

FAST GRAPHICS ROMPACK PRIMER

65 HIGH LEVEL GRAPHICS COMMANDS.
MORE THAN 10 TIMES FASTER THAN BASIC.
CROSSHAIR (4010 STYLE) JOYSTICK CURSOR.
8 TIMES THE PACKING DENSITY (>10,000 VECTORS).
SEVERAL COMMANDS ADD CAPABILITIES NOT POSSIBLE IN BASIC.

BENCHMARKS FROM 13.7X FASTER TO 127X FASTER THAN 4051 BASIC.
(BENCHMARKS FROM 4.5X FASTER TO 61X FASTER THAN 4052 BASIC.)

OVERVIEW:

THE MAIN FUNCTION OF FAST GRAPHICS IS TO SUPPLEMENT THE INTERACTIVE GRAPHICS CAPABILITY OF THE 4051. THIS INCLUDES SEVERAL IMAGE MANIPULATION COMMANDS, WRITE-THRU (NON-STORE) IMAGE VECTORS AND CHARACTER STRINGS, AND PROGRAMMABLE SOUND GENERATION. SOME 4051 BASIC DEFICIENCIES HAVE ALSO BEEN CORRECTED AS WELL. MAKING THE ADDITIONAL CALLS REASONABLY ORTHOGONAL HAS BEEN ATTEMPTED WITH OCCASIONAL LAPSES TO MAKE SOMETHING USEFUL FOR NEARLY EVERY SITUATION.

IN ORDER TO IMPROVE THE SPEED AND STORAGE DENSITY OF GRAPHICS, FAST GRAPHICS PACKS IMAGES INTO VALID ASCII CHARACTER STRINGS. THIS RESULTS IN ABOUT EIGHT TIMES THE STORAGE CAPACITY FOR ANY COMPLEX IMAGES. THE NEW FORMAT REQUIRES 3 BYTES/VECTOR (1 BIT MOVE FLAG, 10 BITS X & 10 BITS Y) INSTEAD OF 24 BYTES / VECTOR (X,Y,Z). IMAGES CAN BE CONVERTED FROM FLOATING POINT TO STRING FORMAT OR FROM STRING TO FLOATING POINT FORMAT WITH EASE. STRING IMAGES CAN ALSO BE MANIPULATED AND SAVED OR READ FROM MAG TAPE FILES IN STRING FORM. NOTE THAT A FREE ADVANTAGE OCCURS WHEN IMAGES ARE IN THE STRING FORMAT - ALL OF THE 4051 BASIC STRING FUNCTIONS CAN ALSO BE USED FOR IMAGES OR ANY OTHER PURPOSES.

POWERFUL IMAGE MANIPULATION COMMANDS SUPPORT THE STRING FORMAT AND GIVE THE 4051 INCREDIBLE SPEED INCREASES WHEN COMPARED TO STANDARD 4051 (AND 4052) BASIC GRAPHICS COMMANDS.

ALL IMAGE DISPLAY COMMANDS RESTORE THE GRAPHICS CURSOR TO THE POSITION IT HAD BEFORE THE CALL WAS EXECUTED. SO THE GRAPHICS CURSOR AND THE JOYSTICK CURSOR ARE INDEPENDENTLY MAINTAINED.

NOTE THAT EXTENSIVE ERROR CHECKING EXISTS WITHIN THE FAST GRAPHICS CALL PARAMETERS.

THERE ARE FOUR DIFFERENT FUNDAMENTAL CATAGORIES OF FAST GRAPHICS COMMANDS. THESE CATEGORIES ARE:

1. LOCATION: USED TO LOCATE THE GRAPHICS CURSOR OR THE JOYSTICK CURSOR OR POINTS IN IMAGE VECTOR STRINGS.
2. DISPLAY: USED TO DISPLAY CURSORS OR IMAGES OR TEXT.
3. CREATE/MODIFY: USED TO CREATE IMAGES OR STRINGS OR MODIFY IMAGES OR STRINGS.

4. CHARACTER: USED TO PRINT OR PLAY (SOUND) SPECIAL CHARACTER STRINGS.

NOTE THAT ALL OF THE STANDARD 4051 STRING FUNCTIONS CAN BE USED TO LOCATE A VECTOR OR TO REPLACE PART OF AN IMAGE, OR TO EXTRACT PART OF AN IMAGE, OR TO CONCATENATE IMAGES ETC.

MOST OF THE COMMANDS HAVE FOUR DIFFERENT FORMS. THE DIFFERENT FORMS HAVE DIFFERENT PREFIXES WITH IDENTICAL SUFFIXES. THE DIFFERENT PREFIXES ARE:

A: DISPLAY OR IMAGE STRING "ABSOLUTE"
R: DISPLAY OR IMAGE STRING "RELATIVE"
G: GRAPHICS CURRSOR "RELATIVE"
J: JOYSTICK CURRSOR "ABSOLUTE"

NOTE THAT THE COMMAND CALL SYNTAX IS CONSISTANT AMONG THE DIFFERENT FORMS. I.E. THE JOYSTICK ABSOLUTE FORM SYNTAX IS IDENTICAL TO THE ABSOLUTE FORM SYNTAX WITH SOME ADDITIONAL COUNT OR TIME AND KEYBOARD KEY PARAMETERS.

AN ADDITIONAL FORM EXISTS FOR COMMANDS THAT DO NOT FOLLOW THE ABOVE RULES. THIS IS THE "SPECIAL" FORM.

COMMAND LIST:

LOCATION:

SPECIAL: BOUNDS LOCATE
ABSOLUTE: AGIN APOINT
RELATIVE: RGIN RPOINT
GRAPHICS: GGIN GPOINT
JOYSTICK: JGIN JPOINT

DISPLAY:

SPECIAL: DASHED DOTTED KABOOM RUBBER VERTEX
ABSOLUTE: ACROSS ADRAW ADOTS APRINT AINPUT
RELATIVE: RCROSS RDRAW RDOTS RPRINT RINPUT
GRAPHICS: GCROSS GDRAW GDOTS GPRINT GINPUT
JOYSTICK: JCROSS JDRAW JDOTS JPRINT JINPUT

CREATE/MODIFY:

SPECIAL: IMAGES CHANGE DEFINE POINTS TOGGLE

ABSOLUTE: AMOVE ASCALE ASHEAR ATAPER AROTAT(E)

RELATIVE: RMOVE RSCALE RSHEAR RTAPER RROTAT(E)

GRAPHICS: GMOVE GSCALE GSHEAR GTAPER GROTAT(E)

JOYSTICK: JMOVE JSCALE JSHEAR JTAPER JROTAT(E)

CHARACTER:

SPECIAL: PRINTS INPUTS STRING SOUNDS MUZAKT

COMMAND LIMITATIONS:

ALL X,Y POSITIONS ARE IN GDU'S (0.128 DISPLAY RESOLUTION STEP SIZES).
COORDINATES ARE LIMITED TO THE DEFAULT DISPLAY: 0 TO 130 HORIZONTAL AND
0 to 100 VERTICAL (0 TO 1015 AND 0 TO 781 USABLE POINTS, RESPECTIVELY).

