950 022240 160000          DJMP
2951 022242 020546          FRME3
2952          .SBTTL X AND Y OFFSET SUB-PICTURE
    
```



```

2954 022244 114000 OFFTST: POINT
2955 022246 010000 OFFT1: BIT12
2956 022250 010000 OFFT2: BIT12
2957 022252 164700 CONSL1!BIT7!BIT6 ;ENABLE CONSOLE #1
2958 022254 117000 POINT!INT4
2959 022256 000400 400
2960 022260 000400 400
2961 022262 120000 BASICV
2962 022264 043000 INTX!PATH0!1000 ;DRAW A SQUARE
2963 022266 053000 INTX!PATH2!1000
2964 022270 063000 INTX!PATH4!1000
2965 022272 073000 INTX!PATH6!1000
2966 022274 173400 DSTOP
2967 022276 160000 DJMP
2968 022300 022244 OFFTST
2969
2970 .SBTTL SUPER AND SUBSCRIPT SUB-PICTURE
2971
2972 022302 114000 SUPPIC: POINT
2973 022304 000400 400
2974 022306 001000 1000
2975 022310 110000 LONGV
2976 022312 041000 INTX!1000 ;DRAW REF. LINE
2977 022314 000000 0
2978 022316 114000 POINT
2979 022320 000400 400
2980 022322 001000 1000
2981 022324 160000 DJMP ;BYPASS ROTATED REF. LINE
2982 022326 022352 SUPC1
2983 022330 114000 SUPCO: POINT
2984 022332 001000 1000
2985 022334 000400 400
2986 022336 110000 LONGV
2987 022340 040000 INTX
2988 022342 001000 1000
2989 022344 114000 POINT
2990 022346 001000 1000
2991 022350 000400 400
2992 022352 154340 SUPC1: CHARS3 ;ENSURE MAX CHAR SIZE
2993 022354 170040 STATSA!ITAL0
2994 022356 100000 CHAR
2995 022360 162000 DJSR
2996 022362 022402 SUPSUB
2997 022364 170060 STATSA!ITAL1 ;SET ITALIC
2998 022366 162000 DJSR
2999 022370 022402 SUPSUB
3000 022372 154240 CHARS1
3001 022374 173400 DSTOP
3002 022376 160000 DJMP
3003 022400 022302 SUPPIC
3004
3005 .SBTTL SUPER AND SUBSCRIPT ASCII STRING
3006
3007 022402 105 021 105 SUPSUB: .BYTE 105,SUPON,105,SUPON,105,SUPON,105
3008 ;NOW REVERSE AND INCREASE SIZE
3009 022411 023 105 023 .BYTE SUPOFF,105,SUPOFF,105,SUPOFF,105
  
```

```

3010 ;NOW IT SHOULD BE AT THE BIGGEST SIZE
3011 022417 022 105 022 .BYTE SUBON,105,SUBON,105,SUBON,105
3012 ;REVERSE AND INCREASE SIZE
3013 022425 024 105 024 .BYTE SUBOFF,105,SUBOFF,105,SUBOFF,105
3014 022433 040 .BYTE 40
3015
3016 022434 166000 DPOP
3017
3018
3019 .SBTTL SYNC SPEED SUBPICTURE
3020
3021 022436 170000 SYNPIC: STATSA ;VARIABLE WORD TO HANDLE SYNC SPEED
3022 022440 164700 CONSL1!BIT7!BIT6 ;ENABLE CONSOLE #1
3023 022442 114000 POINT
3024 022444 001000 1000
3025 022446 000000 0
3026 022450 120000 BASICV
3027 022452 047000 INTX!PATH1!1000 ;DRAW A DIAMOND
3028 022454 057000 INTX!PATH3!1000
3029 022456 067000 INTX!PATH5!1000
3030 022460 077000 INTX!PATH7!1000
3031 022462 114000 POINT
3032 022464 000600 600
3033 022466 001000 1000
3034 022470 176003 STRNG1 ;ENABLE CHARACTER TERMINATE
3035 022472 100000 CHAR
3036 022474 162000 DJSR
3037 022476 022510 SYNTAX ;DISPLAY SYNC SPEED MESSAGE
3038 022500 176002 STRNG0 ;DISABLE CHARACTER STRING ESCAPE
3039 022502 173400 DSTOP
3040 022504 160000 DJMP
3041 022506 022436 SYNPIC ;CONTINUE
3042
3043 022510 044124 051511 043040 SYNTAX: .ASCII /THIS FRAME USES /
3044 022530 047516 047131 020103 SYNSPD: .ASCII /NO/
3045 022532 051440 .ASCII / SYNC /
3046 022555 177 .BYTE 177
3047
3048 ;SHOULD NEVER GET HERE UNLESS CHAR TERM. FAILS
3049 022556 114000 1$: POINT
3050 022560 000200 200
3051 022562 000700 700
3052 022564 100000 CHAR
3053 022566 044103 051101 041501 .ASCIZ /CHARACTER TERMINATE FAILED TO CAUSE A POP AND RESTORE/
3054 022654 160000 DJMP
3055 022656 022556 1$
3056
3057 .SBTTL DASH LINE SUB-PICTURE
3058
3059 022660 117000 FRMES: POINT!INT4
3060 022662 000000 0
3061 022664 001000 1000
3062 022666 154240 CHARS1
3063 022670 164700 CONSL1!BIT7!BIT6 ;ENABLE CONSOLE #1
3064 022672 100004 CHAR!LINE0
3065 022674 047523 044514 020104 .ASCII /SOLID /
    
```


3066	022706	110004			LONGV:LINE0	
3067	022710	040400			40400	
3068	022712	000000			0	
3069	022714	000400			400	
3070	022716	000000			0	
3071	022720	110030			LONGV:BLKON	
3072	022722	040400			40400	
3073	022724	000000			0	
3074	022726	100020			CHAR:BLKOFF	
3075	022730	015	012	012	.BYTE 15,12,12,12,12,12	
3076	022736	040504	044123	044440	.ASCII /DASH I /	
3077	022750	110005			LONGV:LINE1	
3078	022752	040400			40400	
3079	022754	000000			0	
3080	022756	000400			400	
3081	022760	000000			0	
3082	022762	110030			LONGV:BLKON	
3083	022764	040400			40400	
3084	022766	000000			0	
3085	022770	100020			CHAR:BLKOFF	
3086	022772	015	012	012	.BYTE 15,12,12,12,12,12	
3087	023000	040504	044123	044440	.ASCII /DASH II /	
3088	023012	110006			LONGV:LINE2	
3089	023014	040400			40400	
3090	023016	000000			0	
3091	023020	000400			400	
3092	023022	000000			0	
3093	023024	110030			LONGV:BLKON	
3094	023026	040400			40400	
3095	023030	000000			0	
3096	023032	100020			CHAR:BLKOFF	
3097	023034	015	012	012	.BYTE 15,12,12,12,12,12	
3098	023042	040504	044123	044440	.ASCII /DASH III /	
3099	023054	110007			LONGV:LINE3	
3100	023056	040400			40400	
3101	023060	000000			0	
3102	023062	000400			400	
3103	023064	000000			0	
3104	023066	110030			LONGV:BLKON	
3105	023070	040400			40400	
3106	023072	000000			0	
3107	023074	110024			LONGV:BLKOFF:LINE0	
3108	023076	000000			0	
3109	023100	000000			0	
3110	023102	173400			DSTOP	
3119	023104	161665			DJMPR:BIT8!WHERE1	;DJMP RELATIVE TO THE TAG "FRME5"
3120						
3121					.SBTTL VECTOR LENGTH SUB-PICTURE	
3122						
3123	023106	154024	FRME6:		VCTR00!4	;NORMAL VECTOR
3124	023110	114000			POINT	
3125	023112	001777			MAXX	
3126	023114	000000			0	
3127	023116	164700			CONSL1:BIT7:BIT6	;ENABLE CONSOLE #1
3128	023120	113600			LONGV:INT7	
3129	023122	040000			INTX	

3130 023124 001777
3131 023126 114000
3132 023130 000000
3133 023132 001777
3134 023134 110000
3135 023136 041777
3136 023140 000000
3137 023142 114000
3138 023144 000000
3139 023146 000000
3140 023150 154037
3141 023152 144000
3142 023154 160000
3143 023156 036656
3144
3145
3146
3147

MAXY
POINT
0
MAXY
LONGV
INTX!MAXX
0
POINT
0
0
VCTROO!17
ABSVCT
DJMP
BUFFER

;MAX LENGTH VECTOR
;ABSOLUTE VECTOR

.SBTTL HORIZONTAL PHOSPHOR SUB-PICTURE

(1)	023336	002002	PATH0!2	:MOVE RIGHT 2 UNITS
(1)	023340	053777	INTX!PATH2!MAXY	:VECTOR STRAIGHT UP
(1)	023342	002002	PATH0!2	:MOVE RIGHT 2 UNITS
(1)	023344	073777	INTX!PATH6!MAXY	:VECTOR DOWN
(1)	023346	002002	PATH0!2	:MOVE RIGHT 2 UNITS
(1)	023350	053777	INTX!PATH2!MAXY	:VECTOR STRAIGHT UP
(1)	023352	002002	PATH0!2	:MOVE RIGHT 2 UNITS
(1)	023354	073777	INTX!PATH6!MAXY	:VECTOR DOWN
(1)	023356	002002	PATH0!2	:MOVE RIGHT 2 UNITS
(1)	023360	053777	INTX!PATH2!MAXY	:VECTOR STRAIGHT UP
(1)	023362	002002	PATH0!2	:MOVE RIGHT 2 UNITS
(1)	023364	073777	INTX!PATH6!MAXY	:VECTOR DOWN
(1)	023366	002002	PATH0!2	:MOVE RIGHT 2 UNITS
(1)	023370	053777	INTX!PATH2!MAXY	:VECTOR STRAIGHT UP
(1)	023372	002002	PATH0!2	:MOVE RIGHT 2 UNITS
(1)	023374	073777	INTX!PATH6!MAXY	:VECTOR DOWN
(1)	023376	002002	PATH0!2	:MOVE RIGHT 2 UNITS
(1)	023400	053777	INTX!PATH2!MAXY	:VECTOR STRAIGHT UP
(1)	023402	002002	PATH0!2	:MOVE RIGHT 2 UNITS
(1)	023404	073777	INTX!PATH6!MAXY	:VECTOR DOWN
(1)	023406	002002	PATH0!2	:MOVE RIGHT 2 UNITS
(1)	023410	053777	INTX!PATH2!MAXY	:VECTOR STRAIGHT UP
(1)	023412	002002	PATH0!2	:MOVE RIGHT 2 UNITS
(1)	023414	073777	INTX!PATH6!MAXY	:VECTOR DOWN
(1)	023416	002002	PATH0!2	:MOVE RIGHT 2 UNITS
(1)	023420	053777	INTX!PATH2!MAXY	:VECTOR STRAIGHT UP
(1)	023422	002002	PATH0!2	:MOVE RIGHT 2 UNITS
(1)	023424	073777	INTX!PATH6!MAXY	:VECTOR DOWN
(1)	023426	002002	PATH0!2	:MOVE RIGHT 2 UNITS
3160	023430	173400	DSTOP	
3161	023432	160000	DJMP	
3162	023434	023166	DFI10A	
3163				
3164			.SBTTL	MAIN VERTICAL PHOSPHOR SUB-PICTURE
3165				
3166	023436	114000	FRME11: POINT	
3167	023440	000000	0	
3168	023442	000000	DELTY7: 0	
3169	023444	170002	DMENUO	
3170	023446	123600	DFI11C: BASICV!INT7	
3176	023450	043777	INTX!PATH0!MAXX	:VECTOR RIGHT FULL WIDTH
(1)	023452	012002	PATH2!2	:MOVE UP 2 UNITS
(1)	023454	063777	INTX!PATH4!MAXX	:VECTOR LEFT FULL SCREEN
(1)	023456	012002	PATH2!2	:MOVE UP 2 UNITS
(1)	023460	043777	INTX!PATH0!MAXX	:VECTOR RIGHT FULL WIDTH
(1)	023462	012002	PATH2!2	:MOVE UP 2 UNITS
(1)	023464	063777	INTX!PATH4!MAXX	:VECTOR LEFT FULL SCREEN
(1)	023466	012002	PATH2!2	:MOVE UP 2 UNITS
(1)	023470	043777	INTX!PATH0!MAXX	:VECTOR RIGHT FULL WIDTH
(1)	023472	012002	PATH2!2	:MOVE UP 2 UNITS
(1)	023474	063777	INTX!PATH4!MAXX	:VECTOR LEFT FULL SCREEN
(1)	023476	012002	PATH2!2	:MOVE UP 2 UNITS
(1)	023500	043777	INTX!PATH0!MAXX	:VECTOR RIGHT FULL WIDTH
(1)	023502	012002	PATH2!2	:MOVE UP 2 UNITS
(1)	023504	063777	INTX!PATH4!MAXX	:VECTOR LEFT FULL SCREEN
(1)	023506	012002	PATH2!2	:MOVE UP 2 UNITS

(1)	023510	043777	INTX!PATH0!MAXX	:VECTOR RIGHT FULL WIDTH
(1)	023512	012002	PATH2!2	:MOVE UP 2 UNITS
(1)	023514	063777	INTX!PATH4!MAXX	:VECTOR LEFT FULL SCREEN
(1)	023516	012002	PATH2!2	:MOVE UP 2 UNITS
(1)	023520	043777	INTX!PATH0!MAXX	:VECTOR RIGHT FULL WIDTH
(1)	023522	012002	PATH2!2	:MOVE UP 2 UNITS
(1)	023524	063777	INTX!PATH4!MAXX	:VECTOR LEFT FULL SCREEN
(1)	023526	012002	PATH2!2	:MOVE UP 2 UNITS
(1)	023530	043777	INTX!PATH0!MAXX	:VECTOR RIGHT FULL WIDTH
(1)	023532	012002	PATH2!2	:MOVE UP 2 UNITS
(1)	023534	063777	INTX!PATH4!MAXX	:VECTOR LEFT FULL SCREEN
(1)	023536	012002	PATH2!2	:MOVE UP 2 UNITS
(1)	023540	043777	INTX!PATH0!MAXX	:VECTOR RIGHT FULL WIDTH
(1)	023542	012002	PATH2!2	:MOVE UP 2 UNITS
(1)	023544	063777	INTX!PATH4!MAXX	:VECTOR LEFT FULL SCREEN
(1)	023546	012002	PATH2!2	:MOVE UP 2 UNITS
(1)	023550	043777	INTX!PATH0!MAXX	:VECTOR RIGHT FULL WIDTH
(1)	023552	012002	PATH2!2	:MOVE UP 2 UNITS
(1)	023554	063777	INTX!PATH4!MAXX	:VECTOR LEFT FULL SCREEN
(1)	023556	012002	PATH2!2	:MOVE UP 2 UNITS
(1)	023560	043777	INTX!PATH0!MAXX	:VECTOR RIGHT FULL WIDTH
(1)	023562	012002	PATH2!2	:MOVE UP 2 UNITS
(1)	023564	063777	INTX!PATH4!MAXX	:VECTOR LEFT FULL SCREEN
(1)	023566	012002	PATH2!2	:MOVE UP 2 UNITS
(1)	023570	043777	INTX!PATH0!MAXX	:VECTOR RIGHT FULL WIDTH
(1)	023572	012002	PATH2!2	:MOVE UP 2 UNITS
(1)	023574	063777	INTX!PATH4!MAXX	:VECTOR LEFT FULL SCREEN
(1)	023576	012002	PATH2!2	:MOVE UP 2 UNITS
(1)	023600	043777	INTX!PATH0!MAXX	:VECTOR RIGHT FULL WIDTH
(1)	023602	012002	PATH2!2	:MOVE UP 2 UNITS
(1)	023604	063777	INTX!PATH4!MAXX	:VECTOR LEFT FULL SCREEN
(1)	023606	012002	PATH2!2	:MOVE UP 2 UNITS
(1)	023610	043777	INTX!PATH0!MAXX	:VECTOR RIGHT FULL WIDTH
(1)	023612	012002	PATH2!2	:MOVE UP 2 UNITS
(1)	023614	063777	INTX!PATH4!MAXX	:VECTOR LEFT FULL SCREEN
(1)	023616	012002	PATH2!2	:MOVE UP 2 UNITS
(1)	023620	043777	INTX!PATH0!MAXX	:VECTOR RIGHT FULL WIDTH
(1)	023622	012002	PATH2!2	:MOVE UP 2 UNITS
(1)	023624	063777	INTX!PATH4!MAXX	:VECTOR LEFT FULL SCREEN
(1)	023626	012002	PATH2!2	:MOVE UP 2 UNITS
(1)	023630	043777	INTX!PATH0!MAXX	:VECTOR RIGHT FULL WIDTH
(1)	023632	012002	PATH2!2	:MOVE UP 2 UNITS
(1)	023634	063777	INTX!PATH4!MAXX	:VECTOR LEFT FULL SCREEN
(1)	023636	012002	PATH2!2	:MOVE UP 2 UNITS
(1)	023640	043777	INTX!PATH0!MAXX	:VECTOR RIGHT FULL WIDTH
(1)	023642	012002	PATH2!2	:MOVE UP 2 UNITS
(1)	023644	063777	INTX!PATH4!MAXX	:VECTOR LEFT FULL SCREEN
(1)	023646	012002	PATH2!2	:MOVE UP 2 UNITS
(1)	023650	043777	INTX!PATH0!MAXX	:VECTOR RIGHT FULL WIDTH
(1)	023652	012002	PATH2!2	:MOVE UP 2 UNITS
(1)	023654	063777	INTX!PATH4!MAXX	:VECTOR LEFT FULL SCREEN
(1)	023656	012002	PATH2!2	:MOVE UP 2 UNITS
(1)	023660	043777	INTX!PATH0!MAXX	:VECTOR RIGHT FULL WIDTH
(1)	023662	012002	PATH2!2	:MOVE UP 2 UNITS
(1)	023664	063777	INTX!PATH4!MAXX	:VECTOR LEFT FULL SCREEN
(1)	023666	012002	PATH2!2	:MOVE UP 2 UNITS


```

(1) 023670 043777 INTX!PATH0!MAXX :VECTOR RIGHT FULL WIDTH
(1) 023672 012002 PATH2!2 :MOVE UP 2 UNITS
(1) 023674 063777 INTX!PATH4!MAXX :VECTOR LEFT FULL SCREEN
(1) 023676 012002 PATH2!2 :MOVE UP 2 UNITS
(1) 023700 043777 INTX!PATH0!MAXX :VECTOR RIGHT FULL WIDTH
(1) 023702 012002 PATH2!2 :MOVE UP 2 UNITS
(1) 023704 063777 INTX!PATH4!MAXX :VECTOR LEFT FULL SCREEN
(1) 023706 012002 PATH2!2 :MOVE UP 2 UNITS
3177 023710 173400 DSTOP
3178 023712 160000 DJMP
3179 023714 023446 DFI11C
3180
3181 .SBTTL MENU VERTICAL PHOSPHOR SUB-PICTURE
3182
3183 023716 114000 FRM11S: POINT
3184 023720 000000 0
3185 023722 000000 DELT11: 0
3186 023724 123600 FRM11D: BASICV!INT7
3192 023726 042177 INTX!PATH0!MAXMUX :VECTOR RIGHT FULL SCREEN IN MENU
(1) 023730 012002 PATH2!2 :MOVE UP 2 UNITS
(1) 023732 062177 INTX!PATH4!MAXMUX :VECTOR LEFT FULL MENU SCREEN
(1) 023734 012002 PATH2!2 :MOVE UP 2 UNITS
(1) 023736 042177 INTX!PATH0!MAXMUX :VECTOR RIGHT FULL SCREEN IN MENU
(1) 023740 012002 PATH2!2 :MOVE UP 2 UNITS
(1) 023742 062177 INTX!PATH4!MAXMUX :VECTOR LEFT FULL MENU SCREEN
(1) 023744 012002 PATH2!2 :MOVE UP 2 UNITS
(1) 023746 042177 INTX!PATH0!MAXMUX :VECTOR RIGHT FULL SCREEN IN MENU
(1) 023750 012002 PATH2!2 :MOVE UP 2 UNITS
(1) 023752 062177 INTX!PATH4!MAXMUX :VECTOR LEFT FULL MENU SCREEN
(1) 023754 012002 PATH2!2 :MOVE UP 2 UNITS
(1) 023756 042177 INTX!PATH0!MAXMUX :VECTOR RIGHT FULL SCREEN IN MENU
(1) 023760 012002 PATH2!2 :MOVE UP 2 UNITS
(1) 023762 062177 INTX!PATH4!MAXMUX :VECTOR LEFT FULL MENU SCREEN
(1) 023764 012002 PATH2!2 :MOVE UP 2 UNITS
(1) 023766 042177 INTX!PATH0!MAXMUX :VECTOR RIGHT FULL SCREEN IN MENU
(1) 023770 012002 PATH2!2 :MOVE UP 2 UNITS
(1) 023772 062177 INTX!PATH4!MAXMUX :VECTOR LEFT FULL MENU SCREEN
(1) 023774 012002 PATH2!2 :MOVE UP 2 UNITS
(1) 023776 042177 INTX!PATH0!MAXMUX :VECTOR RIGHT FULL SCREEN IN MENU
(1) 024000 012002 PATH2!2 :MOVE UP 2 UNITS
(1) 024002 062177 INTX!PATH4!MAXMUX :VECTOR LEFT FULL MENU SCREEN
(1) 024004 012002 PATH2!2 :MOVE UP 2 UNITS
(1) 024006 042177 INTX!PATH0!MAXMUX :VECTOR RIGHT FULL SCREEN IN MENU
(1) 024010 012002 PATH2!2 :MOVE UP 2 UNITS
(1) 024012 062177 INTX!PATH4!MAXMUX :VECTOR LEFT FULL MENU SCREEN
(1) 024014 012002 PATH2!2 :MOVE UP 2 UNITS
(1) 024016 042177 INTX!PATH0!MAXMUX :VECTOR RIGHT FULL SCREEN IN MENU
(1) 024020 012002 PATH2!2 :MOVE UP 2 UNITS
(1) 024022 062177 INTX!PATH4!MAXMUX :VECTOR LEFT FULL MENU SCREEN
(1) 024024 012002 PATH2!2 :MOVE UP 2 UNITS
(1) 024026 042177 INTX!PATH0!MAXMUX :VECTOR RIGHT FULL SCREEN IN MENU
(1) 024030 012002 PATH2!2 :MOVE UP 2 UNITS
(1) 024032 062177 INTX!PATH4!MAXMUX :VECTOR LEFT FULL MENU SCREEN
(1) 024034 012002 PATH2!2 :MOVE UP 2 UNITS
(1) 024036 042177 INTX!PATH0!MAXMUX :VECTOR RIGHT FULL SCREEN IN MENU
(1) 024040 012002 PATH2!2 :MOVE UP 2 UNITS

```


(1)	024042	062177	INTX!PATH4!MAXMUX	:VECTOR LEFT FULL MENU SCREEN
(1)	024044	012002	PATH2!2	:MOVE UP 2 UNITS
(1)	024046	042177	INTX!PATH0!MAXMUX	:VECTOR RIGHT FULL SCREEN IN MENU
(1)	024050	012002	PATH2!2	:MOVE UP 2 UNITS
(1)	024052	062177	INTX!PATH4!MAXMUX	:VECTOR LEFT FULL MENU SCREEN
(1)	024054	012002	PATH2!2	:MOVE UP 2 UNITS
(1)	024056	042177	INTX!PATH0!MAXMUX	:VECTOR RIGHT FULL SCREEN IN MENU
(1)	024060	012002	PATH2!2	:MOVE UP 2 UNITS
(1)	024062	062177	INTX!PATH4!MAXMUX	:VECTOR LEFT FULL MENU SCREEN
(1)	024064	012002	PATH2!2	:MOVE UP 2 UNITS
(1)	024066	042177	INTX!PATH0!MAXMUX	:VECTOR RIGHT FULL SCREEN IN MENU
(1)	024070	012002	PATH2!2	:MOVE UP 2 UNITS
(1)	024072	062177	INTX!PATH4!MAXMUX	:VECTOR LEFT FULL MENU SCREEN
(1)	024074	012002	PATH2!2	:MOVE UP 2 UNITS
(1)	024076	042177	INTX!PATH0!MAXMUX	:VECTOR RIGHT FULL SCREEN IN MENU
(1)	024100	012002	PATH2!2	:MOVE UP 2 UNITS
(1)	024102	062177	INTX!PATH4!MAXMUX	:VECTOR LEFT FULL MENU SCREEN
(1)	024104	012002	PATH2!2	:MOVE UP 2 UNITS
(1)	024106	042177	INTX!PATH0!MAXMUX	:VECTOR RIGHT FULL SCREEN IN MENU
(1)	024110	012002	PATH2!2	:MOVE UP 2 UNITS
(1)	024112	062177	INTX!PATH4!MAXMUX	:VECTOR LEFT FULL MENU SCREEN
(1)	024114	012002	PATH2!2	:MOVE UP 2 UNITS
(1)	024116	042177	INTX!PATH0!MAXMUX	:VECTOR RIGHT FULL SCREEN IN MENU
(1)	024120	012002	PATH2!2	:MOVE UP 2 UNITS
(1)	024122	062177	INTX!PATH4!MAXMUX	:VECTOR LEFT FULL MENU SCREEN
(1)	024124	012002	PATH2!2	:MOVE UP 2 UNITS
(1)	024126	042177	INTX!PATH0!MAXMUX	:VECTOR RIGHT FULL SCREEN IN MENU
(1)	024130	012002	PATH2!2	:MOVE UP 2 UNITS
(1)	024132	062177	INTX!PATH4!MAXMUX	:VECTOR LEFT FULL MENU SCREEN
(1)	024134	012002	PATH2!2	:MOVE UP 2 UNITS
(1)	024136	042177	INTX!PATH0!MAXMUX	:VECTOR RIGHT FULL SCREEN IN MENU
(1)	024140	012002	PATH2!2	:MOVE UP 2 UNITS
(1)	024142	062177	INTX!PATH4!MAXMUX	:VECTOR LEFT FULL MENU SCREEN
(1)	024144	012002	PATH2!2	:MOVE UP 2 UNITS
(1)	024146	042177	INTX!PATH0!MAXMUX	:VECTOR RIGHT FULL SCREEN IN MENU
(1)	024150	012002	PATH2!2	:MOVE UP 2 UNITS
(1)	024152	062177	INTX!PATH4!MAXMUX	:VECTOR LEFT FULL MENU SCREEN
(1)	024154	012002	PATH2!2	:MOVE UP 2 UNITS
(1)	024156	042177	INTX!PATH0!MAXMUX	:VECTOR RIGHT FULL SCREEN IN MENU
(1)	024160	012002	PATH2!2	:MOVE UP 2 UNITS
(1)	024162	062177	INTX!PATH4!MAXMUX	:VECTOR LEFT FULL MENU SCREEN
(1)	024164	012002	PATH2!2	:MOVE UP 2 UNITS
3193	024166	173400	DSTOP	
3194	024170	160000	DJMP	
3195	024172	023724	FRM11D	
3196				
3197	024174	117600	FRM10: POINT!INT7	
3198	024176	000000	0	
3199	024200	000000	0	
3200	024202	164700	CONSL1!BIT7!BIT6	:ENABLE CONSOLE #1
3201	024204	110000	LONGV	
3202	024206	041777	INTX!MAXX	
3203	024210	000000	0	
3204	024212	040000	INTX	
3205	024214	001777	MAXY	
3206	024216	061777	INTX!MINUSX!MAXX	


```
3207 024220 000000 0 *
3208 024222 040000 INTX
3209 024224 021777 MINUSX!MAXY
3210 024226 173400 DSTOP
3211 024230 160000 DJMP
3212 024232 024174 FRM10
3213
3214 024234 170003 FRM11M: DMENU1 ;ENABLE MENU
3215 024236 117600 POINT!INT7
3216 024240 000000 0
3217 024242 000000 0
3218 024244 110000 LONGV
3219 024246 040177 INTX!MAXMUX
3220 024250 000000 0
3221 024252 040000 INTX
3222 024254 001777 MAXY
3223 024256 060177 INTX!MINUSX!MAXMUX
3224 024260 000000 0
3225 024262 040000 INTX
3226 024264 021777 MINUSX!MAXY
3227 024266 173400 DSTOP
3228 024270 160000 DJMP
3229 024272 024234 FRM11M
3230
```



```
3232
3244 .SBTTL SHORT VECTOR AND RELATIVE POINT SUB-PICTURE
3245
3246
3247 024274 164700 FRME14: CONSL1!BIT7!BIT6 ;ENABLE CONSOLE #1
3248 024276 114000 POINT
3249 024300 000000 FRM14A: 0
3250 024302 000000 FRM14B: 0
3251 024304 104000 SHORTV
3252 024306 056200 INTX+16200
(1) 024310 056271 INTX+16200+71
(1) 024312 040071 INTX+71
(1) 024314 076271 INTX!MINUSX+16200+71
(1) 024316 076200 INTX!MINUSX+16200
(1) 024320 076371 INTX!MINUSX+16200+MINSUY+71
(1) 024322 040171 INTX+MINSUY+71
(1) 024324 056371 INTX+16200+MINSUY+71
(1) 024326 020504 20504
3253 024330 130000 RELATP
3254 024332 057000 INTX+17000
(1) 024334 057074 INTX+17000+74
(1) 024336 040074 INTX+74
(1) 024340 077074 INTX!MINUSX+17000+74
(1) 024342 077000 INTX!MINUSX+17000
(1) 024344 077174 INTX!MINUSX+17000+MINSUY+74
(1) 024346 040174 INTX+MINSUY+74
(1) 024350 057174 INTX+17000+MINSUY+74
(1) 024352 020504 20504
3255 024354 104000 SHORTV
3256 024356 057600 INTX+17600
(1) 024360 057677 INTX+17600+77
(1) 024362 040077 INTX+77
(1) 024364 077677 INTX!MINUSX+17600+77
(1) 024366 077600 INTX!MINUSX+17600
(1) 024370 077777 INTX!MINUSX+17600+MINSUY+77
(1) 024372 040177 INTX+MINSUY+77
(1) 024374 057777 INTX+17600+MINSUY+77
(1) 024376 020504 20504
3257 024400 173400 DSTOP
3258 024402 160000 DJMP
3259 024404 024274 FRME14
```

```

3261 .SBTTL GRAPHPLOT INCREMENT SUB-PICTURE
3262
3263 024406 114004 GRAPH: POINT!LINE0
3264 024410 000400 400
3265 024412 000200 200
3266 024414 164700 CONSL1!BIT7!BIT6 ;ENABLE CONSOLE #1
3267 024416 110000 LONGV ;DRAW BASE REF. VECTOR FOR GRAPH Y
3268 024420 041200 INTX+1200
3269 024422 000000 0
3270
3271 024424 114000 POINT
3272 024426 000440 440
3273 024430 000200 200
3274 024432 174104 GRPINC: STATSB!INCR+4 ;LOAD GRAPHPLOT INCR. REGISTER
3275 024434 124000 GRAPHY
3276 024436 162000 DJSR ;DJSR TO "SINE DATA"
3277 024440 024510 SINE
3278 024442 162000 DJSR ;DJSR TO SINE DATE
3279 024444 024510 SINE
3280
3281 024446 114000 POINT
3282 024450 000200 200
3283 024452 000040 40
3284 024454 110000 LONGV ;DRAW BASE REF. VECTOR FOR GRAPH X
3285 024456 040000 INTX
3286 024460 001200 1200
3287
3288 024462 114000 POINT
3289 024464 000200 200
3290 024466 000100 100
3291 024470 120000 GRAPHX
3292 024472 162000 DJSR ;DJSR TO "SINE DATA"
3293 024474 024510 SINE
3294 024476 162000 DJSR ;DJSR TO "SINE DATA"
3295 024500 024510 SINE
3296 024502 173400 DSTOP
3297 024504 160000 DJMP
3298 024506 024406 GRAPH
3299
3300 .SBTTL DATA STRING FOR A SINE WAVE
3301
3302 024510 000200 000205 000212 SINE: .WORD 0200,0205,0212,0217,0224,0231,0236,0243,0247,0253
3303 024534 000257 000262 000265 .WORD 0257,0262,0265,0270,0272,0274,0276,0277,0277,0277
3304 024560 000277 000276 000275 .WORD 0277,0276,0275,0274,0272,0267,0264,0261,0256,0252
3305 024604 000246 000241 000235 .WORD 0246,0241,0235,0230,0223,0216,0211,0203,0176,0171
3306 024630 000163 000156 000151 .WORD 0163,0156,0151,0144,0137,0133,0127,0123,0117,0114
3307 024654 000111 000106 000104 .WORD 0111,0106,0104,0102,0101,0100,0100,0100,0100,0101
3308 024700 000102 000104 000106 .WORD 0102,0104,0106,0111,0113,0117,0122,0126,0132,0137
3309 024724 000144 000151 000156 .WORD 0144,0151,0156,0163,0170,0175
3310 024740 166000 DPOP ;DISPLAY POP AND RESTORE
    