ALL GRAPHICS DISPLAY COMMANDS RESTORE THE GRAPHICS CURSOR.
ALL COMMANDS ASSUME DEFAULT WINDOW AND VIEWPORT CONDITIONS.
ALL RELATIVE IMAGE COMMANDS ARE RELATIVE TO THE FIRST IMAGE POINT.
SOME RELATIVE NON-IMAGE COMMANDS ARE RELATIVE TO THE DISPLAY CENTER.
SOME RELATIVE NON-IMAGE COMMANDS ARE RELATIVE TO THE GRAPHICS CURSOR.
ANY VECTORS THAT ARE BEYOND THE DEFAULT DISPLAY BOUNDARY WILL BE LIMITED
TO THE DEFAULT DISPLAY BOUNDARY.
UNDEFINED NUMERIC AND UNDEFINED STRING PARAMETERS ARE ALLOWED IN MANY OF
THE COMMAND RESULTS.
ANY UNDEFINED STRING VARIABLE RESULTS WILL BE DIMENSIONED TO THE 4051
BASIC DEFAULT OF 72 CHARACTERS.
ALL DEFINED IMAGE STRINGS MUST HAVE A LENGTH OF 0 MOD 3. THUS, NO PARTIAL
VECTORS (LESS THAN 3 BYTES) ARE ALLOWED.
ALL DEFINED ARGUMENT CHARACTER STRINGS MUST HAVE A LENGTH > 0. THUS, NO
NULL ARGUMENT STRINGS ARE ALLOWED.

NOTE THAT ALL OF THE BELOW VARIABLE NAMES ARE ONLY FOR ILLUSTRATION
PURPOSES. ANY VALID EQUIVALENT TYPE BASIC VARIABLE NAMES CAN BE USED.

CALL PARAMETER ABBREVIATIONS:

CHARACTER PARAMETERS:

C\$: CHARACTER STRING.
I\$: IMAGE VECTOR STRING.
K\$: KEYBOARD KEY(S).

C: DISPLAY REPETITION COUNT PARAMETER:

C >= +0.5: DRAW ABS(C) TIMES IN STORAGE MODE.
C <= -0.5: DRAW ABS(C) TIMES IN REFRESH MODE.

T: TIME LIMIT PARAMETER:

MAXIMUM WAIT TIME IN MILLISECONDS. ZERO TIME IS ALLOWED.

N: IMAGE VECTOR STRING INDEX PARAMETER:

$0 \leq N \leq ((\text{LEN}(\text{I\$}) / 3) - 1)$.

THE FIRST IMAGE STRING POINT IS AT INDEX 0, AND IT IS USUALLY A MOVE. ADDITIONALLY, THERE ARE TIMES THAT N CAN BE A NON-INTEGGER VALUE TO PRESERVE AN ARGUMENT SIGN. $-0.5 < N < 0.5$ ALLOWS FOR THE SIGN OF N TO BE PRESERVED WHILE ACCESSING IMAGE VECTOR ZERO.

DISPLAY COORDINATE PARAMETERS:

A: ROTATION ANGLE.

H,V: HORIZONTAL AND VERTICAL SCALE FACTORS.

X,Y,X0,Y0,X1,Y1: X AND Y DISPLAY COORDINATES.

CALL PARAMETER TYPES:

ARY.ARG: COMPLETELY DEFINED NUMERIC ARRAY ARGUMENT

NUM.ARG: DEFINED SIMPLE NUMERIC ARGUMENT (EXPRESSION, VALUE, OR VARIABLE)

IMG.ARG: DEFINED IMAGE STRING ARGUMENT (LENGTH > 0 & = 0 MOD 3)

STR.ARG: LITERAL STRING OR A DEFINED CHARACTER STRING ARGUMENT (LENGTH > 0)

ARY.RES: DIMENSIONED NUMERIC ARRAY RESULT (MAY BE UNDEFINED)

NUM.RES: SIMPLE NUMERIC VARIABLE RESULT (MAY BE UNDEFINED)

IMG.RES: IMAGE STRING RESULT (MAY BE UNDEFINED)

STR.RES: CHARACTER STRING RESULT (MAY BE UNDEFINED)

ARY.ARG&RES: COMPLETELY DEFINED NUMERIC ARRAY ARGUMENT & RESULT

NUM.ARG&RES: DEFINED SIMPLE NUMERIC VARIABLE ARGUMENT & RESULT

IMG.ARG&RES: DEFINED IMAGE STRING ARGUMENT & RESULT (LENGTH > 0 & = 0 MOD 3)

STR.ARG&RES: DEFINED CHARACTER STRING ARGUMENT & RESULT (LENGTH > 0)

IMAGE BOUNDS:

CALL "BOUNDS", I\$, X0, Y0, X1, Y1

CALL "BOUNDS", IMG.ARG, NUM.RES, NUM.RES, NUM.RES, NUM.RES

FIND VECTOR IMAGE I\$, MIN X, MIN Y, MAX X, MAX Y.

JOYSTICK LOCATION:

CALL "LOCATE", C, X, Y, K\$

CALL "LOCATE", NUM.ARG&RES, NUM.RES, NUM.RES, STR.RES

C >= +0.5 DRAW CROSSHAIR ABS(C) TIMES IN STORAGE MODE.
C <= -0.5 DRAW CROSSHAIR ABS(C) TIMES IN REFRESH MODE.

PRINT JOYSTICK CURSOR LOCATION AT GRAPHICE CURSOR POSITION.
DRAW CROSSHAIR JOYSTICK CURSOR C TIMES OR UNTIL A KEYBOARD KEY IS PRESSED,
WHICHEVER OCCURS FIRST.
AFTER THE COMMAND EXIT: C = ACTUAL DISPLAY COUNT, X AND Y ARE UPDATED TO
THE JOYSTICK POSITION, AND ANY PENDING KEYBOARD KEY IS SAVED IN K\$.

ABSOLUTE GIN:

CALL "AGIN",X,Y

CALL "AGIN",NUM.RES,NUM.RES

FIND GRAPHIC CURSOR X,Y POSITION.
NO CROSSHAIR CURSOR IS DRAWN FOR THE GIN CALLS.

RELATIVE GIN:

CALL "RGIN",X,Y

CALL "RGIN",NUM.ARG&RES,NUM.ARG&RES

FIND GRAPHIC CURSOR X,Y POSITION RELATIVE TO POSITION X,Y.
NO CROSSHAIR CURSOR IS DRAWN FOR THE GIN CALLS.

GRAPHICS GIN:

CALL "GGIN",X,Y

CALL "GGIN",NUM.ARG&RES,NUM.ARG&RES

FIND POSITION AT DELTA X,Y RELATIVE TO THE GRAPHIC CURSOR POSITION.
NO CROSSHAIR CURSOR IS DRAWN FOR THE GIN CALLS.