```



```

3312
3313 024742 164700
3314 024744 117000
3315 024746 000000
3316 024750 000000
3317 024752 173400
3318 024754 160000
3319 024756 024742
3320
3321 024760 114000
3322 024762 000000
3323 024764 000000
3324 024766 110000
3325 024770 040000
3326 024772 000000
3327 024774 173400
3328 024776 160000
3329 025000 024760
3330
3331
3332
3333 025002 164700
3334 025004 154024
3335 025006 114000
3336 025010 000000
3337 025012 000000
3338 025014 110000
3339 025016 040000
3340 025020 001777
3341 025022 041777
3342 025024 000000
3343 025026 040000
3344 025030 021777
3345 025032 061777
3346 025034 000000
3347 025036 114000
3348 025040 000040
3349 025042 000000
3350 025044 110000
3356 025046 060100
(1) 025050 000200
(1) 025052 040100
(1) 025054 000200
(1) 025056 060100
(1) 025060 000200
(1) 025062 040100
(1) 025064 000200
(1) 025066 060100
(1) 025070 000200
(1) 025072 040100
(1) 025074 000200
(1) 025076 060100
(1) 025100 000200
(1) 025102 040100
(1) 025104 000200
3357 025106 114000

```

.SBTTL SHORT TERM DRIFT SUB-PICTURE
 STDPIC: CONSL1!BIT7!BIT6 ;ENABLE CONSOLE #1
 POINT!INT4
 STDRA: 0
 STDRB: 0
 DSTOP
 DJMP
 STDPIC
 PICVTL: POINT
 PICVTA: 0
 PICVTB: 0
 LONGV
 PICVTC: INTX
 0
 PICVTE: DSTOP
 DJMP
 PICVTL
 .SBTTL SCREEN SCISSORING SUB-PICTURE
 PICSCS: CONSL1!BIT7!BIT6 ;ENABLE CONSOLE #1
 VCTR00!4
 POINT
 0
 0
 LONGV ;BOX
 INTX
 MAXY
 INTX!MAXX
 0
 INTX
 MINUSX!MAXY
 INTX!MINUSX!MAXX
 0
 POINT
 40
 0
 LONGV
 INTX!MINUSX!100
 200
 INTX!100
 200
 INTX!MINUSX!100
 200
 INTX!100
 200
 INTX!MINUSX!100
 200
 INTX!100
 200
 INTX!MINUSX!100
 200
 INTX!100
 200
 INTX!MINUSX!100
 200
 INTX!100
 200
 POINT

3358	025110	000000	0
3359	025112	001737	MAXY-40
3360	025114	110000	LONGV
3366	025116	040200	INTX!200
(1)	025120	000100	100
(1)	025122	040200	INTX!200
(1)	025124	020100	MINUSX!100
(1)	025126	040200	INTX!200
(1)	025130	000100	100
(1)	025132	040200	INTX!200
(1)	025134	020100	MINUSX!100
(1)	025136	040200	INTX!200
(1)	025140	000100	100
(1)	025142	040200	INTX!200
(1)	025144	020100	MINUSX!100
(1)	025146	040200	INTX!200
(1)	025150	000100	100
(1)	025152	040200	INTX!200
(1)	025154	020100	MINUSX!100
3367	025156	114000	POINT
3368	025160	001737	MAXX-40
3369	025162	001777	MAXY
3370	025164	110000	LONGV
3376	025166	040100	INTX!100
(1)	025170	020200	MINUSX!200
(1)	025172	060100	INTX!MINUSX!100
(1)	025174	020200	MINUSX!200
(1)	025176	040100	INTX!100
(1)	025200	020200	MINUSX!200
(1)	025202	060100	INTX!MINUSX!100
(1)	025204	020200	MINUSX!200
(1)	025206	040100	INTX!100
(1)	025210	020200	MINUSX!200
(1)	025212	060100	INTX!MINUSX!100
(1)	025214	020200	MINUSX!200
(1)	025216	040100	INTX!100
(1)	025220	020200	MINUSX!200
(1)	025222	060100	INTX!MINUSX!100
(1)	025224	020200	MINUSX!200


```

3378 025226 114000 POINT
3379 025230 001777 MAXX
3380 025232 000040 40
3381 025234 110000 LONGV
3387 025236 060200 INTX!MINUSX!200
(1) 025240 020100 MINUSX!100
(1) 025242 060200 INTX!MINUSX!200
(1) 025244 000100 100
(1) 025246 060200 INTX!MINUSX!200
(1) 025250 020100 MINUSX!100
(1) 025252 060200 INTX!MINUSX!200
(1) 025254 000100 100
(1) 025256 060200 INTX!MINUSX!200
(1) 025260 020100 MINUSX!100
(1) 025262 060200 INTX!MINUSX!200
(1) 025264 000100 100
(1) 025266 060200 INTX!MINUSX!200
(1) 025270 020100 MINUSX!100
(1) 025272 060200 INTX!MINUSX!200
(1) 025274 000100 100
3388 ;POSITION THE STARTING POINT OFF OF THE VIEWING SCRENE
3389 025276 114000 POINT
3390 025300 000777 MAXX/2
3391 025302 000000 0
3392 025304 110000 LONGV
3393 025306 000000 0
3394 025310 020200 MINUSX!200
3395 ;NOW DRAW AN DIAMOND THAT INTERSECTS EACH OF THE FOUR EDGES
3396 025312 110000 LONGV
3397 025314 041200 INTX!1200
3398 025316 001200 1200
3399 025320 061200 INTX!MINUSX!1200
3400 025322 001200 1200
3401 025324 061200 INTX!MINUSX!1200
3402 025326 021200 MINUSX!1200
3403 025330 041200 INTX!1200
3404 025332 021200 MINUSX!1200
3405
3406 .SBTTL VECTOR SCALE SUB-PICTURE
3407
3408 025334 154024 VCTR00!4
3409 025336 117600 POINT!INT7
3410 025340 000777 MAXX/2
3411 025342 000777 MAXY/2
3412 025344 154024 PICSCA: VCTR00!4
3413 025346 110000 LONGV
3414 025350 020150 MINUSX!150
3415 025352 020150 MINUSY!150
3416 025354 040320 INTX!320
3417 025356 000000 0
3418 025360 040000 INTX
3419 025362 000320 320
3420 025364 060320 INTX!MINUSX!320
3421 025366 000000 0
3422 025370 040000 INTX
3423 025372 020320 MINUSX!320
  
```

```

3424 025374 154024 VCTROO!4
3425 025376 113400 DSTOP
3426 025400 160000 DJMP
3427 025402 025002 PICSCS
3428 .SBTTL VECTOR STARTING SUB-PICTURE
3429
3430 025404 114000 VSTRT: POINT
3431 025406 001003 1003
3432 025410 001200 640.
3433 025412 110000 LONGV ;VECTOR 1
3434 025414 040000 INTX
3435 025416 000577 383.
3436 025420 114000 POINT
3437 025422 001003 1003
3438 025424 001400 768.
3439 025426 110000 LONGV ;VECTOR 2
3440 025430 040200 INTX!128.
3441 025432 000000 0
3442 025434 114000 POINT
3443 025436 001004 1004
3444 025440 001366 758.
3445 025442 110000 LONGV ;VECTOR 3
3446 025444 040177 INTX!127.
3447 025446 000000 0
3448 025450 114000 POINT
3449 025452 001003 1003
3450 025454 001400 768.
3451 025456 110000 LONGV ;VECTOR 4
3452 025460 060200 INTX!MINUSX!128.
3453 025462 000000 0
3454 025464 114000 POINT
3455 025466 001002 1002
3456 025470 001366 758.
3457 025472 110000 LONGV ;VECTOR 5
3458 025474 060177 INTX!MINUSX!127.
3459 025476 000000 0
3460 025500 114000 POINT
3461 025502 001003 1003
3462 025504 001200 640.
3463 025506 110000 LONGV ;VECTOR 6
3464 025510 040200 INTX!128.
3465 025512 000000 0
3466 025514 114000 POINT
3467 025516 001003 1003
3468 025520 001200 640.
3469 025522 110000 LONGV ;VECTOR 7
3470 025524 040000 INTX
3471 025526 020200 MINUSX!128.
3472 025530 114000 POINT
3473 025532 001003 1003
3474 025534 001200 640.
3475 025536 110000 LONGV ;VECTOR 8
3476 025540 060200 INTX!MINUSX!128.
3477 025542 000000 0
3478
3479

```


Line No.	ID	Offset	Sub-Picture	Minor Axis
3480			.SBTTL MAJOR AXIS OFFSET SUB-PICTURE	
3481			+X +Y	
3482	025544	114000	POINT	
3483	025546	001000	1000	
3484	025550	000400	400	
3485	025552	110000	LONGV	
3486	025554	040177	INTX!177	;X MINOR
3487	025556	000200	200	
3488	025560	114000	POINT	
3489	025562	001000	1000	
3490	025564	000400	400	
3491	025566	110000	LONGV	
3492	025570	040200	INTX!200	;Y MINOR
3493	025572	000177	177	
3494			+X -Y	
3495				
3496	025574	114000	POINT	
3497	025576	001000	1000	
3498	025600	000400	400	
3499	025602	110000	LONGV	
3500	025604	040177	INTX!177	;X MINOR
3501	025606	020200	MINUSY!200	
3502	025610	114000	POINT	
3503	025612	001000	1000	
3504	025614	000400	400	
3505	025616	110000	LONGV	
3506	025620	040200	INTX!200	;Y MINOR
3507	025622	020177	MINUSX!177	
3508				
3509			-X -Y	
3510	025624	114000	POINT	
3511	025626	001000	1000	
3512	025630	000400	400	
3513	025632	110000	LONGV	
3514	025634	060177	INTX!MINUSX!177	;X MINOR
3515	025636	020200	MINUSY!200	
3516	025640	114000	POINT	
3517	025642	001000	1000	
3518	025644	000400	400	
3519	025646	110000	LONGV	
3520	025650	060200	INTX!MINUSX!200	;Y MINOR
3521	025652	020177	MINUSX!177	
3522				
3523			-X +Y	
3524	025654	114000	POINT	
3525	025656	001000	1000	
3526	025660	000400	400	
3527	025662	110000	LONGV	
3528	025664	060177	INTX!MINUSX!177	;X MINOR
3529	025666	000200	200	
3530	025670	114000	POINT	
3531	025672	001000	1000	
3532	025674	000400	400	
3533	025676	110000	LONGV	
3534	025700	060200	INTX!MINUSX!200	
3535	025702	000177	177	

```

3536
3537 025704 173400          DSTOP
3538 025706 160000          DJMP
3539 025710 025404          VSTRT
3540
3541
3542          .SBTTL CHARACTER SCALE SUB-PICTURE
3543
3544          ;'A' CHARACTER
3545
3546 025712 164700          CHAQU:  CONSL1!BIT7!BIT6          ;ENABLE CONSOLE #1
3547 025714 114000          POINT
3548 025716 000700          700
3549 025720 001400          1400
3550 025722 110000          LONGV          ;DRAW REF. LINE
3551 025724 040400          INTX!400
3552 025726 000000          0
3553 025730 114000          POINT
3554 025732 000700          700
3555 025734 001400          1400
3556 025736 154340          CHARS3          ;CHAR SIZE 3 ( X2)
(1) 025740 100000          CHAR          ;CHARACTER MODE
(1) 025742 101          .BYTE 101
(1) 025743 000          .BYTE 0
(1) 025744 154300          CHARS2          ;CHAR SIZE 2 ( 1 1/2 X)
(1) 025746 100000          CHAR          ;CHAR MODE
(1) 025750 101          .BYTE 101
(1) 025751 000          .BYTE 0
(1) 025752 154240          CHARS1          ;CHAR SIZE 1 ( 1X)
(1) 025754 100000          CHAR
(1) 025756 101          .BYTE 101
(1) 025757 000          .BYTE 0
(1) 025760 154200          CHARSO          ;CHAR SIZE ( 1/2)
(1) 025762 100000          CHAR
(1) 025764 101          .BYTE 101
(1) 025765 000          .BYTE 0
3557
3558          ;'B' CHARACTER
3559
3560 025766 114000          POINT
3561 025770 000700          700
3562 025772 001200          1200
3563 025774 110000          LONGV          ;DRAW REF. LINE
3564 025776 040400          INTX!400
3565 026000 000000          0
3566 026002 114000          POINT
3567 026004 000700          700
3568 026006 001200          1200
3569 026010 154340          CHARS3          ;CHAR SIZE 3 ( X2)
(1) 026012 100000          CHAR          ;CHARACTER MODE
(1) 026014 102          .BYTE 102
(1) 026015 000          .BYTE 0
(1) 026016 154300          CHARS2          ;CHAR SIZE 2 ( 1 1/2 X)
(1) 026020 100000          CHAR          ;CHAR MODE
(1) 026022 102          .BYTE 102
(1) 026023 000          .BYTE 0
    
```


(1)	026024	154240	CHARS1		;CHAR SIZE 1 (1X)
(1)	026026	100000	CHAR		
(1)	026030	102	.BYTE	102	
(1)	026031	000	.BYTE	0	
(1)	026032	154200	CHARS0		;CHAR SIZE (1/2)
(1)	026034	100000	CHAR		
(1)	026036	102	.BYTE	102	
(1)	026037	000	.BYTE	0	
3570					
3571					
3572					
3573	026040	114000	POINT		
3574	026042	000700	700		
3575	026044	001000	1000		
3576	026046	110000	LONGV		;DRAW REF. LINE
3577	026050	040400	INTX!400		
3578	026052	000000	0		
3579	026054	114000	POINT		
3580	026056	000700	700		
3581	026060	001000	1000		
3582	026062	154340	CHARS3		;CHAR SIZE 3 (X2)
(1)	026064	100000	CHAR		;CHARACTER MODE
(1)	026066	106	.BYTE	106	
(1)	026067	000	.BYTE	0	
(1)	026070	154300	CHARS2		;CHAR SIZE 2 (1 1/2 X)
(1)	026072	100000	CHAR		;CHAR MODE
(1)	026074	106	.BYTE	106	
(1)	026075	000	.BYTE	0	
(1)	026076	154240	CHARS1		;CHAR SIZE 1 (1X)
(1)	026100	100000	CHAR		
(1)	026102	106	.BYTE	106	
(1)	026103	000	.BYTE	0	
(1)	026104	154200	CHARS0		;CHAR SIZE (1/2)
(1)	026106	100000	CHAR		
(1)	026110	106	.BYTE	106	
(1)	026111	000	.BYTE	0	
3583					
3584					
3585					
3586	026112	117000	POINT!INT4		
3587	026114	000700	700		
3588	026116	000600	600		
3589					
3590	026120	154340	CHARS3		;CHAR SIZE 3 (X2)
(1)	026122	100000	CHAR		;CHARACTER MODE
(1)	026124	117	.BYTE	117	
(1)	026125	000	.BYTE	0	
(1)	026126	154300	CHARS2		;CHAR SIZE 2 (1 1/2 X)
(1)	026130	100000	CHAR		;CHAR MODE
(1)	026132	117	.BYTE	117	
(1)	026133	000	.BYTE	0	
(1)	026134	154240	CHARS1		;CHAR SIZE 1 (1X)
(1)	026136	100000	CHAR		
(1)	026140	117	.BYTE	117	
(1)	026141	000	.BYTE	0	
(1)	026142	154200	CHARS0		;CHAR SIZE (1/2)

(1)	026144	100000	CHAR		
(1)	026146	117	.BYTE	117	
(1)	026147	000	.BYTE	0	
3591					
3592	026150	154024	VCTR00!4		;LOAD VECTOR SCALE TO NORMAL SIZE
3593	026152	114000	POINT		
3594	026154	000700	700		
3595	026156	000600	600		
3596	026160	154030	VCTR00!10		;LOAD 2X VECTOR SIZE
3597	026162	162000	DJSR		;DJSR TO DISPLAY SCALED POINTS AROUND THE '0'
3598	026164	026364	ORELPT		
3599	026166	154026	VCTR00!6		;LOAD VECTOR SCALE TO 1 1/2 SIZE
3600	026170	162000	DJSR		;DJSR TO DISPLAY SCALED POINTS
3601	026172	026364	ORELPT		
3602	026174	154024	VCTR00!4		;LOAD VECTOR SCALE TO 1 SIZE
3603	026176	162000	DJSR		;DJSR TO DISPLAY POINTS
3604	026200	026364	ORELPT		
3605	026202	154022	VCTR00!2		;LOAD VECTOR SCALE TO 1/2 SIZE
3606	026204	162000	DJSR		;DJSR TO DISPLAY RELATIVE POINTTS
3607	026206	026364	ORELPT		
3608	026210	154024	VCTR00!4		;RETURN TO NORMAL SIZE
3609	026212	164000	DNOP		
3610	026214	164000	DNOP		
3611	026216	164000	DNOP		
3612	026220	164000	DNOP		
3613	026222	164000	DNOP		
3614	026224	164000	DNOP		
3615	026226	164000	DNOP		
3616					
3617					
3618					
3619	026230	114000	POINT		
3620	026232	000700	700		
3621	026234	000400	400		
3622	026236	110000	LONGV		
3623	026240	040400	INTX!400		
3624	026242	000000	0		
3625	026244	114000	POINT		
3626	026246	000700	700		
3627	026250	000400	400		
3628	026252	154340	CHARS3		;CHAR SIZE 3 (X2)
(1)	026254	100000	CHAR		;CHARACTER MODE
(1)	026256	124	.BYTE	124	
(1)	026257	000	.BYTE	0	
(1)	026260	154300	CHARS2		;CHAR SIZE 2 (1 1/2 X)
(1)	026262	100000	CHAR		;CHAR MODE
(1)	026264	124	.BYTE	124	
(1)	026265	000	.BYTE	0	
(1)	026266	154240	CHARS1		;CHAR SIZE 1 (1X)
(1)	026270	100000	CHAR		
(1)	026272	124	.BYTE	124	
(1)	026273	000	.BYTE	0	
(1)	026274	154200	CHARS0		;CHAR SIZE (1/2)
(1)	026276	100000	CHAR		
(1)	026300	124	.BYTE	124	
(1)	026301	000	.BYTE	0	


```

3629 ;'X' CHARACTER
3630 026302 114000 POINT
3631 026304 000700 700
3632 026306 000200 200
3633 026310 110000 LONGV
3634 026312 040400 INTX!400
3635 026314 000000 0
3636 026316 114000 POINT
3637 026320 000700 700
3638 026322 000200 200
3639 026324 154340 CHARS3 ;CHAR SIZE 3 ( X2)
(1) 026326 100000 CHAR ;CHARACTER MODE
(1) 026330 130 .BYTE 130
(1) 026331 000 .BYTE 0
(1) 026332 154300 CHARS2 ;CHAR SIZE 2 ( 1 1/2 X)
(1) 026334 100000 CHAR ;CHAR MODE
(1) 026336 130 .BYTE 130
(1) 026337 000 .BYTE 0
(1) 026340 154240 CHARS1 ;CHAR SIZE 1 ( 1X)
(1) 026342 100000 CHAR
(1) 026344 130 .BYTE 130
(1) 026345 000 .BYTE 0
(1) 026346 154200 CHARSO ;CHAR SIZE ( 1/2)
(1) 026350 100000 CHAR
(1) 026352 130 .BYTE 130
(1) 026353 000 .BYTE 0
3640 026354 154240 CHARS1
3641 026356 173400 DSTOP
3642 026360 160000 DJMP
3643 026362 025712 CHAQU
3644
3645 026364 130000 ORELPT: RELATP ;ENABLE RELATIVE POINT MODE
3646 026366 041600 INTX!1600
3647 026370 040013 INTX!13
3648 026372 061600 INTX!MINUSX!1600
3649 026374 040113 INTX!113
3650 026376 003400 3400
3651 026400 166000 DPOP
3652
3653
3654 .SBTTL ROTATE CHARACTERS SUBPICTURE
3655
  
```


3657 026402 170003
 3658 026404 114000
 3659 026406 000000
 3660 026410 000000
 3661 026412 120000
 3662 026414 042177
 3663 026416 053777
 3664 026420 062177
 3665 026422 073777
 3666 026424 114000
 3667 026426 000050
 3668 026430 000000
 3669 026432 155400
 3676 026434 163005
 3677 026436 155000
 3678 026440 170002
 3679 026442 173400
 3680 026444 160000
 3681 026446 036656

ROTCHR: DMENU1 ;ENABLE MENU
 POINT
 0
 0
 BASICV ;DRAW REF. BOX
 INTX!PATH0!177
 INTX!PATH2!MAXY
 INTX!PATH4!177
 INTX!PATH6!MAXY
 POINT
 50
 0
 CHRRT1 ;ENABLE CHAR ROTATION
 DJSRR!WHERE2 ;DJSR RELATIVE TO THE TAG 'CHARQA'
 CHRRT0 ;DISABLE ROTATION
 DMENU0 ;RETURN TO MAIN SCREEN
 DSTOP
 DJMP ;JUMP BACK TO MAIN TEXT
 BUFFER

3682
 3683
 3684

;TWO COPIES OF THE 'QUICK BROWN FOX' MESSAGE

3685 026450 170040
 3686 026452 100000
 3692 026454 044124 020105 052521
 (2) 026531 124 042510 050440
 3693 026606 015 012
 3694 026610 170060
 3695 026612 044124 020105 052521
 (2) 026667 124 042510 050440
 3696 026744 015 012

CHARQA: STATSA!ITAL0 ;NON ITALIC
 CHAR
 .ASCII /THE QUICK BROWN FOX JUMPS OVER THE LAZY DOGS /
 .ASCII /THE QUICK BROWN FOX JUMPS OVER THE LAZY DOGS /
 .BYTE 15,12
 STATSA!ITAL1
 .ASCII /THE QUICK BROWN FOX JUMPS OVER THE LAZY DOGS /
 .ASCII /THE QUICK BROWN FOX JUMPS OVER THE LAZY DOGS /
 .BYTE 15,12

3697
 3698
 3699

;LOWER CASE ASCII MESSAGES

3700 026746 170040
 3701 026750 100000
 3710 026752 164 150 145
 (2) 026771 040 146 157
 (2) 027010 040 164 150
 (2) 027022 144 157 147
 (2) 027027 164 150 145
 (2) 027046 040 146 157
 (2) 027065 040 164 150
 (2) 027077 144 157 147
 3711 027104 015 012
 3712 027106 170060
 3713 027110 164 150 145
 (2) 027127 040 146 157
 (2) 027146 040 164 150
 (2) 027160 144 157 147
 (2) 027165 164 150 145
 (2) 027204 040 146 157
 (2) 027223 040 164 150
 (2) 027235 144 157 147
 3714 027242 015 012
 3715 027244 170040

CHARQD: STATSA!ITAL0
 CHAR
 .BYTE 164,150,145,40,161,165,151,143,153,40,142,162,157,167,156
 .BYTE 40,146,157,170,40,152,165,155,160,163,40,157,166,145,162
 .BYTE 40,164,150,145,40,154,141,172,171,40
 .BYTE 144,157,147,163,40
 .BYTE 164,150,145,40,161,165,151,143,153,40,142,162,157,167,156
 .BYTE 40,146,157,170,40,152,165,155,160,163,40,157,166,145,162
 .BYTE 40,164,150,145,40,154,141,172,171,40
 .BYTE 144,157,147,163,40
 .BYTE 15,12
 STATSA!ITAL1 ;SET ITALICS
 .BYTE 164,150,145,40,161,165,151,143,153,40,142,162,157,167,156
 .BYTE 40,146,157,170,40,152,165,155,160,163,40,157,166,145,162
 .BYTE 40,164,150,145,40,154,141,172,171,40
 .BYTE 144,157,147,163,40
 .BYTE 164,150,145,40,161,165,151,143,153,40,142,162,157,167,156
 .BYTE 40,146,157,170,40,152,165,155,160,163,40,157,166,145,162
 .BYTE 40,164,150,145,40,154,141,172,171,40
 .BYTE 144,157,147,163,40
 .BYTE 15,12
 STATSA!ITAL0

3716	027246	166000			DPOP
3717					
3718					.SBTTL
3719					.SBTTL LIGHT-PEN SUBPICTURE
3720					.SBTTL
3721					
3722					
3723					.SBTTL POSITION THE OCTAGON
3724	027250	164774			FRME16: CONSL1!BIT7!BIT6!BIT5!BIT4!BIT3!BIT2 ;ENABLE CONSOLE #1
3725	027252	164374			CONSLO!BIT7!BIT6!BIT5!BIT4!BIT3!BIT2
3726	027254	114140			POINT!LPON
3727	027256	001400			1400
3728	027260	001200			1200
3729	027262	110000			LONGV ;OCTOGON BY LENGTH OF 137
(1)	027264	040137			INTX+137
(1)	027266	000000			0
(1)	027270	040137			INTX+137
(1)	027272	000137			137
(1)	027274	040000			INTX
(1)	027276	000137			137
(1)	027300	060137			INTX!MINUSX+137
(1)	027302	000137			137
(1)	027304	060137			INTX!MINUSX+137
(1)	027306	000000			0
(1)	027310	060137			INTX!MINUSX+137
(1)	027312	020137			MINUSX+137
(1)	027314	040000			INTX
(1)	027316	020137			MINUSX+137
(1)	027320	040137			INTX+137
(1)	027322	020137			MINUSX+137
3730					.SBTTL DISPLAY ON CONSOLE #0 THE X-Y READOUT VALUE
3731	027324	164640			CONSL1!BIT7!BIT5 ;DISABLE CONSOLE #1
3732	027326	114000			POINT
3733	027330	001300			1300
3734	027332	001500			1500
3735	027334	100000			CHAR
3736	027336	036530			.ASCII /X=/
3737	027340	030061	030060		DLT14A: .ASCII /1000/
3738	027344	040	040	040	.BYTE 40,40,40
3739	027347	131	020075		.ASCII /Y= /
3740	027352	030061	030060		DLT14B: .ASCII /1000/
3741	027356	114000			POINT
3742	027360	001250			1250
3743	027362	001340			1340
3744	027364	100000			CHAR
3745	027366	160000			DJMP
3746	027370	027500			MSOPEN: PENOF0 ;JUMP TO PEN SWITCH MESSAGE FOR CONSOLE #0
3747					
3748					.SBTTL DISPLAY ON CONSOLE #1 THE X-Y READOUT VALUE
3749					
3750	027372	164760			LPRTA: CONSL1!BIT7!BIT6!BIT5!BIT4 ;ENABLE CONSOLE #1
3751	027374	164240			CONSLO!BIT7!BIT5 ;DISABLE CONSOLE #0
3752	027376	114000			POINT
3753	027400	001300			1300
3754	027402	001500			1500 ;POSITION THE X-Y MESSAGE
3755	027404	100000			CHAR

3756	027406	036530							
3757	027410	030061	030060			DLT14C:	.ASCII /X=/		
3758	027414	040	040	040			.ASCII /1000/		
3759	027417	131	020075				.BYTE 40,40,40		
3760	027422	030061	030060			DLT14D:	.ASCII /Y= /		
3761	027426	114000					.ASCII /1000/		
3762	027430	001250					POINT		
3763	027432	001340					1250		;POSITION THE PEN SWITCH MESSAGE FOR CONSOLE #1
3764	027434	100000					1340		
3765	027436	160000					CHAR		
3766	027440	027560				MS1PEN:	DJMP		;JUMP TO MESSAGE FOR #1
3767							PENOF1		
3768						.SBTTL	DISPLAY HIT COUNT MESSAGE		
3769	027442	117140				LPRTC:	POINT!INT4!LPON		
3770	027444	001300					1300		
3771	027446	000200					200		
3772	027450	164360					CONSL0!BIT7!BIT6!BIT5!BIT4		;ENABLE CONSOLE #0
3773	027452	100000					CHAR		
3774	027454	044510	020124	047503		FRM16B:	.ASCII /HIT COUNT = 0000/		
3775	027474	160000					DJMP		
3776	027476	027640					FRM16C		
3777									
3778	027500	042520	020116	053523		PENOF0:	.ASCII /PEN SWITCH #0 IS OFF/		
3779	027524	160000					DJMP		
3780	027526	027372					LPRTA		
3781	027530	042520	020116	053523		PENON0:	.ASCII /PEN SWITCH #0 IS ON /		
3782	027554	160000					DJMP		
3783	027556	027372					LPRTA		
3784	027560	042520	020116	053523		PENOF1:	.ASCII /PEN SWITCH #1 IS OFF/		
3785	027604	160000					DJMP		
3786	027606	027442					LPRTC		
3787	027610	042520	020116	053523		PENON1:	.ASCII /PEN SWITCH #1 IS ON /		
3788	027634	160000					DJMP		
3789	027636	027442					LPRTC		
3790									
3791						.SBTTL	HORIZONTAL REF. LINE SECTION		
3800									
3801	027640	114000				FRM16C:	POINT		
3802	027642	000000					0		
3803	027644	000700					700		
3804	027646	110000					LONGV		
3805	027650	041777					INTX!MAXX		
3806	027652	000000					0		
3807									
3808	027654	114000					POINT		;POINT TO X CORDINATE "0"
(1)	027656	000000					0		
(1)	027660	000640					640		;Y CORD. = 640
(1)	027662	110000					LONGV		;DRAW 30 UNIT VERTICAL LINE
(1)	027664	040000					INTX		
(1)	027666	000030					30		
3809	027670	114000					POINT		;POINT TO X CORDINATE "200"
(1)	027672	000200					200		
(1)	027674	000640					640		;Y CORD. = 640
(1)	027676	110000					LONGV		;DRAW 30 UNIT VERTICAL LINE
(1)	027700	040000					INTX		
(1)	027702	000030					30		

3810	027704	114000	POINT		;POINT TO X CORDINATE "400"
(1)	027706	000400	400		
(1)	027710	000640	640		;Y CORD. = 640
(1)	027712	110000	LONGV		;DRAW 30 UNIT VERTICAL LINE
(1)	027714	040000	INTX		
(1)	027716	000030	30		
3811	027720	114000	POINT		;POINT TO X CORDINATE "600"
(1)	027722	000600	600		
(1)	027724	000640	640		;Y CORD. = 640
(1)	027726	110000	LONGV		;DRAW 30 UNIT VERTICAL LINE
(1)	027730	040000	INTX		
(1)	027732	000030	30		
3812	027734	114000	POINT		;POINT TO X CORDINATE "1000"
(1)	027736	001000	1000		
(1)	027740	000640	640		;Y CORD. = 640
(1)	027742	110000	LONGV		;DRAW 30 UNIT VERTICAL LINE
(1)	027744	040000	INTX		
(1)	027746	000030	30		
3813	027750	114000	POINT		;POINT TO X CORDINATE "1200"
(1)	027752	001200	1200		
(1)	027754	000640	640		;Y CORD. = 640
(1)	027756	110000	LONGV		;DRAW 30 UNIT VERTICAL LINE
(1)	027760	040000	INTX		
(1)	027762	000030	30		
3814	027764	114000	POINT		;POINT TO X CORDINATE "1400"
(1)	027766	001400	1400		
(1)	027770	000640	640		;Y CORD. = 640
(1)	027772	110000	LONGV		;DRAW 30 UNIT VERTICAL LINE
(1)	027774	040000	INTX		
(1)	027776	000030	30		
3815	030000	114000	POINT		;POINT TO X CORDINATE "1600"
(1)	030002	001600	1600		
(1)	030004	000640	640		;Y CORD. = 640
(1)	030006	110000	LONGV		;DRAW 30 UNIT VERTICAL LINE
(1)	030010	040000	INTX		
(1)	030012	000030	30		
3816	030014	114000	POINT		;POINT TO X CORDINATE "1777"
(1)	030016	001777	1777		
(1)	030020	000640	640		;Y CORD. = 640
(1)	030022	110000	LONGV		;DRAW 30 UNIT VERTICAL LINE
(1)	030024	040000	INTX		
(1)	030026	000030	30		
3817					
3837			.SBTTL	VERTICAL SPACEING SECTION	
3838					
3839	030030	114000	POINT		
3840	030032	000200	200		
3841	030034	000010	10		
3842	030036	100000	CHAR		
3843	030040	020130	.ASCII	/X COORD = 200 /	
3844	030056	114000	POINT		
3845	030060	000200	200		
3846	030062	000060	60		
3847					
3848	030064	110000	LONGV		;DRAW LOWER LINE
(1)	030066	040200	INTX!200		

047503 051117

(1)	030070	000000	0	
(1)	030072	000000	0	
(1)	030074	000011	9.	
(1)	030076	060200	INTX!MINUSX!200	;DRAW NEXT HIGHER LINE
(1)	030100	000000	0	
(1)	030102	000000	0	
(1)	030104	000011	9.	
(1)	030106	040200	INTX!200	;DRAW NEXT HIGHER LINE
(1)	030110	000000	0	
(1)	030112	000000	0	
(1)	030114	000011	9.	
(1)	030116	060200	INTX!MINUSX!200	;DRAW UPPER LINE
(1)	030120	000000	0	
(1)	030122	000000	0	
(1)	030124	000040	40	;OFFSET FOR NEXT LINE
3849	030126	110000	LONGV	;DRAW LOWER LINE
(1)	030130	040200	INTX!200	
(1)	030132	000000	0	
(1)	030134	000000	0	
(1)	030136	000010	8.	
(1)	030140	060200	INTX!MINUSX!200	;DRAW NEXT HIGHER LINE
(1)	030142	000000	0	
(1)	030144	000000	0	
(1)	030146	000010	8.	
(1)	030150	040200	INTX!200	;DRAW NEXT HIGHER LINE
(1)	030152	000000	0	
(1)	030154	000000	0	
(1)	030156	000010	8.	
(1)	030160	060200	INTX!MINUSX!200	;DRAW UPPER LINE
(1)	030162	000000	0	
(1)	030164	000000	0	
(1)	030166	000040	40	;OFFSET FOR NEXT LINE
3850	030170	110000	LONGV	;DRAW LOWER LINE
(1)	030172	040200	INTX!200	
(1)	030174	000000	0	
(1)	030176	000000	0	
(1)	030200	000007	7	
(1)	030202	060200	INTX!MINUSX!200	;DRAW NEXT HIGHER LINE
(1)	030204	000000	0	
(1)	030206	000000	0	
(1)	030210	000007	7	
(1)	030212	040200	INTX!200	;DRAW NEXT HIGHER LINE
(1)	030214	000000	0	
(1)	030216	000000	0	
(1)	030220	000007	7	
(1)	030222	060200	INTX!MINUSX!200	;DRAW UPPER LINE
(1)	030224	000000	0	
(1)	030226	000000	0	
(1)	030230	000040	40	;OFFSET FOR NEXT LINE
3851	030232	110000	LONGV	;DRAW LOWER LINE
(1)	030234	040200	INTX!200	
(1)	030236	000000	0	
(1)	030240	000000	0	
(1)	030242	000006	6	
(1)	030244	060200	INTX!MINUSX!200	;DRAW NEXT HIGHER LINE
(1)	030246	000000	0	

(1)	030250	000000	0	
(1)	030252	000006	6	
(1)	030254	040200	INTX!200	;DRAW NEXT HIGHER LINE
(1)	030256	000000	0	
(1)	030260	000000	0	
(1)	030262	000006	6	
(1)	030264	060200	INTX!MINUSX!200	;DRAW UPPER LINE
(1)	030266	000000	0	
(1)	030270	000000	0	
(1)	030272	000040	40	;OFFSET FOR NEXT LINE
3852	030274	110000	LONGV	;DRAW LOWER LINE
(1)	030276	040200	INTX!200	
(1)	030300	000000	0	
(1)	030302	000000	0	
(1)	030304	000005	5	
(1)	030306	060200	INTX!MINUSX!200	;DRAW NEXT HIGHER LINE
(1)	030310	000000	0	
(1)	030312	000000	0	
(1)	030314	000005	5	
(1)	030316	040200	INTX!200	;DRAW NEXT HIGHER LINE
(1)	030320	000000	0	
(1)	030322	000000	0	
(1)	030324	000005	5	
(1)	030326	060200	INTX!MINUSX!200	;DRAW UPPER LINE
(1)	030330	000000	0	
(1)	030332	000000	0	
(1)	030334	000040	40	;OFFSET FOR NEXT LINE
3853	030336	110000	LONGV	;DRAW LOWER LINE
(1)	030340	040200	INTX!200	
(1)	030342	000000	0	
(1)	030344	000000	0	
(1)	030346	000004	4	
(1)	030350	060200	INTX!MINUSX!200	;DRAW NEXT HIGHER LINE
(1)	030352	000000	0	
(1)	030354	000000	0	
(1)	030356	000004	4	
(1)	030360	040200	INTX!200	;DRAW NEXT HIGHER LINE
(1)	030362	000000	0	
(1)	030364	000000	0	
(1)	030366	000004	4	
(1)	030370	060200	INTX!MINUSX!200	;DRAW UPPER LINE
(1)	030372	000000	0	
(1)	030374	000000	0	
(1)	030376	000040	40	;OFFSET FOR NEXT LINE
3854	030400	110000	LONGV	;DRAW LOWER LINE
(1)	030402	040200	INTX!200	
(1)	030404	000000	0	
(1)	030406	000000	0	
(1)	030410	000003	3	
(1)	030412	060200	INTX!MINUSX!200	;DRAW NEXT HIGHER LINE
(1)	030414	000000	0	
(1)	030416	000000	0	
(1)	030420	000003	3	
(1)	030422	040200	INTX!200	;DRAW NEXT HIGHER LINE
(1)	030424	000000	0	
(1)	030426	000000	0	