JOYSTICK GIN:

CALL "JGIN",T,X,Y,K\$

CALL "JGIN",NUM.ARG&RES,NUM.RES,NUM.RES,STR.RES

WAIT FOR T MILLISECONDS OR UNTIL A KEYBOARD KEY IS PRESSED, WHICHEVER
OCCURS FIRST, THEN FIND JOYSTICK CURSOR POSITION.
AFTER THE COMMAND EXIT: T = ACTUAL WAIT TIME, X AND Y ARE UPDATED TO
THE JOYSTICK CURSOR POSITION, ANY PENDING KEYBOARD KEY IS SAVED IN K\$.
THE FASTEST WAY TO READ THE JOYSTICK POSITION, ZERO DELAY TIME IS ALLOWED.

NO CROSSHAIR CURSOR IS DRAWN FOR THE GIN CALLS.

ABSOLUTE POINT:

CALL "APOINT", I\$, N, X, Y

CALL "APOINT", IMG.ARG, NUM.ARG&RES, NUM.ARG&RES, NUM.ARG&RES

FIND VECTOR IMAGE I\$ POINT N & X, Y POSITION NEAREST TO INPUT POSITION X, Y.
AT CALL ENTRY N = BEGINNING VECTOR NUMBER TO START SEARCH FROM.
AT CALL EXIT SIGN(N) INDICATES IF THE VECTOR IS A MOVE OR A DRAW.

RELATIVE POINT:

CALL "RPOINT", I\$, N, X, Y

CALL "RPOINT", IMG.ARG, NUM.ARG&RES, NUM.ARG&RES, NUM.ARG&RES

FIND VECTOR IMAGE I\$ POINT N & X, Y NEAREST TO RELATIVE INPUT POSITION X, Y.
AT CALL ENTRY N = BEGINNING VECTOR NUMBER TO START SEARCH FROM.
AT CALL EXIT SIGN(N) INDICATES IF THE VECTOR IS A MOVE OR A DRAW.

GRAPHICS POINT:

CALL "GPOINT", I\$, N, X, Y

CALL "GPOINT", IMG.ARG, NUM.ARG&RES, NUM.ARG&RES, NUM.ARG&RES

FIND VECTOR IMAGE I\$ POINT N & X, Y NEAREST TO X, Y RELATIVE TO THE GRAPHIC CURSOR POSITION.
AT CALL ENTRY N = BEGINNING VECTOR NUMBER TO START SEARCH FROM.
AT CALL EXIT SIGN(N) INDICATES IF THE VECTOR IS A MOVE OR A DRAW.

JOYSTICK POINT:

CALL "JPOINT", I\$, C, N, X, Y, K\$

CALL "JPOINT", IMG.ARG, NUM.ARG&RES, NUM.ARG&RES, NUM.RES, NUM.RES, STR.RES

C >= +0.5 DRAW CROSSHAIR ABS(C) TIMES IN STORAGE MODE.
C <= -0.5 DRAW CROSSHAIR ABS(C) TIMES IN REFRESH MODE.

FIND VECTOR IMAGE I\$ POINT N & X, Y POSITION NEAREST TO JOYSTICK CURSOR POSITION.
DRAW CROSSHAIR JOYSTICK CURSOR C TIMES OR UNTIL A KEYBOARD KEY IS PRESSED, WHICHEVER OCCURS FIRST.
AT CALL ENTRY N = BEGINNING VECTOR NUMBER TO START SEARCHING FROM.
AT CALL EXIT EXIT: C = ACTUAL DISPLAY COUNT, SIGN(N) INDICATES IF THE

VECTOR IS A MOVE OR A DRAW, X AND Y ARE UPDATED TO THE JOYSTICK POSITION,
AND ANY PENDING KEYBOARD KEY IS SAVED IN K\$.

DASHED GRID:

CALL "DASHED",H,V,X0,Y0,X1,Y1

CALL "DASHED",NUM.ARG,NUM.ARG,NUM.ARG,NUM.ARG,NUM.ARG,NUM.ARG

DRAW DASHED GRID WITH HORIZONTAL PITCH H AND VERTICLE PITCH V.
THE DASHED GRID STARTS AT X0,Y0 AND ENDS AT X1,Y1 (OR BEFORE).

DOTTED GRID:

CALL "DOTTED",H,V,X0,Y0,X1,Y1

CALL "DOTTED",NUM.ARG,NUM.ARG,NUM.ARG,NUM.ARG,NUM.ARG,NUM.ARG

DRAW DOTTED GRID WITH HORIZONTAL PITCH H AND VERTICLE PITCH V.
THE DOTTED GRID STARTS AT X0,Y0 AND ENDS AT X1,Y1 (OR BEFORE).

KABOOM:

CALL "KABOOM",C,X,Y

CALL "KABOOM",NUM.ARG,NUM.ARG,NUM.ARG

DRAW EXPLOSION C TIMES AT POSITION X,Y. MARVIN REALLY LIKES THIS COMMAND.

RUBBER BAND LINE:

CALL "RUBBER",C,X,Y,K\$

CALL "RUBBER",NUM.ARG&RES,NUM.RES,NUM.RES,STR.RES

C >= +0.5 DRAW CROSSHAIR ABS(C) TIMES IN STORAGE MODE.

C <= -0.5 DRAW CROSSHAIR ABS(C) TIMES IN REFRESH MODE.

DRAW RUBBER BAND LINE FROM JOYSTICK CURSOR TO GRAPHIC CURSOR POSITION.
DRAW CROSSHAIR JOYSTICK CURSOR C TIMES OR UNTIL A KEYBOARD KEY IS PRESSED,
WHICHEVER OCCURS FIRST.

AFTER THE COMMAND EXIT: C = ACTUAL DISPLAY COUNT, X AND Y ARE UPDATED TO
THE JOYSTICK POSITION, AND ANY PENDING KEYBOARD KEY IS SAVED IN K\$.

VERTEX:

CALL "VERTEX",I\$,C,X,Y,K\$

CALL "VERTEX",IMG.ARG,NUM.ARG&RES,NUM.RES,NUM.RES,STR.RES

C >= +0.5 DRAW CROSSHAIR ABS(C) TIMES IN STORAGE MODE.

C <= -0.5 DRAW CROSSHAIR ABS(C) TIMES IN REFRESH MODE.

DRAW RUBBER BAND LINES FROM JOYSTICK CURSOR TO EVERY POINT IN VECTOR IMAGE I\$.

DRAW VERTICES C TIMES OR UNTIL A KEYBOARD KEY IS PRESSED, WHICHEVER OCCURS FIRST.

AFTER THE COMMAND EXIT: C = ACTUAL DISPLAY COUNT, X AND Y ARE UPDATED TO THE JOYSTICK POSITION, AND ANY PENDING KEYBOARD KEY IS SAVED IN K\$.

BY TAKING A TWO VECTOR SEGMENT OF AN IMAGE STRING, AND USING THAT WITH VERTEX, YOU CAN SEE WHAT AN INSERTED POINT BETWEEN THEM WOULD LOOK LIKE.

ABSOLUTE CROSS:

CALL "ACROSS",C,X,Y

CALL "ACROSS",NUM.ARG,NUM.ARG,NUM.ARG

C >= +0.5 DRAW CROSSHAIR ABS(C) TIMES IN STORAGE MODE.

C <= -0.5 DRAW CROSSHAIR ABS(C) TIMES IN REFRESH MODE.

DRAW CROSSHAIR CURSOR C TIMES AT POSITION X,Y.

RELATIVE CROSS:

CALL "RCROSS",C,X,Y

CALL "RCROSS",NUM.ARG,NUM.ARG,NUM.ARG

C >= +0.5 DRAW CROSSHAIR ABS(C) TIMES IN STORAGE MODE.

C <= -0.5 DRAW CROSSHAIR ABS(C) TIMES IN REFRESH MODE.

DRAW CROSSHAIR CURSOR C TIMES AT X,Y POSITION RELATIVE TO THE DISPLAY CENTER POSITION.

GRAPHICS CROSS:

CALL "GCROSS",C,X,Y

CALL "GCROSS",NUM.ARG,NUM.ARG,NUM.ARG

C >= +0.5 DRAW CROSSHAIR ABS(C) TIMES IN STORAGE MODE.

C <= -0.5 DRAW CROSSHAIR ABS(C) TIMES IN REFRESH MODE.

DRAW CROSSHAIR CURSOR C TIMES AT X,Y RELATIVE TO THE GRAPHIC CURSOR POSITION.

JOYSTICK CROSS:

CALL "JCROSS",C,X,Y,K\$

CALL "JCROSS",NUM.ARG&RES,NUM.RES,NUM.RES,STR.RES

C >= +0.5 DRAW CROSSHAIR ABS(C) TIMES IN STORAGE MODE.

C <= -0.5 DRAW CROSSHAIR ABS(C) TIMES IN REFRESH MODE.

DRAW CROSSHAIR JOYSTICK CURSOR C TIMES, OR UNTIL A KEYBOARD KEY IS PRESSED, WHICHEVER OCCURS FIRST.

AFTER THE COMMAND EXIT: C = ACTUAL DISPLAY COUNT, X AND Y ARE UPDATED TO THE JOYSTICK POSITION, AND ANY PENDING KEYBOARD KEY IS SAVED IN K\$.

ABSOLUTE DRAW:

CALL "ADRAW",I\$,C,X,Y

CALL "ADRAW",IMG.ARG,NUM.ARG,NUM.ARG,NUM.ARG

C >= +0.5 DRAW IMAGE ABS(C) TIMES IN STORAGE MODE.

C <= -0.5 DRAW IMAGE ABS(C) TIMES IN REFRESH MODE.

DRAW VECTOR IMAGE I\$ C TIMES AT POSITION X,Y.

RELATIVE DRAW:

CALL "RDRAW",I\$,C,X,Y

CALL "RDRAW",IMG.ARG,NUM.ARG,NUM.ARG,NUM.ARG

C >= +0.5 DRAW IMAGE ABS(C) TIMES IN STORAGE MODE.

C <= -0.5 DRAW IMAGE ABS(C) TIMES IN REFRESH MODE.

DRAW VECTOR IMAGE I\$ C TIMES AT X,Y RELATIVE TO THE IMAGE START POSITION.

GRAPHICS DRAW:

CALL "GDRAW",I\$,C,X,Y

CALL "GDRAW",IMG.ARG,NUM.ARG,NUM.ARG,NUM.ARG

C >= +0.5 DRAW IMAGE ABS(C) TIMES IN STORAGE MODE.

C <= -0.5 DRAW IMAGE ABS(C) TIMES IN REFRESH MODE.

DRAW VECTOR IMAGE I\$ C TIMES AT X,Y RELATIVE TO THE GRAPHIC CURSOR POSITION.

JOYSTICK DRAW:

CALL "JDRAW", I\$, C, X, Y, K\$

CALL "JDRAW", IMG.ARG, NUM.ARG&RES, NUM.RES, NUM.RES, STR.RES

C >= +0.5 DRAW CROSSHAIR ABS(C) TIMES IN STORAGE MODE.

C <= -0.5 DRAW CROSSHAIR ABS(C) TIMES IN REFRESH MODE.

DRAW VECTOR IMAGE I\$ C TIMES AT JOYSTICK CURSOR POSITION OR UNTIL A KEYBOARD KEY IS PRESSED, WHICHEVER OCCURS FIRST.

AFTER THE COMMAND EXIT: C = ACTUAL DISPLAY COUNT, X AND Y ARE UPDATED TO THE JOYSTICK POSITION, AND ANY PENDING KEYBOARD KEY IS SAVED IN K\$.

ABSOLUTE DOTS:

CALL "ADOTS", I\$, C, X, Y

CALL "ADOTS", IMG.ARG, NUM.ARG, NUM.ARG, NUM.ARG

C >= +0.5 DRAW IMAGE ABS(C) TIMES IN STORAGE MODE.

C <= -0.5 DRAW IMAGE ABS(C) TIMES IN REFRESH MODE.

DRAW VECTOR IMAGE I\$ DOTS C TIMES AT POSITION X, Y.

RELATIVE DOTS:

CALL "RDOTS", I\$, C, X, Y

CALL "RDOTS", IMG.ARG, NUM.ARG, NUM.ARG, NUM.ARG

C >= +0.5 DRAW IMAGE ABS(C) TIMES IN STORAGE MODE.

C <= -0.5 DRAW IMAGE ABS(C) TIMES IN REFRESH MODE.

DRAW VECTOR IMAGE I\$ DOTS C TIMES AT X, Y RELATIVE TO THE IMAGE START POSITION.

GRAPHICS DOTS:

CALL "GDOTS", I\$, C, X, Y

CALL "GDOTS", IMG.ARG, NUM.ARG, NUM.ARG, NUM.ARG

C >= +0.5 DRAW IMAGE ABS(C) TIMES IN STORAGE MODE.

C <= -0.5 DRAW IMAGE ABS(C) TIMES IN REFRESH MODE.

DRAW VECTOR IMAGE I\$ DOTS C TIMES AT X, Y RELATIVE TO THE GRAPHIC CURSOR

POSITION.

JOYSTICK DOTS:

CALL "JDOTS", I\$, C, X, Y, K\$

C >= +0.5 DRAW CROSSHAIR ABS(C) TIMES IN STORAGE MODE.

C <= -0.5 DRAW CROSSHAIR ABS(C) TIMES IN REFRESH MODE.

CALL "JDOTS", IMG.ARG, NUM.ARG&RES, NUM.RES, NUM.RES, STR.RES

DRAW VECTOR IMAGE I\$ DOTS C TIMES AT JOYSTICK CURSOR POSITION, OR UNTIL A KEYBOARD KEY IS PRESSED, WHICHEVER OCCURS FIRST.

AFTER THE COMMAND EXIT: C = ACTUAL DISPLAY COUNT, X AND Y ARE UPDATED TO THE JOYSTICK POSITION, AND ANY PENDING KEYBOARD KEY IS SAVED IN K\$.