```

(1) 030430 000003 3
(1) 030432 060200 INTX!MINUSX!200 ;DRAW UPPER LINE
(1) 030434 000000 0
(1) 030436 000000 0
(1) 030440 000040 ;OFFSET FOR NEXT LINE
3855 030442 110000 LONGV ;DRAW LOWER LINE
(1) 030444 040200 INTX!200
(1) 030446 000000 0
(1) 030450 000000 0
(1) 030452 000002 2
(1) 030454 060200 INTX!MINUSX!200 ;DRAW NEXT HIGHER LINE
(1) 030456 000000 0
(1) 030460 000000 0
(1) 030462 000002 2
(1) 030464 040200 INTX!200 ;DRAW NEXT HIGHER LINE
(1) 030466 000000 0
(1) 030470 000000 0
(1) 030472 000002 2
(1) 030474 060200 INTX!MINUSX!200 ;DRAW UPPER LINE
(1) 030476 000000 0
(1) 030500 000000 0
(1) 030502 000040 ;OFFSET FOR NEXT LINE
3856 .SBTTL VARIABLE HORIZ. LINE LENGTH
3857
3858 030504 114000 POINT
3859 030506 001000 1000
3860 030510 000020 20
3861 030512 100000 CHAR
3862 030514 020130 047503 051117 .ASCII /X COORDINATE = 1000 /
3863 000001 L=1
3864 000600 M=600
3876 030540 114000 POINT ;POINT TO Y CORD. " M "
(1) 030542 001000 1000
(1) 030544 000600 M
(1) 030546 110000 LONGV ;DRAW A VECTOR " L " UNITS LONG
(1) 030550 040001 INTX! L
(1) 030552 000000 0
(1) 030554 114000 POINT ;POINT TO Y CORD. " M "
(1) 030556 001000 1000
(1) 030560 000560 M
(1) 030562 110000 LONGV ;DRAW A VECTOR " L " UNITS LONG
(1) 030564 040002 INTX! L
(1) 030566 000000 0
(1) 030570 114000 POINT ;POINT TO Y CORD. " M "
(1) 030572 001000 1000
(1) 030574 000540 M
(1) 030576 110000 LONGV ;DRAW A VECTOR " L " UNITS LONG
(1) 030600 040003 INTX! L
(1) 030602 000000 0
(1) 030604 114000 POINT ;POINT TO Y CORD. " M "
(1) 030606 001000 1000
(1) 030610 000520 M
(1) 030612 110000 LONGV ;DRAW A VECTOR " L " UNITS LONG
(1) 030614 040004 INTX! L
(1) 030616 000000 0
(1) 030620 114000 POINT ;POINT TO Y CORD. " M "
    
```


(1)	030622	001000	1000	
(1)	030624	000500	M	
(1)	030626	110000	LONGV	;DRAW A VECTOR " L " UNITS LONG
(1)	030630	040005	INTX! L	
(1)	030632	000000	0	
(1)	030634	114000	POINT	;POINT TO Y CORD. " M "
(1)	030636	001000	1000	
(1)	030640	000460	M	
(1)	030642	110000	LONGV	;DRAW A VECTOR " L " UNITS LONG
(1)	030644	040006	INTX! L	
(1)	030646	000000	0	
(1)	030650	114000	POINT	;POINT TO Y CORD. " M "
(1)	030652	001000	1000	
(1)	030654	000440	M	
(1)	030656	110000	LONGV	;DRAW A VECTOR " L " UNITS LONG
(1)	030660	040007	INTX! L	
(1)	030662	000000	0	
(1)	030664	114000	POINT	;POINT TO Y CORD. " M "
(1)	030666	001000	1000	
(1)	030670	000420	M	
(1)	030672	110000	LONGV	;DRAW A VECTOR " L " UNITS LONG
(1)	030674	040010	INTX! L	
(1)	030676	000000	0	
(1)	030700	114000	POINT	;POINT TO Y CORD. " M "
(1)	030702	001000	1000	
(1)	030704	000400	M	
(1)	030706	110000	LONGV	;DRAW A VECTOR " L " UNITS LONG
(1)	030710	040011	INTX! L	
(1)	030712	000000	0	
(1)	030714	114000	POINT	;POINT TO Y CORD. " M "
(1)	030716	001000	1000	
(1)	030720	000360	M	
(1)	030722	110000	LONGV	;DRAW A VECTOR " L " UNITS LONG
(1)	030724	040012	INTX! L	
(1)	030726	000000	0	
(1)	030730	114000	POINT	;POINT TO Y CORD. " M "
(1)	030732	001000	1000	
(1)	030734	000340	M	
(1)	030736	110000	LONGV	;DRAW A VECTOR " L " UNITS LONG
(1)	030740	040013	INTX! L	
(1)	030742	000000	0	
(1)	030744	114000	POINT	;POINT TO Y CORD. " M "
(1)	030746	001000	1000	
(1)	030750	000320	M	
(1)	030752	110000	LONGV	;DRAW A VECTOR " L " UNITS LONG
(1)	030754	040014	INTX! L	
(1)	030756	000000	0	
(1)	030760	114000	POINT	;POINT TO Y CORD. " M "
(1)	030762	001000	1000	
(1)	030764	000300	M	
(1)	030766	110000	LONGV	;DRAW A VECTOR " L " UNITS LONG
(1)	030770	040015	INTX! L	
(1)	030772	000000	0	
(1)	030774	114000	POINT	;POINT TO Y CORD. " M "
(1)	030776	001000	1000	
(1)	031000	000260	M	

3894					
3895		001000		J=1000	
3896	031172	114000		POINT	
3897	031174	000020		20	
3898	031176	001700		1700	
3899	031200	103600		CHAR!INT7	;CHAR MODE
3900	031202	036531	033461 030060	.ASCII /Y=1700/	
3901	031210	114000		POINT	
3902	031212	000200		200	
3903	031214	001700		1700	
3904	031216	110000		LONGV	;DRAW HORIZ. LINE
3905	031220	040600		INTX!600	
3906	031222	000000		0	
3907	031224	130000		RELATP	
3908	031226	057600		57600	
3909	031230	114000		POINT	
3910	031232	000020		20	
3911	031234	001600		1600	
3912	031236	103400		CHAR!INT6	;CHAR MODE
3913	031240	036531	033061 030060	.ASCII /Y=1600/	
3914	031246	114000		POINT	
3915	031250	000200		200	
3916	031252	001600		1600	
3917	031254	110000		LONGV	;DRAW HORIZ. LINE
3918	031256	040600		INTX!600	
3919	031260	000000		0	
3920	031262	130000		RELATP	
3921	031264	057600		57600	
3922	031266	114000		POINT	
3923	031270	000020		20	
3924	031272	001500		1500	
3925	031274	103200		CHAR!INT5	;CHAR MODE
3926	031276	036531	032461 030060	.ASCII /Y=1500/	
3927	031304	114000		POINT	
3928	031306	000200		200	
3929	031310	001500		1500	
3930	031312	110000		LONGV	;DRAW HORIZ. LINE
3931	031314	040600		INTX!600	
3932	031316	000000		0	
3933	031320	130000		RELATP	
3934	031322	057600		57600	
3935	031324	114000		POINT	
3936	031326	000020		20	
3937	031330	001400		1400	
3938	031332	103000		CHAR!INT4	;CHAR MODE
3939	031334	036531	032061 030060	.ASCII /Y=1400/	
3940	031342	114000		POINT	
3941	031344	000200		200	
3942	031346	001400		1400	
3943	031350	110000		LONGV	;DRAW HORIZ. LINE
3944	031352	040600		INTX!600	
3945	031354	000000		0	
3946	031356	130000		RELATP	
3947	031360	057600		57600	
3948	031362	114000		POINT	
3949	031364	000020		20	

3950	031366	001300			1300	
3951	031370	102600			CHAR!INT3	;CHAR MODE
3952	031372	036531	031461	030060	.ASCII /Y=1300/	
3953	031400	114000			POINT	
3954	031402	000200			200	
3955	031404	001300			1300	
3956	031406	110000			LONGV	;DRAW HORIZ. LINE
3957	031410	040600			INTX!600	
3958	031412	000000			0	
3959	031414	130000			RELATP	
3960	031416	057600			57600	
3961	031420	114000			POINT	
3962	031422	000020			20	
3963	031424	001200			1200	
3964	031426	102400			CHAR!INT2	;CHAR MODE
3965	031430	036531	031061	030060	.ASCII /Y=1200/	
3966	031436	114000			POINT	
3967	031440	000200			200	
3968	031442	001200			1200	
3969	031444	110000			LONGV	;DRAW HORIZ. LINE
3970	031446	040600			INTX!600	
3971	031450	000000			0	
3972	031452	130000			RELATP	
3973	031454	057600			57600	
3974	031456	114000			POINT	
3975	031460	000020			20	
3976	031462	001100			1100	
3977	031464	102200			CHAR!INT1	;CHAR MODE
3978	031466	036531	030461	030060	.ASCII /Y=1100/	
3979	031474	114000			POINT	
3980	031476	000200			200	
3981	031500	001100			1100	
3982	031502	110000			LONGV	;DRAW HORIZ. LINE
3983	031504	040600			INTX!600	
3984	031506	000000			0	
3985	031510	130000			RELATP	
3986	031512	057600			57600	
3987	031514	114000			POINT	
3988	031516	000020			20	
3989	031520	001000			1000	
3990	031522	102000			CHAR!INT0	;CHAR MODE
3991	031524	036531	030061	030060	.ASCII /Y=1000/	
3992	031532	114000			POINT	
3993	031534	000200			200	
3994	031536	001000			1000	
3995	031540	110000			LONGV	;DRAW HORIZ. LINE
3996	031542	040600			INTX!600	
3997	031544	000000			0	
3998	031546	130000			RELATP	
3999	031550	057600			57600	
4000						
4001					.SBTTL	DRAW OUTER REFERENCE BOX
4002						
4003	031552	117000			POINT!INT4	
4004	031554	000000			0	
4005	031556	000000			0	


```

4006 031560 110000 LONGV
4007 031562 041777 INTX!MAXX
4008 031564 000000 0
4009 031566 040000 INTX
4010 031570 001777 MAXY
4011 031572 061777 INTX!MINUSX!MAXX
4012 031574 000000 0
4013 031576 040000 INTX
4014 031600 021777 MINUSX!MAXY
4015 031602 173400 DSTOP
4016 031604 160000 DJMP
4017 031606 027250 FRME16
4018
4019
4020 .SBTTL
      .SBTTL KEYBOARD CHARACTER ECHO SUB-PICTURE
ECHOFR: POINT
4021 031610 114000 0
4022 031612 000000 MAXY-200
4023 031614 001577 STATSA!SYNC40 ;ENABLE SYNC
4024 031616 170010 CHARS1 ;ENABLE NORMAL CHAR. SIZE
4025 031620 154240 CHAR
4026 031622 100000 .BYTE 17,17
4027 031624 017 017 .ASCII /KEYBOARD CHARACTER ECHO LOOP/<15><12>
4028 031626 042513 041131 040517 .ASCII / CTRL C TO EXIT LOOP/<15><12><12>
4029 031664 020040 052103 046122 .ASCII /CHARACTER CODE IS = /
4030 031714 044103 051101 041501 .BYTE 0,0,0,0 ;OCTAL VALUE CODE IS LOADED HERE
4031 031742 000 000 000 ECODEV: .BYTE 15,12
4032 031746 015 012 ECHJMP: DJMPR!10 ;BR OVER IF NOT "SHIFTOUT" MODE
4033 031750 161010 CHAR
4034 031752 100000 .ASCII /SHIFT-OUT MODE/
4035 031754 044123 043111 026524 .BYTE 15,12
4036 031772 015 012 DJMP
4037 031774 160000 BUFFER
4038 031776 036656
4039
4040 .SBTTL
4041 .SBTTL DYNAMIC EXT. STOP FRAME
4042 .SBTTL
4043
4044 ;DISPLAY A BOX AROUND THE SCREEN
4045 ; EACH LINE IS A DIFFERENT LINE TYPE AND INTENSITY LEVEL
4046
4047 032000 164300 FRME17: CONSLO!BIT7!BIT6 ;ENABLE CONSOLE #0
4048 032002 164700 CONSL1!BIT7!BIT6 ;ENABLE CONSOLE #1
4049 032004 114000 POINT ;POINT
4050 032006 000000 0
4051 032010 001777 MAXY ;TO TOP LEFT CORNOR
4052
4053 032012 150001 DNAME!BIT0 ;LOAD NAME REG. WITH #1
4054
4055 032014 113407 LONGV!INT6!LINE3 ;LONG VECTOR WITH INTENS. 6 AND LINE TYPE 3
4056 032016 041777 INTX!MAXX
4057 032020 000000 0
4058
4059 032022 113006 LONGV!INT4!LINE2 ;LONG VECTOR WITH INTENS. 4 AND LINE TYPE 2
4060 032024 040000 INTX
4061 032026 021777 MINUSX!MAXY
  
```

4062
4063 032030 112405
4064 032032 061777
4065 032034 000000
4066
4067 032036 113604
4068 032040 040000
4069 032042 001777
4070

LONGV!INT2!LINE1
INTX!MINUSX!MAXX
0
LONGV!INT7!LINE0
INTX
MAXY

;LONG VECTOR WITH INTENS. 2 AND LINE TYPE 1
;LONG VECTOR WITH INTENS. 7 AND LINE TYPE 0


```

4072 ;DISPLAY A DIAMOND -- WITH SHORT VECTORS AND DIFFERENT INTENSITY LEVELS
4073
4074 032044 150004 DNAME!BIT2 ;LOAD NAME REG. WITH BIT 2
4075 032046 114000 POINT
4076 032050 001000 1000
4077 032052 001500 1500
4078 032054 106200 SHORTV!INT1
4079 032056 057677 57677 ;+X +Y
4080 032060 106600 SHORTV!INT3
4081 032062 077677 77677 ;+X -Y
4082 032064 107200 SHORTV!INT5
4083 032066 077777 77777 ;-X -Y
4084 032070 107600 SHORTV!INT7
4085 032072 057777 57777 ;-X +Y
4086
4087 ;DISPLAY FOUR BLINKING POINTS -- WITH RELATIVE POINT AND BLINK ENABLED
4088
4089 032074 150010 DNAME!BIT3 ;LOAD NAME REG. WITH #10
4090 032076 114000 POINT
4091 032100 001000 1000
4092 032102 000700 700
4093 032104 133030 RELATP!INT4!BLKON ;RELATIVE POINT AND BLINK ON
4094 032106 057677 57677 ;+X +Y
4095 032110 077677 77677 ;+X -Y
4096 032112 077777 77777 ;-X -Y
4097 032114 057777 57777 ;-X +Y
4098
4099 ;DISPLAY FIVE GRAPH PLOT X DATA POINTS
4100
4101 032116 150020 DNAME!BIT4 ;LOAD NAME REG. WITH BIT4
4102 032120 174110 STATSB!INCR+10 ;LOAD GRAPH INCREMENT
4103 032122 114020 POINT!BLKOFF
4104 032124 001000 1000
4105 032126 001600 1600
4106
4107 032130 120000 GRAPHX
4108 032132 001500 001510 001520 1500, 1510, 1520, 1530, 1540
4109
4110 ;DISPLAY FIVE GRAPH PLOT Y DATA POINTS
4111
4112 032144 150040 DNAME!BIT5 ;LOAD NAME REG. WITH BIT5
4113 032146 114000 POINT
4114 032150 001540 1540
4115 032152 001200 1200
4116
4117 032154 124000 GRAPHY
4118 032156 001640 001630 001620 1640, 1630, 1620, 1610
4119
  
```



```
4121 ;DISPLAY AN OCTOGON -- USING BASIC VECTOR'S
4122
4123 032166 150060 DNAME!BIT5!BIT4 ;LOAD NAME REG. WITH #60
4124 032170 114000 POINT
4125 032172 001540 1540
4126 032174 000640 640
4127
4128 032176 120000 BASICV
4129 032200 042100 INTX!PATH0!100
4130 032202 046100 INTX!PATH1!100
4131 032204 052100 INTX!PATH2!100
4132 032206 056100 INTX!PATH3!100
4133 032210 062100 INTX!PATH4!100
4134 032212 066100 INTX!PATH5!100
4135 032214 072100 INTX!PATH6!100
4136 032216 076100 INTX!PATH7!100
4137
4138 ;DISPLAY A LARGE SQUARE IN THE CENTER -- USING ABSOLUTE VECTORS
4139
4140 032220 150100 DNAME!BIT6 ;LOAD NAME REG. WITH BIT6
4141 032222 114000 POINT
4142 032224 000400 400
4143 032226 000400 400
4144
4145 032230 144000 ABSVCT
4146 032232 041400 INTX!1400
4147 032234 000400 400
4148
4149 032236 041400 INTX!1400
4150 032240 001400 1400
4151
4152 032242 040400 INTX!400
4153 032244 001400 1400
4154
4155 032246 040400 INTX!400
4156 032250 000400 400
4157
```



```

4159 ;NOW USE CHAR MODE AND DISP. JSR'S, CHAR. ROTATE, CHAR ITALICS
4160
4161 032252 150400 DNAME!BIT8 ;LOAD NAME REG. WITH BIT8
4162 032254 170060 STATSA!ITAL1 ;ITALICS ON
4163 032256 155400 CHRRT1 ;CHAR. ROTATE ON
4164
4165 032260 162000 DJSR ;ABSOLUTE JSR TO CHAR. FRAME
4166 032262 032276 SHOWCH
4167
4168 ;NOW USE CHAR MODE, DISP. JSR'S
4169
4170 032264 151000 DNAME!BIT9 ;LOAD NAME REG. WITH BIT9
4171 032266 170040 STATSA!ITAL0 ;ITALICS OFF
4172 032270 155000 CHRRT0 ;CHAR. ROTATE OFF
4173
4174 032272 163001 DJSRR!1 ;RELATIVE DJSR TO CHAR. FRAME
4175 032274 161036 DJMPR!36 ;RELATIVE DJUMP OVER CHAR. SUBROUTINE
4176
4177 032276 114000 SHOWCH: POINT
4178 032300 000200 200
4179 032302 000200 200
4180
4181 032304 154200 CHARSO ;SET CHAR. SIZE TO 00
4182 032306 100000 CHAR
4183 032310 020040 027060 020065 .ASCII " 0.5 SIZE"
4184
4185 032322 154240 CHAR1 ;SET CHAR. SIZE TO 01
4186 032324 020040 027061 020060 .ASCII " 1.0 SIZE"
4187
4188 032336 154300 CHAR2 ;SET CHAR. SIZE TO 10
4189 032340 020040 027061 020065 .ASCII " 1.5 SIZE"
4190
4191 032352 154340 CHAR3 ;SET CHAR. SIZE TO 11
4192 032354 020040 027062 020060 .ASCII " 2.0 SIZE"
4193 032366 154240 CHAR1 ;RESET CHAR. SIZE TO NORMAL
4194 032370 166000 DPOP ;EXIT
4195
  
```

```
4197 ;DISPLAY A RECTANGLE IN THE MENU AREA -- USE DIFFERENT VECTOR SCALES
4198
4199 032372 151400 $FILE2: DNAME!BIT9!BIT8 ;LOAD NAME REG. WITH #1400
4200 032374 170003 DMENU1 ;ENABLE THE MENU AREA
4201 032376 114000 POINT
4202 032400 000000 0
4203 032402 000040 40
4204
4205 032404 154037 VCTR00!17 ;LOAD VECTOR SCALE
4206 032406 110000 LONGV
4207 032410 040000 INTX ;DRAW VERT. LINE
4208 032412 000400 400
4209
4210 032414 154021 VCTR00!1 ;LOAD VECTOR SCALE
4211 032416 040700 INTX!700
4212 032420 000000 0
4213
4214 032422 154037 VCTR00!17 ;LOAD VECTOR SCALE
4215 032424 040000 INTX
4216 032426 020400 MINUSX!400 ;DRAW VERT. LINE
4217
4218 032430 154021 VCTR00!1 ;LOAD VECTOR SCALE
4219 032432 060700 INTX!MINUSX!700
4220 032434 000000 0
4221
4222 032436 170040 STATSA!ITALO ;DISABLE ITALICS
4223 032440 154024 VCTR00!4 ;RETURN TO NORMAL SCALE
4224 032442 170002 DMENU0 ;EXIT MENU AREA
4225
4226 032444 173400 DSTOP
4227
4228 032446 160000 DJMP ;JUMP TO START OF FILE
4229 032450 032000 FRM17F: FRME17
4230
4231
4232 032452 164700 FRM17E: CONSL1!BIT7!BIT6 ;ENABLE CONSOLE #1
4233 032454 117030 POINT!INT4!BLKON
4234 032456 000000 0
4235 032460 001000 1000
4236 032462 170040 STATSA!ITALO ;ITALICS OFF
4237 032464 155000 CHRRT0 ;CHAR. ROT. OFF
4238 032466 154340 CHARS3
4239
4240 032470 100000 CHAR
4241 032472 162000 DJSR ;JSR TO ASCII ERROR MESSAGE
4242 032474 032506 WHY: WHY0 ;ADDRESS OF ERROR TYPE
4243 032476 100020 CHAR!BLKOFF
4244 032500 173400 DSTOP
4245 032502 160000 DJMP
4246 032504 032452 FRM17E
4247
```


4249	032506	047516	042440	052130	WHY0:	.ASCII /NO EXTERNAL STOP INTERRUPT/ DPOP
4250	032540	166000				
4251						
4252	032542	047125	054105	042520	WHY1:	.ASCII /UNEXPECTED INTERRUPT TO VECTOR +4 / DPOP
4253	032604	166000				
4254						
4255	032606	047125	054105	042520	WHY2:	.ASCII /UNEXPECTED INTERRUPT TO VECTOR +10/ DPOP
4256	032650	166000				
4257						
4258	032652	047125	054105	042520	WHY3:	.ASCII /UNEXPECTED INTERRUPT TO VECTOR +14/ DPOP
4259	032714	166000				
4260						
4261	032716	027104	027120	027103	WHY4:	.ASCII /D.P.C. TOO LOW/ DPOP
4262	032734	166000				
4263						
4264	032736	027104	027120	027103	WHY5:	.ASCII /D.P.C. TOO HIGH / DPOP
4265	032756	166000				
4266						
4267	032760	052123	050117	044440	WHY6:	.ASCII /STOP INTERRUPT BUT NO STOP FLAGS/ DPOP
4268	033020	166000				

```
4270 .SBTTL POINT CORRELATOR DISPLAY FILES.
4271 ;
4272 ; FIRST SOME RE-DEFINITION, AND A COUPLE OF MACROS.
4273 ;
4274 164003 PCSET= CONSLO!3 ; VT X/Y TO POINT CORRELATOR.
4275 164002 PCRUN= CONSLO!2 ; CORRELATOR RUN.
4276 164060 PCHIT= CONSLO!60 ; INTERRUPT ON CORRELATOR HIT.
4277 164040 PCNOHIT= CONSLO!40 ; DON'T
4278 170200 PCDIM= STATSA!200 ; DIM POINT OF IMPACT.
4279 170300 PCNODIM= STATSA!300 ; DON'T
4280 ;
4281 175000 APPERO= STATSB!1000 ; APPERTURE SIZE 0 (MIN)
4282 175200 APPER1= APPERO!0200 ;
4283 175400 APPER2= APPERO!0400 ;
4284 175600 APPER3= APPERO!0600 ; APPERTURE SIZE 3 (MAX).
4285 ;
4286 161000 DSKP= DJMPR ; DISPLAY SKIPS
4287 161001 DSKP1= DSKP+1
4288 161002 DSKP2= DSKP+2
4289 161003 DSKP3= DSKP+3
4290 161004 DSKP4= DSKP+4
4291 ;
4292 .LIST MD
4293 .MACRO EVNTEXT TEXT ; MAKE AN 'EVEN' TEXT STRING.
4294 .ASCII TEXT
4295 .IIF NZ .81, .ASCII <0>
4296 .ENDM EVNTEXT
4297 ;
4298 .MACRO OCTAGON RAD ; RAD MUST BE A MULTIPLE OF 5 (DECIMAL).
4299 LV= <RAD/5>*4 ; BUILD A 4:3 OCTAGON...
4300 SV= <RAD/5>*3 ; ...OF RADIUS RAD !
4301 BASICV
4302 PATH6!RAD
4303 PATH4!<LV/2>
4304 INTX!PATH0!LV
4305 INTX!PATH1!SV
4306 INTX!PATH2!LV
4307 INTX!PATH3!SV
4308 INTX!PATH4!LV
4309 INTX!PATH5!SV
4310 INTX!PATH6!LV
4311 INTX!PATH7!SV
4312 PATH0!<LV/2>
4313 PATH2!RAD
4314 .ENDM OCTAGON
4315 .NLIST MD
```



```

4317
4318
4319
4320
4321
4322 033022 175000
4323 033024 164000
4324 033026 114100
4325 033030 000000 000000
4326 033034 164003
4327 033036 164000 164000
4328 033042 164002
4329
4330 033044 170002 164060
4331 033050 116600
4332 033052 040002 000002
4333 033056 040002 000400
4334 033062 040002 001000
4335 033066 040002 001400
4336 033072 040002 001775
4337 033076 040400 000002
4338 033102 040400 000400
4339 033106 040400 001000
4340 033112 040400 001400
4341 033116 040400 001775
4342 033122 041000 000002
4343 033126 041000 000400
4344 033132 041000 001000
4345 033136 041000 001400
4346 033142 041000 001775
4347 033146 041400 000002
4348 033152 041400 000400
4349 033156 041400 001000
4350 033162 041400 001400
4351 033166 041400 001775
4352 033172 041775 000002
4353 033176 041775 000400
4354 033202 041775 001000
4355 033206 041775 001400
4356 033212 041775 001775
4357 033216 170003 164000
4358 033222 040300 000002
4359 033226 040300 000400
4360 033232 040300 001000
4361 033236 040300 001400
4362 033242 040300 001775
4363 033246 170002 164040
4364 033252 162000 033650
4365 033256 162000 033446
4366 033262 162000 035336
4367 033266 173400
4368 033270 160000 033022

```