ABSOLUTE PRINT:

CALL "APRINT", C\$, C, X, Y

CALL "APRINT", STR.ARG, NUM.ARG, NUM.ARG, NUM.ARG

C >= +0.5 PRINT IMAGE ABS(C) TIMES IN STORAGE MODE.

C <= -0.5 PRINT IMAGE ABS(C) TIMES IN REFRESH MODE.

PRINT CHARACTER STRING C\$ C TIMES AT POSITION X, Y.

RELATIVE PRINT:

CALL "RPRINT", C\$, C, X, Y

CALL "RPRINT", STR.ARG, NUM.ARG, NUM.ARG, NUM.ARG

C >= +0.5 PRINT CHARACTER STRING ABS(C) TIMES IN STORAGE MODE.

C <= -0.5 PRINT CHARACTER STRING ABS(C) TIMES IN REFRESH MODE.

PRINT CHARACTER STRING C\$ C TIMES AT X, Y RELATIVE TO THE DISPLAY CENTER POSITION.

GRAPHICS PRINT:

CALL "GPRINT", C\$, C, X, Y

CALL "GPRINT", STR.ARG, NUM.ARG, NUM.ARG, NUM.ARG

C >= +0.5 PRINT CHARACTER STRING ABS(C) TIMES IN STORAGE MODE.

C <= -0.5 PRINT CHARACTER STRING ABS(C) TIMES IN REFRESH MODE.

PRINT CHARACTER STRING C\$ C TIMES AT X,Y RELATIVE TO THE GRAPHIC CURSOR POSITION.

JOYSTICK PRINT:

CALL "JPRINT",C\$,C,X,Y,K\$

CALL "JPRINT",STR.ARG,NUM.ARG&RES,NUM.RES,NUM.RES,STR.RES

C >= +0.5 PRINT CHARACTER STRING ABS(C) TIMES IN STORAGE MODE.

C <= -0.5 PRINT CHARACTER STRING ABS(C) TIMES IN REFRESH MODE.

PRINT CHARACTERS C\$ C TIMES AT JOYSTICK CURSOR POSITION OR UNTIL A KEYBOARD KEY IS PRESSED, WHICHEVER OCCURS FIRST.

AFTER THE COMMAND EXIT: C = ACTUAL DISPLAY COUNT, X AND Y ARE UPDATED TO THE JOYSTICK POSITION, AND ANY PENDING KEYBOARD KEY IS SAVED IN K\$.

ABSOLUTE INPUT:

CALL "AINPUT",C\$,C,X,Y,K\$

CALL "AINPUT",STR.ARG,NUM.ARG&RES,NUM.ARG,NUM.ARG,STR.RES

C >= +0.5 PRINT CHARACTER STRING ABS(C) TIMES IN STORAGE MODE.

C <= -0.5 PRINT CHARACTER STRING ABS(C) TIMES IN REFRESH MODE.

PRINT CHARACTER STRING C\$ AND ANY KEYBOARD CHARACTERS C TIMES OR UNTIL A CARRIAGE RETURN IS PRESSED, WHICHEVER OCCURS FIRST, AT POSITION X,Y.

AFTER INPUT COMMAND EXIT: C = ACTUAL DISPLAY COUNT, ANY PENDING KEYBOARD CHARACTERS ARE SAVED IN K\$.

IF MORE THAN 28 CHARACTERS ARE ENTERED THEN BEEP THE BELL, SINCE THE KEYBOARD BUFFER IS LIMITED TO STORING A MAXIMUM OF 28 CHARACTERS.

RELATIVE INPUT:

CALL "RINPUT",C\$,C,X,Y,K\$

C >= +0.5 PRINT CHARACTER STRING ABS(C) TIMES IN STORAGE MODE.

C <= -0.5 PRINT CHARACTER STRING ABS(C) TIMES IN REFRESH MODE.

CALL "RINPUT",STR.ARG,NUM.ARG&RES,NUM.ARG,NUM.ARG,STR.RES

PRINT CHARACTER STRING C\$ AND ANY KEYBOARD CHARACTERS C TIMES OR UNTIL A CARRIAGE RETURN IS PRESSED, WHICHEVER OCCURS FIRST, AT X,Y RELATIVE TO DISPLAY CENTER POSITION.

AFTER INPUT COMMAND EXIT: C = ACTUAL DISPLAY COUNT, ANY PENDING KEYBOARD CHARACTERS ARE SAVED IN K\$.

IF MORE THAN 28 CHARACTERS ARE ENTERED THEN BEEP THE BELL, SINCE THE KEYBOARD BUFFER IS LIMITED TO STORING A MAXIMUM OF 28 CHARACTERS.

GRAPHICS INPUT:

CALL "GINPUT",C\$,C,X,Y,K\$

C >= +0.5 PRINT CHARACTER STRING ABS(C) TIMES IN STORAGE MODE.
C <= -0.5 PRINT CHARACTER STRING ABS(C) TIMES IN REFRESH MODE.

CALL "GINPUT",STR.ARG,NUM.ARG&RES,NUM.ARG,NUM.ARG.STR.RES

PRINT CHARACTER STRING C\$ AND ANY KEYBOARD CHARACTERS C TIMES OR UNTIL A CARRIAGE RETURN IS PRESSED, WHICHEVER OCCURS FIRST, AT X,Y RELATIVE TO GRAPHIC CURSOR POSITION.

AFTER INPUT COMMAND EXIT: C = ACTUAL DISPLAY COUNT, ANY PENDING KEYBOARD CHARACTERS ARE SAVED IN K\$.

IF MORE THAN 28 CHARACTERS ARE ENTERED THEN BEEP THE BELL, SINCE THE KEYBOARD BUFFER IS LIMITED TO STORING A MAXIMUM OF 28 CHARACTERS.

JOYSTICK INPUT:

CALL "JINPUT",C\$,C,X,Y,K\$

C >= +0.5 PRINT CHARACTER STRING ABS(C) TIMES IN STORAGE MODE.
C <= -0.5 PRINT CHARACTER STRING ABS(C) TIMES IN REFRESH MODE.

CALL "JINPUT",STR.ARG,NUM.ARG&RES,NUM.RES,NUM.RES,STR.RES

PRINT CHARACTER STRING C\$ AND ANY KEYBOARD CHARACTERS C TIMES OR UNTIL A CARRIAGE RETURN IS PRESSED, WHICHEVER OCCURS FIRST, AT JOYSTICK CURSOR POSITION.

AFTER INPUT COMMAND EXIT: C = ACTUAL DISPLAY COUNT, X AND Y ARE UPDATED TO THE JOYSTICK POSITION, AND ANY PENDING KEYBOARD CHARACTERS ARE SAVED IN K\$.

IF MORE THAN 28 CHARACTERS ARE ENTERED THEN BEEP THE BELL, SINCE THE KEYBOARD BUFFER IS LIMITED TO STORING A MAXIMUM OF 28 CHARACTERS.