```

:
: FRAME 1. OVERALL ALIGNMENT VERIFICATION FRAME.
: DISPLAY A 6 X 5 DOT ARRAY.
: CORRELATOR SHOULD "SEE" ALL THE DOTS.
:
PIX1:  APPERO                ; INIT APPERTURE SIZE.
      DNOP                  ; OR DMENU ON !!
      POINT!LPOFF
PIX1X:  0,0                  ; X/Y SET BY CALLER.
      PCSET                 ; SET CORRELATOR X/Y...
      DNOP, DNOP
      PCRUN                 ; START CORRELATOR.
:
PIX1P:  DMENU0, PCHIT        ; MENU OFF, HITS ENABLED.
      POINT!INT3
      INTX!2,2
      INTX!2,400
      INTX!2,1000
      INTX!2,1400
      INTX!2,1775
      INTX!400,2
      INTX!400,400
      INTX!400,1000
      INTX!400,1400
      INTX!400,1775
      INTX!1000,2
      INTX!1000,400
      INTX!1000,1000
      INTX!1000,1400
      INTX!1000,1775
      INTX!1400,2
      INTX!1400,400
      INTX!1400,1000
      INTX!1400,1400
      INTX!1400,1775
      INTX!1775,2
      INTX!1775,400
      INTX!1775,1000
      INTX!1775,1400
      INTX!1775,1775
PIX1M:  DMENU1, DNOP        ; NOW MENU ON...
      INTX!300,2            ; ...FOR 5 Y POINTS.
      INTX!300,400
      INTX!300,1000
      INTX!300,1400
      INTX!300,1775
      DMENU0, PCNOHIT     ; MENU OFF, HITS OFF.
      DJSR, INSTR1       ; INSTRUCTIONS...
      DJSR, GONOGO       ; ...AND RESULTS.
      DJSR, PCMENU       ; REFRESH MENU
      DSTOP
      DJMP, PIX1         ; LOOP.

```

```

4370
4371
4372
4373 033274 175000
4374 033276 114100
4375 033300 000200 001000
4376 033304 164003
4377 033306 164000 164000
4378 033312 164002
4379 033314 162000 033416
4380 033320 164060
4381 033322 130000
4382 033324 162000 036656
4383 033330 164040
4384 033332 164000
4385 033334 162000 034020
4386 033340 160000 033404
4387
4388 033344 114000
4389 033346 001600 001000
4390 033352 164003
4391 033354 164000 164000
4392 033360 164002
4393 033362 162000 033416
4394 033366 164060
4395 033370 130000
4396 033372 162000 050634
4397 033376 164040
4398 033400 162000 034612
4399
4400 033404 162000 035336
4401 033410 173400
4402 033412 160000 033274
4403
4404 033416 133400 040000
4405 033422 120000
4406 033424 026000
4407 033426 042000
4408 033430 052000
4409 033432 062000
4410 033434 072000
4411 033436 006000
4412 033440 026031
4413 033442 032001
4414 033444 166000

```

```

:
: FRAME 2 -- OFFSET AND SCALE ADJUSTMENT FRAME.
:
PIX2:  APPERO                ; INIT APPERTURE SIZE.
      POINT!LPOFF
PIX2X1: 200, 1000           ; SET POINT 1.
      PCSSET                ; SET CORRELATOR X/Y.
      DNOP, DNOP
      PCRUN
      DJSR, RBOX           ; DRAW REFERENCE BOX 1.
      PCHIT                ; ENABLE HITS...
      RELATP
      DJSR, BUFR1         ; ...AND XCT 1ST OVERLAY.
      PCNOHIT             ; DISABLE HITS.
PIX2SW: DNOP                ; SKIP4 FOR DUAL PATTERN PIX.
      DJSR, INSTR2       ; INSTRUCTIONS -- CENTER ADJUST.
      DJMP, PIX2M
      POINT
PIX2X2: 1600, 1000         ; SET POINT 2.
      PCSSET
      DNOP, DNOP
      PCRUN
      DJSR, RBOX           ; DRAW REFERENCE BOX 2.
      PCHIT                ; HITS ON...
      RELATP
      DJSR, BUFR2         ; ...AND XCT 2ND OVERLAY.
      PCNOHIT             ; HITS OFF.
      DJSR, INSTR3       ; INSTRUCTIONS -- OFF/SCA ADJ.
PIX2M:  DJSR, PCMENU       ; REFRESH MENU.
      DSTOP
      DJMP, PIX2
RBOX:  RELATP!INT6, INTX   ; INTENSIFY CENTRAL POINT.
      BASICV              ; DRAW THE BOX...
RBX:   PATH5!0
      INTX!PATH0!0        ; ...SHOWING INTENDED SIZE.
      INTX!PATH2!0
      INTX!PATH4!0
      INTX!PATH6!0
      PATH1!0             ; FINISH AT DEAD-CENTER.
      PATH5!25.
      PATH6!1             ; NOW MOVE TO OVERLAY ORIGIN...
      DPOP                ; ...AND RETURN.

```



```

4416
4417
4418
4419 033446 166000
4420 033450 160000 033460
4421 033454 160000 033534
4422 033460 114000 000600 001200 GO: POINT, 600, 1200
4423 033466 100000
4424 033470 047503 051122 046105
(1) 033527 000
4425 033530 160000 033602
4426 033534 114030 000600 001200 NOGO: POINT!BLKON, 600, 1200
4427 033542 100000
4428 033544 047503 051122 046105
4429 033602 114020 000600 001140 VPER: POINT!BLKOFF, 600, 1140
4430 033610 100000
4431 033612 020040 053101 051105
4432 033636 000000 000000 000000 VPX: 0,0,0 ; 6 BYTES.
4433 033644 022440
4434 033646 166000
4435
4436 033650 166000 INSTR1: DPOP ; NOP IF MANUAL VERIFY MODE.
4437 033652 114000 000000 000640
4438 033660 100000
4439 033662 020040 025040 025052
4440 033733 015 005012
4441 033736 020040 052040 050131
(1) 034013 000
4442 034014 160000 035300
4443
4444 034020 114000 000000 000640 INSTR2: POINT, 0, 640
4445 034026 100000
4446 034030 025052 020052 044523
4447 034102 005015 012
4448 034105 061 020056 054524
4449 034152 005015 012
4450 034155 062 020056 042101
4451 034244 005015 012
4452 034247 063 020056 054524
4453 034336 005015
4454 034340 020040 053040 051105
4455 034433 015 005012
4456 034436 027064 040440 045104
4457 034516 005015 012
4458 034521 065 020056 054524
4459 034606 160000 035224
4460
4461 034612 114000 000000 000640 INSTR3: POINT, 0, 640
4462 034620 100000
4463 034622 025052 020052 041523
4464 034672 005015 012
4465 034675 061 020056 042101
4466 034710 032060 052440 052116 S.POT:
4467 034743 015 005012
4468 034746 027062 044440 020106
(1) 035001 000

```

```

: PROCEDURAL INSTRUCTIONS ETC, ETC.
:
: GONOGO: DPOP ; NOP IF "GO", SKIP 2 IF "NO GO".
: DJMP, GO
: DJMP, NOGO
: POINT, 600, 1200
: CHAR
: .ASCII /CORRELATOR ALIGNMENT ACCEPTABLE/
: .ASCII <0>
: DJMP, VPER
: POINT!BLKON, 600, 1200
: CHAR
: .ASCII /CORRELATOR ADJUSTMENT REQUIRED/
: POINT!BLKOFF, 600, 1140
: CHAR
: .ASCII / AVERAGE VISIBILITY/
: 0,0,0 ; 6 BYTES.
: .ASCII / %/
: DPOP
: INSTR1: DPOP ; NOP IF MANUAL VERIFY MODE.
: POINT, 0, 640
: CHAR
: .ASCII / *** OVERALL ALIGNMENT VERIFICATION ***/
: .ASCII <15><12><12>
: .ASCII / TYPE <CR> TO PROCEED TO ADJUSTMENT FRAMES./
: .ASCII <0>
: DJMP, INSTR5
: INSTR2: POINT, 0, 640
: CHAR
: .ASCII / *** SIZE AND INITIAL OFFSET ADJUSTMENT ***/
: .ASCII <15><12><12>
: .ASCII /1. TYPE <3> TO SET MAXIMUM APPERTURE./
: .ASCII <15><12><12>
: .ASCII \2. ADJUST R05 AND R18 UNTIL ACTUAL = SPEC SIZE (+/- 2).\
: .ASCII <15><12><12>
: .ASCII /3. TYPE <2>, <1>, OR <0> TO CHANGE APPERTURE SELECTION./
: .ASCII <15><12>
: .ASCII \ VERIFY THAT ACTUAL = SPEC SIZE (+/- 2) FOR EACH SETTING.\
: .ASCII <15><12><12>
: .ASCII \4. ADJUST R02 AND R15 UNTIL OFFSETS = 0 (+/- 1).\
: .ASCII <15><12><12>
: .ASCII /5. TYPE <X> OR <Y> TO PROCEED WITH SCALE ADJUSTMENTS./
: DJMP, INSTR4
: INSTR3: POINT, 0, 640
: CHAR
: .ASCII / *** SCALE ADJUSTMENT AND FINE-TUNING ***/
: .ASCII <15><12><12>
: .ASCII /1. ADJUST R/
: .ASCII \04 UNTIL SCALE = 0 (+/- 1).\ ; R04(X), R17(Y)
: .ASCII <15><12><12>
: .ASCII /2. IF NECESSARY, READJUST R/
: .ASCII <0>

```

4469	035002	031060	052440	052116	O.POT:	.ASCII	\02 UNTIL OFFSET = 0 (+/- 1).\ ; R02(X), R15(Y)
4470	035036	005015	012			.ASCII	<15><12><12>
4471	035041	063	020056	042522		.ASCII	/3. REPEAT 1 AND 2 AS NECESSARY./
4472	035100	005015	012			.ASCII	<15><12><12>
4473	035103	064	020056	054524		.ASCII	/4. TYPE /
(1)	035113	000				.ASCII	<0>
4474	035114	054474	020076	047524	OTHRAX:	.ASCII	/<Y> TO ADJUST IN THE OTHER AXIS/ ; OR <X>
4475	035153	015	012			.ASCII	<15><12>
4476	035155	040	020040	054524		.ASCII	/ TYPE <C> TO RETURN TO CENTER SCREEN./
4477							
4478	035224	005015			INSTR4:	.ASCII	<15><12>
4479	035226	020040	052040	050131		.ASCII	/ TYPE <CR> TO VERIFY OVERALL ALIGNMENT./
(1)	035277	000				.ASCII	<0>
4480	035300	005015			INSTR5:	.ASCII	<15><12>
4481	035302	020040	052040	050131		.ASCII	/ TYPE <CNTRL C> TO EXIT./
4482	035334	166000			DPOP		

4484
 4485
 4486
 4487 035336 170003
 4488 035340 114000 000000 001360
 4489 035346 154240 102600
 4490 035352 162000 035442
 4491 035356 162000 035466
 4492 035362 162000 035544
 4493 035366 162000 035406
 4494 035372 162000 035574
 4495 035376 162000 035644
 4496 035402 170002 166000
 4497
 4498 035406 166000
 4499 035410 005015 053012 054124
 4500 035420 030060 030060
 4501 035424 005015 053012 054524
 4502 035434 030060 030060
 4503 035440 166000
 4504
 4505 035442 166000
 4506 035444 005015 040412 050120
 4507 035462 000060
 4508 035464 166000
 4509
 4510 035466 166000
 4511 035470 005015 051412 042520
 4512 035506 030060
 4513 035510 054040 040
 (1) 035513 000
 4514 035514 030060
 4515 035516 005015 040412 052103
 (1) 035531 000
 4516 035532 030060
 4517 035534 054040 040
 (1) 035537 000
 4518 035540 030060
 4519 035542 166000
 4520
 4521 035544 166000
 4522 035546 005015 044012 052111
 4523 035564 000000 000000 000000
 4524 035572 166000
 4525
 4526 035574 166000
 4527 035576 005015 054012 047440
 4528 035612 000000 000000 000000
 4529 035620 005015 054412 047440
 4530 035634 000000 000000 000000
 4531 035642 166000
 4532
 4533 035644 166000
 4534 035646 005015 000012
 4535 035652
 (1) 035652 020130 043117 051506

```

: CORRELATOR ALIGNMENT PARAMETERS GO IN THE MENU AREA.
:
: PCMENU: DMENU1 ; MENU ON.
: POINT, 0, 1360
: CHARS1, CHAR!INT3 ; CHAR SIZE 1, INTENSITY 3.
: DJSR, APPERT ; SELECTED APPERTURE SIZE.
: DJSR, SPECS ; SPEC SIZE VS. ACTUAL SIZE.
: DJSR, HITS ; HIT COUNTER.
: DJSR, VTX ; GRAPHICS X/Y CO-ORDS.
: DJSR, CTR.ER ; CENTERING ERROR (PART 1).
: DJSR, OAS.ER ; OFFSET/SCALE ERROR (PART 2).
: DMENU0, DPOP ; MENU OFF AND RETURN.
:
VTXY: DPOP ; NOP TO TURN ON.
.ASCII <15><12><12>/VTX: /
VT.X: .ASCII /0000/
.ASCII <15><12><12>/VTY: /
VT.Y: .ASCII /0000/
DPOP
APPERT: DPOP
.ASCII <15><12><12>/APPERTURE: /
A.SIZ: .ASCIZ /0/
DPOP
SPECS: DPOP
.ASCII <15><12><12>/SPEC SIZE: /
A.SPC: .ASCII /00/
.ASCII / X /
.ASCII <0>
A.SPC1: .ASCII /00/
.ASCII <15><12><12>/ACTUAL: /
.ASCII <0>
X.DIM: .ASCII /00/
.ASCII / X /
.ASCII <0>
Y.DIM: .ASCII /00/
DPOP
HITS: DPOP
.ASCII <15><12><12>/HIT COUNT: /
HIT.K: 0,0,0 ; 6 BYTES.
DPOP
CTR.ER: DPOP
.ASCII <15><12><12>/X OFFSET:/
X.VAL: 0,0,0 ; 6 BYTES.
.ASCII <15><12><12>/Y OFFSET:/
Y.VAL: 0,0,0 ; 6 BYTES.
DPOP
OAS.ER: DPOP
.ASCIZ <15><12><12>
O.ID: .ASCII /X OFFSET:/
    
```

(1)	035663	000				.ASCII	<0>	
4536	035664	000000	000000	000000	O.VAL:	0,0,0		; 6 BYTES.
4537	035672	005015	000012			.ASCIZ	<15><12><12>	
4538	035676				S.ID:			
(1)	035676	020130	041523	046101		.ASCII	/X SCALE: /	
(1)	035707	000				.ASCII	<0>	
4539	035710	000000	000000	000000	S.VAL:	0,0,0		; 6 BYTES.
4540	035716	166000				DPOP		


```

4542
4543
4544
4545 035720 164000
4546 035722 164000
4547 035724 162000 036050
4548 035730 116600
4549 035732 000000 000000
4550 035736 164060
4551 035740 120000
4552 035742 043777
4553 035744 053777
4554 035746 063777
4555 035750 073777
4556 035752 006250
4557 035754 043257
4558 035756 053257
4559 035760 063257
4560 035762 073257
4561 035764 006250
4562 035766 042537
4563 035770 052537
4564 035772 062537
4565 035774 072537
4566 035776 114000
4567 036000 000770 001000
4568 036004 120000
4569 036006 042020
4570 036010 016010
4571 036012 072020
4572 036014 164040
4573
4574 036016 114000
4575 036020 000000 000000
4576 036024 120000
4577 036026 006010
4578 036030 066020
4579 036032 012020
4580 036034 076020
4581
4582 036036 162000 036172
4583 036042 173400
4584 036044 160000 035720
  
```

```

: FRAME 3 -- WRITING TABLET EXERCISOR.
PIX3: DNOP
      DNOP ; "DSKP2" WHEN PEN ENTERS MENU.
      DJSR, TBALL ; TRACK THE TABLET PEN.
      POINT!INT3
      0, 0
      PCHIT ; ENABLE CORRELATOR HITS.
      BASICV
      INTX!PATH0!1777 ; OUTER BOX.
      INTX!PATH2!1777
      INTX!PATH4!1777
      INTX!PATH6!1777
      PATH1!250
      INTX!PATH0!<1777-520> ; MIDDLE BOX.
      INTX!PATH2!<1777-520>
      INTX!PATH4!<1777-520>
      INTX!PATH6!<1777-520>
      PATH1!250
      INTX!PATH0!<1777-1240> ; INNER BOX.
      INTX!PATH2!<1777-1240>
      INTX!PATH4!<1777-1240>
      INTX!PATH6!<1777-1240>
      POINT
      770, 1000
      BASICV
      INTX!PATH0!20 ; CROSS AT DEAD-CENTER.
      PATH3!10
      INTX!PATH6!20
      PCNOHIT ; DISABLE HITS.
      POINT ; NOW MARK THE LAST HIT POINT.
      0, 0
      BASICV
      PATH1!10
      INTX!PATH5!20 ; X-MARKS-THE-SPOT.
      PATH2!20
      INTX!PATH7!20
      DJSR, TBMENU ; DO THE MENU.
      DSTOP
      DJMP, PIX3
  
```