IMAGES:

CALL "IMAGES",I\$

CALL "IMAGES",IMG.RES

INPUT MAG TAPE ASCII FILE INTO IMAGE STRING I\$.

EXIT OCCURS WHEN EOF IS ENCOUNTERED OR WHEN IMAGE STRING IS FULL.

NO CHARACTERS ARE LOST IF THE IMAGE STRING IS FULL AND THE CALL IS REPEATED WHEN THE DIMENSIONED LENGTH IS 0 MOD 3 TO ENSURE THAT NO PARTIAL VECTORS ARE READ.

CHANGE:

CALL "CHANGE", F, I\$

CALL "CHANGE", ARG, STR.RES

CHANGE FLOATING POINT IMAGE ARRAY F INTO IMAGE STRING I\$.
NEGATIVE VALUES IN ARRAY F INDICATE MOVES.

CALL "CHANGE", I\$, F

CALL "CHANGE", STR.ARG, RES.RES

CHANGE IMAGE STRING I\$ INTO FLOATING POINT IMAGE ARRAY F.
NEGATIVE VALUES IN ARRAY F INDICATE MOVES.

DEFINE POINT:

CALL "DEFINE", I\$, N, X, Y

CALL "DEFINE", IMG.ARG&RES, NUM.ARG&RES, NUM.ARG, NUM.ARG

DEFINE VECTOR IMAGE I\$ POINT N X, Y POSITION.
VECTOR IMAGE I\$ POINT N MUST EXIST PRIOR TO THE CALL.
SIGN(N) INDICATES IF THE IMAGE VECTOR IS A MOVE OR A DRAW.
-0.5 < N < 0.5 ALLOWS FOR IMAGE VECTOR ZERO TO BE SET TO A MOVE OR DRAW.

LOCATE POINTS:

CALL "POINTS", I\$, N, X, Y

CALL "POINTS", IMG.ARG, NUM.ARG&RES, NUM.RES, NUM.RES

FIND VECTOR IMAGE I\$ POINT N X, Y POSITION.
VECTOR IMAGE I\$ POINT N MUST EXIST PRIOR TO THE CALL.
SIGN(N) INDICATES IF THE IMAGE VECTOR IS A MOVE OR A DRAW.
THE FIRST VECTOR IN AN IMAGE STRING IS NEARLY ALWAYS A MOVE, AND THE
DEFINE CALL CAN FORCE IT INTO EITHER A MOVE OR DRAW STATE.

TOGGLE MOVE FLAG:

CALL "TOGGLE", I\$, N

CALL "TOGGLE", IMG.ARG&RES, NUM.ARG&RES

TOGGLE VECTOR IMAGE STRING I\$ POINT N MOVE FLAG.
SIGN(N) INDICATES IF THE IMAGE VECTOR A MOVE OR A DRAW.
A DRAW IS CHANGED TO A MOVE AND A MOVE IS CHANGED TO A DRAW.

ABSOLUTE MOVE:

```
CALL "AMOVE", I$, X, Y  
CALL "AMOVE", IMG.ARG&RES, NUM.ARG, NUM.ARG  
MOVE VECTOR IMAGE I$ STARTING POSITION TO X, Y.
```

RELATIVE MOVE:

```
CALL "RMOVE", I$, X, Y  
CALL "RMOVE", IMG.ARG&RES, NUM.ARG, NUM.ARG  
MOVE VECTOR IMAGE I$ STARTING POSITION BY DELTA X, Y.
```

GRAPHICS MOVE:

```
CALL "GMOVE", I$, X, Y  
CALL "GMOVE", IMG.ARG&RES, NUM.ARG, NUM.ARG  
MOVE VECTOR IMAGE I$ STARTING POSITION BY DELTA X, Y RELATIVE TO THE GRAPHIC  
CURSOR POSITION.
```

JOYSTICK MOVE:

```
CALL "JMOVE", I$, C, X, Y, K$  
CALL "JMOVE", IMG.ARG, NUM.ARG&RES, NUM.RES, NUM.RES, STR.RES  
C >= +0.5    DRAW IMAGE ABS(C) TIMES IN STORAGE MODE.  
C <= -0.5    DRAW IMAGE ABS(C) TIMES IN REFRESH MODE.  
  
CROSSHAIR JOYSTICK CURSOR WILL BE DRAWN C TIMES OR UNTIL A KEYBOARD KEY IS  
PRESSED, WHICHEVER OCCURS FIRST, THEN MOVE VECTOR IMAGE I$ TO JOYSTICK  
CURSOR POSITION.  
AFTER THE COMMAND EXIT: C = ACTUAL DISPLAY COUNT, X AND Y ARE UPDATED TO  
THE JOYSTICK POSITION, AND ANY PENDING KEYBOARD KEY IS SAVED IN K$.
```

ABSOLUTE SCALE:

```
CALL "ASCALE", I$, H, V, X, Y  
CALL "ASCALE", IMG.ARG&RES, NUM.ARG, NUM.ARG, NUM.ARG, NUM.ARG  
SCALE VECTOR IMAGE I$ BY FACTORS H, V AROUND POSITION X, Y.  
NEGATIVE VALUES FOR H OR V RESULT IN A MIRROR IMAGE SCALE.
```

-65.535 =< SCALE FACTOR RANGE <= +65.535 (17 BIT RESOLUTION).

RELATIVE SCALE:

CALL "RSCALE", I\$, H, V, X, Y

CALL "RSCALE", IMG.ARG&RES, NUM.ARG, NUM.ARG, NUM.ARG, NUM.ARG

SCALE VECTOR IMAGE I\$ BY FACTORS H, V AROUND RELATIVE POSITION X, Y.

NEGATIVE VALUES FOR H OR V RESULT IN A MIRROR IMAGE SCALE.

-65.535 =< SCALE FACTOR RANGE <= +65.535 (17 BIT RESOLUTION).

GRAPHICS SCALE:

CALL "GSCALE", I\$, H, V, X, Y

CALL "GSCALE", IMG.ARG&RES, NUM.ARG, NUM.ARG, NUM.ARG, NUM.ARG

SCALE VECTOR IMAGE I\$ BY FACTORS H, V AROUND X, Y RELATIVE TO THE GRAPHIC CURSOR POSITION.

NEGATIVE VALUES FOR H OR V RESULT IN A MIRROR IMAGE SCALE.

-65.535 =< SCALE FACTOR RANGE <= +65.535 (17 BIT RESOLUTION).

JOYSTICK SCALE:

CALL "JSCALE", I\$, C, H, V, X, Y, K\$

CALL "JSCALE", IMG.ARG&RES, NUM.ARG&RES, NUM.ARG, NUM.ARG, NUM.ARG, NUM.ARG, STR.RES

C >= +0.5 DRAW CROSSHAIR ABS(C) TIMES IN STORAGE MODE.

C <= -0.5 DRAW CROSSHAIR ABS(C) TIMES IN REFRESH MODE.

CROSSHAIR JOYSTICK CURSOR WILL BE DRAWN C TIMES OR UNTIL A KEYBOARD KEY IS PRESSED, WHICHEVER OCCURS FIRST, SCALE VECTOR IMAGE I\$ BY FACTORS H, V AROUND THE JOYSTICK CURSOR POSITION.

NEGATIVE VALUES FOR H OR V RESULT IN A MIRROR IMAGE SCALE.

-65.535 =< SCALE FACTOR RANGE <= +65.535 (17 BIT RESOLUTION).

AFTER THE COMMAND EXIT: C = ACTUAL DISPLAY COUNT, X AND Y ARE UPDATED TO THE JOYSTICK POSITION, AND ANY PENDING KEYBOARD KEY IS SAVED IN K\$.

ABSOLUTE SHEAR:

CALL "ASHEAR", I\$, H, V, X, Y

CALL "ASHEAR", IMG.ARG&RES, NUM.ARG, NUM.ARG, NUM.ARG, NUM.ARG

SHEAR VECTOR IMAGE I\$ BY FACTORS H, V AROUND POSITION X, Y.

NEGATIVE VALUES FOR H OR V RESULT IN A MIRROR IMAGE SHEAR.
-65.535 =< SHEAR FACTOR RANGE <= +65.535 (17 BIT RESOLUTION).

RELATIVE SHEAR:

CALL "RSHEAR", I\$, H, V, X, Y

CALL "RSHEAR", IMG.ARG&RES, NUM.ARG, NUM.ARG, NUM.ARG, NUM.ARG

SHEAR VECTOR IMAGE I\$ BY FACTORS H, V AROUND RELATIVE POSITION X, Y.
NEGATIVE VALUES FOR H OR V RESULT IN A MIRROR IMAGE SHEAR.
-65.535 =< SHEAR FACTOR RANGE <= +65.535 (17 BIT RESOLUTION).

GRAPHICS SHEAR:

CALL "GSHEAR", I\$, H, V, X, Y

CALL "GSHEAR", IMG.ARG&RES, NUM.ARG, NUM.ARG, NUM.ARG, NUM.ARG

SHEAR VECTOR IMAGE I\$ BY FACTORS H, V AROUND X, Y RELATIVE TO THE GRAPHIC
CURSOR POSITION.
NEGATIVE VALUES FOR H OR V RESULT IN A MIRROR IMAGE SHEAR.
-65.535 =< SHEAR FACTOR RANGE <= +65.535 (17 BIT RESOLUTION).

JOYSTICK SHEAR:

CALL "JSHEAR", I\$, C, H, V, X, Y, K\$

C >= +0.5 DRAW CROSSHAIR ABS(C) TIMES IN STORAGE MODE.
C <= -0.5 DRAW CROSSHAIR ABS(C) TIMES IN REFRESH MODE.

CALL "JSHEAR", IMG.ARG&RES, NUM.ARG&RES, NUM.ARG, NUM.ARG, NUM.ARG, NUM.ARG,
STR.RES

CROSSHAIR JOYSTICK CURSOR WILL BE DRAWN C TIMES OR UNTIL A KEYBOARD KEY IS
PRESSED, WHICHEVER OCCURS FIRST, SHEAR VECTOR IMAGE I\$ BY FACTORS H, V
AROUND THE JOYSTICK CURSOR POSITION.
NEGATIVE VALUES FOR H OR V RESULT IN A MIRROR IMAGE SHEAR.
-65.535 =< SHEAR FACTOR RANGE <= +65.535 (17 BIT RESOLUTION).
AFTER THE COMMAND EXIT: C = ACTUAL DISPLAY COUNT, X AND Y ARE UPDATED TO
THE JOYSTICK POSITION, AND ANY PENDING KEYBOARD KEY IS SAVED IN K\$.

ABSOLUTE TAPER:

CALL "ATAPER", I\$, H, V, X, Y

CALL "ATAPER", IMG.ARG&RES, NUM.ARG, NUM.ARG, NUM.ARG, NUM.ARG

TAPER VECTOR IMAGE I\$ BY FACTORS H,V AROUND POSITION X,Y.
NEGATIVE VALUES FOR H OR V RESULT IN A MIRROR IMAGE TAPER.
-65.535 =< TAPER FACTOR RANGE <= +65.535 (17 BIT RESOLUTION).

RELATIVE TAPER:

CALL "RTAPER", I\$, H, V, X, Y

CALL "RTAPER", IMG.ARG&RES, NUM.ARG, NUM.ARG, NUM.ARG, NUM.ARG

TAPER VECTOR IMAGE I\$ BY FACTORS H,V AROUND RELATIVE POSITION X,Y.
NEGATIVE VALUES FOR H OR V RESULT IN A MIRROR IMAGE TAPER.
-65.535 =< TAPER FACTOR RANGE <= +65.535 (17 BIT RESOLUTION).

GRAPHICS TAPER:

CALL "GTAPER", I\$, H, V, X, Y

CALL "GTAPER", IMG.ARG&RES, NUM.ARG, NUM.ARG, NUM.ARG

TAPER VECTOR IMAGE I\$ C TIMES AT X,Y RELATIVE TO THE GRAPHIC CURSOR
POSITION.
NEGATIVE VALUES FOR H OR V RESULT IN A MIRROR IMAGE TAPER.
-65.535 =< TAPER FACTOR RANGE <= +65.535 (17 BIT RESOLUTION).

JOYSTICK TAPER:

CALL "JTAPER", I\$, C, H, V, X, Y, K\$

CALL "JTAPER", IMG.ARG&RES, NUM.ARG&RES, NUM.ARG, NUM.ARG, NUM.ARG, NUM.ARG,
STR.RES.

C >= +0.5 DRAW CROSSHAIR ABS(C) TIMES IN STORAGE MODE.
C <= -0.5 DRAW CROSSHAIR ABS(C) TIMES IN REFRESH MODE.

CROSSHAIR JOYSTICK CURSOR WILL BE DRAWN C TIMES OR UNTIL A KEYBOARD KEY IS
PRESSED, WHICHEVER OCCURS FIRST, THEN TAPER VECTOR IMAGE I\$ BY FACTORS H,V
AROUND THE JOYSTICK CURSOR POSITION,
NEGATIVE VALUES FOR H OR V RESULT IN A MIRROR IMAGE TAPER.
-65.535 =< TAPER FACTOR RANGE <= +65.535 (17 BIT RESOLUTION).
AFTER THE COMMAND EXIT: C = ACTUAL DISPLAY COUNT, X AND Y ARE UPDATED TO
THE JOYSTICK POSITION, AND ANY PENDING KEYBOARD KEY IS SAVED IN K\$.

ABSOLUTE ROTATE:

CALL "AROTAT", I\$, A, X, Y

CALL "AROTAT", IMG.ARG&RES, NUM.ARG, NUM.ARG, NUM.ARG

ROTATE VECTOR IMAGE I\$ BY ANGLE A AROUND POSITION X,Y.
ALL ANGLES ARE IN THE CURRENT TRIGOMETRIC UNITS.
ALL ANGLES ARE MEASURED COUNTER CLOCKWISE.