```

4586
4587
4588
4589
4590 036050 175000
4591 036052 114000
4592 036054 001000 001000
4593 036060 164003
4594 036062 132600
4595 036064 040000
4596 036066 164000 164000
4597
(1) 000014
(1) 000011
(1) 036072 120000
(1) 036074 032017
(1) 036076 022006
(1) 036100 042014
(1) 036102 046011
(1) 036104 052014
(1) 036106 056011
(1) 036110 062014
(1) 036112 066011
(1) 036114 072014
(1) 036116 076011
(1) 036120 002006
(1) 036122 012017
4598 036124 164000 164000
4599
(1) 000020
(1) 000014
(1) 036130 120000
(1) 036132 032024
(1) 036134 022010
(1) 036136 042020
(1) 036140 046014
(1) 036142 052020
(1) 036144 056014
(1) 036146 062020
(1) 036150 066014
(1) 036152 072020
(1) 036154 076014
(1) 036156 002010
(1) 036160 012024
4600 036162 164000 164000
4601 036166 164002
4602 036170 166000
    
```

```

:
: SUBROUTINE TO POSITION THE TABLET TRACKING BALL.
: RETURN WITH X-Y AT BALL ORIGIN.
:
    
```

```

TBALL: APPERO
POINT
TBX: 1000, 1000 ; X-Y COORDINATES...
PCSET ;...SET X-Y => POINT CORRELATOR.
RELATP!INT3
INTX ; INTENSIFY THE ORIGIN.
DNOP, DNOP
LV= <15./5>*4 ; BUILD A 4:3 OCTAGON...
SV= <15./5>*3 ;...OF RADIUS 15. !
BASICV
PATH6!15.
PATH4!<LV/2>
INTX!PATH0!LV
INTX!PATH1!SV
INTX!PATH2!LV
INTX!PATH3!SV
INTX!PATH4!LV
INTX!PATH5!SV
INTX!PATH6!LV
INTX!PATH7!SV
PATH0!<LV/2>
PATH2!15.
DNOP, DNOP
LV= <20./5>*4 ; BUILD A 4:3 OCTAGON...
SV= <20./5>*3 ;...OF RADIUS 20. !
BASICV
PATH6!20.
PATH4!<LV/2>
INTX!PATH0!LV
INTX!PATH1!SV
INTX!PATH2!LV
INTX!PATH3!SV
INTX!PATH4!LV
INTX!PATH5!SV
INTX!PATH6!LV
INTX!PATH7!SV
PATH0!<LV/2>
PATH2!20.
DNOP, DNOP
PCRUN ; CORRELATOR RUN.
DPOP
    
```



```

4604
4605
4606
4607 036172 170003
4608 036174 161002
4609 036176 162000 036050
4610
4611 036202 114000 000000 001700
4612 036210 164060
4613 036212 154240
4614 036214 100000
4615 036216 005015 052012 041101
4616 036230 030060 030060
4617 036234 005015 052012 041101
4618 036246 030060 030060
4619 036252 005015 044012 052111
4620 036264 030060 030060
4621 036270 005015 044012 052111
4622 036302 030060 030060
4623 036306 162000 035442
4624 036312 162000 035544
4625 036316 005015 050012 047105
4626 036332 164000
4627 036334 052440 020120
4628 036340 161002
4629 036342 047504 047127
4630 036346 005015 050012 047522
4631 036364 164000
4632 036366 047111
4633 036370 161002
4634 036372 052517 020124
4635 036376 164040
4636 036400 170002 166000
4637

: A SEPERATE MENU FOR THE TABLET EXERCISER.
:
: BMENU: DMENU1 ; MENU ON
: DSKP2 ; "DNOP" WHEN PEN ENTERS MENU.
: DJSR, TBALL
:
: POINT, 0, 1700
: PCHIT ; ALLOW HITS ON THE TEXT.
: CHARS1
: CHAR
: .ASCII <15><12><12>/TAB X: /
TAB.X: .ASCII /0000/
: .ASCII <15><12><12>/TAB Y: /
TAB.Y: .ASCII /0000/
: .ASCII <15><12><12>/HIT X: /
HIT.X: .ASCII /0000/
: .ASCII <15><12><12>/HIT Y: /
HIT.Y: .ASCII /0000/
: DJSR, APPERT
: DJSR, HITS
: .ASCII <15><12><12>/PEN TIP: /
PEN.SW: DNOP ; DSKP3 IF DOWN.
: .ASCII / UP /
: DSKP2
: .ASCII /DOWN/
: .ASCII <15><12><12>/PROXIMITY: /
PRX.SW: DNOP ; DSKP2 IF OUT.
: .ASCII /IN/
: DSKP2
: .ASCII /OUT /
: PCNOHIT ; DISABLE HITS.
: DMENU0, DPOP ; MENU OFF AND RETURN.
    
```



```

4645      .SBTTL SCOPE HANDLER ROUTINE
(1)
(2)      ;*****
(1)      ;*THIS ROUTINE CONTROLS THE LOOPING OF SUBTESTS. IT WILL INCREMENT
(1)      ;*AND LOAD THE TEST NUMBER($STNM) INTO THE DISPLAY REG.(DISPLAY<7:0>)
(1)      ;*AND LOAD THE ERROR FLAG ($ERFLG) INTO DISPLAY<15:08>
(1)      ;*THE SWITCH OPTIONS PROVIDED BY THIS ROUTINE ARE:
(1)      ;*SW14=1      LOOP ON TEST
(1)      ;*SW08=1      LOOP ON TEST IN SWR<7:0>
(1)      ;*CALL
(1)      ;*      SCOPE      ;;SCOPE=IOT
(1)
(1)      $SCOPE:
(3)      036404 032777 040000 142526      BIT      #BIT14,@SWR      ;TEST IF SW14 = 1
(3)      036412 001047      BNE      $OVER      ;BR IF SET
(3)      036414 005737 014474      TST      HOLD      ;TEST IF LOOP ON PICTURE ?
(3)      036420 001044      BNE      $OVER      ;BR IF LOOP ON THIS TEST
(1)      036422 032777 040000 142510 1$: BIT      #BIT14,@SWR      ;;LOOP ON PRESENT TEST?
(1)      036430 001040      BNE      $OVER      ;;YES IF SW14=1
(1)      ;#####START OF CODE FOR THE XOR TESTER#####
(1)      036432 000416      $XTSTR: BR      6$      ;;IF RUNNING ON THE 'XOR' TESTER CHANGE
(1)      ;THIS INSTRUCTION TO A 'NOP' (NOP=240)
(1)      036434 013746 000004      MOV      @#ERRVEC,-(SP) ;SAVE THE CONTENTS OF THE ERROR VECTOR
(1)      036440 012737 036460 000004      MOV      #5$,@#ERRVEC ;SET FOR TIMEOUT
(1)      036446 005737 177060      TST      @#177060      ;TIME OUT ON XOR?
(1)      036452 012637 000004      MOV      (SP)+,@#ERRVEC ;RESTORE THE ERROR VECTOR
(1)      036456 000414      BR      $SVLAD      ;GO TO THE NEXT TEST
(1)      036460 022626      5$: CMP      (SP)+,(SP)+ ;CLEAR THE STACK AFTER A TIME OUT
(1)      036462 012637 000004      MOV      (SP)+,@#ERRVEC ;RESTORE THE ERROR VECTOR
(1)      036466 000421      BR      $OVER      ;LOOP ON THE PRESENT TEST
(1)      036470      6$:;#####END OF CODE FOR THE XOR TESTER#####
(1)      036470 032777 000400 142442      BIT      #BIT08,@SWR ;LOOP ON SPEC. TEST?
(1)      036476 001404      BEQ      $SVLAD      ;BR IF NO
(1)      036500 127737 142434 001102      CMPB     @SWR,$STNM ;ON THE RIGHT TEST? SWR<7:0>
(1)      036506 001411      BEQ      $OVER      ;BR IF YES
(1)      036510 105237 001102      $SVLAD: INCB     $STNM ;COUNT TEST NUMBERS
(1)      036514 113737 001102 001176      MOV      $STNM,$TESTN ;SET TEST NUMBER IN APT MAILBOX
(1)      036522 011637 001106      MOV      (SP),$LPADR ;SAVE SCOPE LOOP ADDRESS
(1)      036526 105037 001103      CLRB     $ERFLG ;ZERO THE ERROR FLAG
(1)      036532 013777 001102 142402 $OVER: MOV      $STNM,@DISPLAY ;DISPLAY TEST NUMBER
(1)      036540 013716 001106      MOV      $LPADR,(SP) ;FUDGE RETURN ADDRESS
(1)      036544 000002      RTI      ;FIXES PS
4646      .=-2
4647      036544 000005      RESET
4648      036546 005737 002322      TST      KRBD ;TEST IF KEYBOARD CONTROL
4649      036552 001403      BEQ      1$ ;BR IF NOT
4650      036554 052777 000100 142362 1$: BIS      #BIT6,@$TKS ;ENABLE KEYBOARD INTR.
4651      036562 000002      RTI
4652      036564 000240      NOP
4653      036566 000240      NOP
4654      000200      APTSIZE=200
    
```


4656 ;DISPATCH TABLE OF THE STARTING ADDRESSES OF EACH TEST

4657			DISPTC: TST1
4658	036570	002324	TST2
4659	036572	002336	TST3
4660	036574	002350	TST4
4661	036576	002544	TST5
4662	036600	002572	TST6
4663	036602	002630	TST7
4664	036604	003232	TST10
4665	036606	003446	TST11
4666	036610	003460	TST12
4667	036612	003534	TST13
4668	036614	004144	TST14
4669	036616	004172	TST15
4670	036620	004256	TST16
4671	036622	004766	TST17
4672	036624	005130	TST20
4673	036626	005142	TST21
4674	036630	005306	TST22
4675	036632	005372	TST23
4676	036634	005516	TST24
4677	036636	005770	TST25
4678	036640	006024	TST26A
4679	036642	006242	TST27
4680	036644	006526	TST30
4681	036646	007170	TST31
4682	036650	010006	TST32
4683	036652	012522	0
4684	036654	000000	

4685
4686 036656 000000 BUFFER: 0 ; USE REMAINING CORE UNDER 12K FOR BUFFER.

4687
4689
4690 000075
4691 000145
4692 036660 000075
4693 036660 000145
4694
4696 000001

```
*****  
PROGK= . -8.  
TOTLK= BFREND -8.  
.PRINT PROGK ; PROGRAM SIZE IN 1/8K UNITS.  
.PRINT TOTLK ; TOTAL SIZE (BUFFER SPACE INCLUDED).  
*****  
.END
```

ABASE = 172000	19#	33							
ABORT 012137	1155*	1189	1328*	1409	1708*	1716#			
ABSVCT= 144000	190#	2737	2793	2848	2914	3141	4145		
ACDW1 = 175620	23#	33							
ACDW2 = 000310	24#	33							
ACPUOP= 000000	33								
ACRLF 004514	689	696#							
ADDW0 = 000000	33								
ADDW1 = 000000	33								
ADDW10= 000000	33								
ADDW11= 000000	33								
ADDW12= 000000	33								
ADDW13= 000000	33								
ADDW14= 000000	33								
ADDW15= 000000	33								
ADDW2 = 000000	33								
ADDW3 = 000000	33								
ADDW4 = 000000	33								
ADDW5 = 000000	33								
ADDW6 = 000000	33								
ADDW7 = 000000	33								
ADDW8 = 000000	33								
ADDW9 = 000000	33								
ADEVCT= 000000	33								
ADEVN = 000000	33								
AENV = 000000	33								
AENVN = 000000	33								
AFATAL= 000000	33								
AMADR1= 000000	33								
AMADR2= 000000	33								
AMADR3= 000000	33								
AMADR4= 000000	33								
AMAMS1= 000000	33								
AMAMS2= 000000	33								
AMAMS3= 000000	33								
AMAMS4= 000000	33								
AMSGAD= 000000	33								
AMSGLG= 000000	33								
AMSGTY= 000000	33								
AMTYP1= 000000	33								
AMTYP2= 000000	33								
AMTYP3= 000000	33								
AMTYP4= 000000	33								
APASS = 000000	33								
APPERT 035442	1757*	1767*	1866*	4490	4505#	4623			
APPERX 012000	1159	1375	1660*	1675#	1908				
APPER0= 175000	1671	1675	4281#	4282	4283	4284	4322	4373	4590
APPER1= 175200	1672	4282#							
APPER2= 175400	1673	4283#							
APPER3= 175600	1674	4284#							
APPTBL 011770	1660	1671#							
APRIOR= 000200	21#	33							
APTSIZ= 000200	77	4654#							
ASIZE 012136	1147*	1187	1413	1655	1694*	1715#	1905		
ASWREG= 000000	33								
ASZTBL 012002	1661	1677#							

Symbol	17#	77*	1845	1846*	1849*	1853*	4645*												
ERRVEC=	000004	17#	77*	1845	1846*	1849*	1853*	4645*											
EVALU8	011406	1573	1583#																
FILE	014632	2175*	2177	2202#															
FILE4A	004572	669	721#	740															
FILLIT	004536	684	688	704#															
FIL14A	005624	888	898#																
FIXVCT	001730	110	113#	160															
FRME0	015020	300	2274#	2356															
FRME10	023160	838	3150#																
FRME11	023436	855	3166#																
FRME14	024274	903	908	913	918	3247#	3259												
FRME16	027250	959	3724#	4017															
FRME17	032000	1047	1061	1076	4047#	4229													
FRME2	017516	365	374	2448#	2702														
FRME3	020546	504	2707#	2951															
FRME5	022660	803	3059#	3113															
FRME6	023106	822	825	3123#															
FRM10	024174	841	858	3197#	3212														
FRM11D	023724	3186#	3195																
FRM11M	024234	861	3214#	3229															
FRM11S	023716	864	3183#																
FRM14A	024300	899*	904*	909*	914*	3249#													
FRM14B	024302	900*	905*	910*	915*	3250#													
FRM16B	027474	953*	954*	2128	3775#														
FRM16C	027640	3776	3801#																
FRM17E	032452	1114	4232#	4246															
FRM17F	032450	1064	4229#																
GETXY	013430	1877	1994#																
GNS =	***** U	28																	
GO	033460	4420	4422#																
GONOGO	033446	1160*	1222*	4365	4419#														
GRAPH	024406	931	3263#	3298															
GRAPHX=	120000	185#	187	3291	4107														
GRAPHY=	124000	186#	3275	4117															
GRPINC	024432	928*	932*	933	3274#														
HALFX =	000777	278#	438	439	440	441	442	443	444	445									
HIMAG	011402	1477	1577#	1583	1588	1596													
HISGN	011404	1578#	1589	1598	1604														
HITCNT	014462	952*	2126*	2127	2156#														
HITK	012144	1167*	1211	1724#	1725	1727*	1729	1867*											
HITP	007715	1180*	1194	1236*	1243#														
HITS	035544	1759*	1769*	1865*	4492	4521#	4624												
HITUP	012142	1168	1237	1723#	1868	1921													
HIT.K	035564	1729	1730*	4523#															
HIT.X	036264	1929	4620#																
HIT.Y	036302	1931	4622#																
HI2	012514	1514	1517	1826#															
HOLD	014474	99*	156*	169*	171*	2161#	4645												
HT =	000011	17#																	
INCR =	000100	250#	928	933	3274	4102													
INSTR1	033650	1146*	1150*	4364	4436#														
INSTR2	034020	4385	4444#																
INSTR3	034612	4398	4461#																
INSTR4	035224	4459	4478#																
INSTR5	035300	4442	4480#																
INTX =	040000	274#	322	393	407	420	431	438	439	440	441	442	443	444					

.\$APTH	10#	32
.\$APTY	10#	
.\$CATC	7#	28
.\$CMTA	7#	33
.\$EOP	7#	2058
.\$PARM	8#	
.\$POWE	8#	
.\$SCOP	8#	4645
.\$SWDO	8#	
.\$STRAP	9#	
.\$STRPT	9#	
.\$TYPD	9#	
.\$TYPE	9#	

. ABS. 036660 000 CON RO REL LCL D

ERRORS DETECTED: 0

CZVSFA,CZVSFA/LI:TOC/CRF=CZVSFA
 RUN-TIME: 22 13 1 SECONDS
 RUN-TIME RATIO: 91/37=2.4
 CORE USED: 20K (39 PAGES)