RELATIVE ROTATE:

CALL "RROTAT",I\$,A,X,Y

CALL "RROTAT",IMG.ARG&RES,NUM.ARG,NUM.ARG,NUM.ARG

ROTATE VECTOR IMAGE I\$ BY ANGLE A AROUND X,Y RELATIVE TO THE DISPLAY CENTER POSITION.
ALL ANGLES ARE IN THE CURRENT TRIGOMETRIC UNITS.
ALL ANGLES ARE MEASURED COUNTER CLOCKWISE.

GRAPHICS ROTATE:

CALL "GROTAT",I\$,A,X,Y

CALL "GROTAT",IMG.ARG&RES,NUM.ARG,NUM.ARG,NUM.ARG

ROTATE VECTOR IMAGE I\$ BY ANGLE A AROUND X,Y RELATIVE TO THE GRAPHIC CURSOR POSITION.
ALL ANGLES ARE IN THE CURRENT TRIGOMETRIC UNITS.
ALL ANGLES ARE MEASURED COUNTER CLOCKWISE.

JOYSTICK ROTATE:

CALL "JROTAT",I\$,C,A,X,Y,K\$

CALL "JROTAT",IMG.ARG&RES,NUM.ARG&RES,NUM.RES,NUM.RES,STR.RES

C >= +0.5 DRAW CROSSHAIR ABS(C) TIMES IN STORAGE MODE.
C <= -0.5 DRAW CROSSHAIR ABS(C) TIMES IN REFRESH MODE.

CROSSHAIR JOYSTICK CURSOR WILL BE DRAWN C TIMES OR UNTIL A KEYBOARD KEY IS PRESSED, WHICHEVER OCCURS FIRST, THEN ROTATE VECTOR IMAGE I\$ BY ANGLE A AROUND THE JOYSTICK CURSOR POSITION.
ALL ANGLES ARE IN THE CURRENT TRIGOMETRIC UNITS.
ALL ANGLES ARE MEASURED COUNTER CLOCKWISE.
AFTER THE COMMAND EXIT: C = ACTUAL DISPLAY COUNT, X AND Y ARE UPDATED TO THE JOYSTICK POSITION, AND ANY PENDING KEYBOARD KEY IS SAVED IN K\$.

PRINTS:

CALL "PRINTS",C\$

CALL "PRINTS",STR.ARG

PRINTS CHARACTER STRING C\$ (PRINTABLE CONTROL CHARACTERS).

INPUTS:

CALL "INPUTS",C\$

CALL "INPUTS",STR.ARG

ENTER CHARACTER STRING C\$ INTO THE KEYBOARD BUFFER.

IF MORE THAN 28 CHARACTERS ARE ENTERED THEN BEEP THE BELL, SINCE THE KEYBOARD BUFFER IS LIMITED TO STORING A MAXIMUM OF 28 CHARACTERS.

STRING:

CALL "STRING",F,C\$

CALL "STRING",ARY.ARG,STR.RES

COMVERT FLOATING POINT ARRAY F INTO ASCII CHARACTER STRING C\$.
ASCII STRING C\$ WILL BE VALID ASCII CHARACTERS (ASCII 0..127).

CALL "STRING",C\$,F

CALL "STRING",STR.ARG,ARY.RES

COMVERT ASCII CHARACTER STRING C\$ INTO FLOATING POINT ARRAY F.
FLOATING POINT ARRAY F WILL BE VALID ASCII CHARACTER CODES (0..127).

SOUNDS:

CALL "SOUNDS",C\$

CALL "SOUNDS",STR:ARG

OUTPUT ZERO CROSSING PERIODS IN ASCII STRING C\$ TO SPEAKER.

SAMPLE PERIOD = (N+1) * 38.4 uS.

00H = 26042 HZ EDGE RATE (FREQ = 13KHZ).

7FH = 203.5 HZ EDGE RATE (FREQ = 100HZ).

SAMPLE INTERVAL = 32 CYCLES AT 1.2 US/CYCLE.

MUZAKT:

CALL "MUZAKT",C\$

CALL "MUZAKT",STR.ARG

OUTPUT MUSICAL NOTES IN ASCII STRING C\$ TO SPEAKER

MUZAKT STRING SYNTAX:

```
<WORD> ::= <TEMP>
          + <REST>
          + <NOTE>
          + <NOTE><DOT>
          + <NOTE><OCT><DOT>
          + <NOTE><OCT><LEN>
          + <NOTE><OCT><LEN><DOT>
<NOTE> ::= <A + B + C + D + E + F + G>
          + <A + B + C + D + E + F + G><FLAT>
          + <A + B + C + D + E + F + G><SHRP>
<OCT>  ::= <0..7> (0 = FIRST OCTAVE ON STANDARD KEYBOARD).
<LEN>  ::= <1..64> (RECIPROCAL NOTE LENGTH: 1,2,4,8,16,32,64).
<REST> ::= <R><1..64> (RECIPROCAL REST LENGTH: 1,2,4,8,16,32,64).
<TEMP> ::= <T><0..9> (SEE MUZAKT TEMPO TABLE BELOW).
<DOT>  ::= <.> (PERIOD).
<SHRP> ::= <#> (POUND SIGN).
<FLAT> ::= <b> (LOWER CASE b).
```

ANY INVALID WORDS OR CHARACTERS ARE IGNORED. ANY MISSING DESCRIPTORS WILL SIMPLY USE THE PREVIOUSLY DEFINED DESCRIPTOR VALUES. THIS ALSO MEANS THAT PREVIOUSLY DEFINED DESCRIPTOR VALUES MUST EXIST, OR ELSE THE SCORES ARE NOT LIKELY TO SOUND THE WAY THAT YOU WANTED.

TO ENTER THIRDS, FIFTHS, ... USE THREE, FIVE, ... TIMES LENGTH ETC.

TEMPO RATE (BEATS PER MINUTE)

0	80
1	89
2	100
3	110
4	120
5	130
6	140
7	150
8	160
9	170

MUZAKT STRING EXAMPLE:

T4E316G332B4EG5B4E5G5.G516G54F#R

NOTE	OCT	LENGTH
------	-----	--------

TEMPO	4	
E	3	1/16
G	3	1/32

B	4	1/32
E	4	1/32
G	5	1/32
B	4	1/32
E	5	1/32
G	5	DOTTED 1/32ND
G	5	1/16
G	5	1/4
F#	5	1/4
REST		1/4

HANDY CALL PARAMETER CODE REFERENCE:

80: NUMERIC ARRAY ARGUMENT (COMBINABLE WITH NUMERIC ARRAY RESULT)
 40: NUMERIC SIMPLE ARGUMENT (COMBINABLE WITH NUMERIC SIMPLE RESULT)
 20: IMAGE STRING ARGUMENT (COMBINABLE WITH IMAGE STRING RESULT)
 10: CHARACTER STRING ARGUMENT (COMBINABLE WITH CHARACTER STRING RESULT)
 08: NUMERIC ARRAY RESULT (COMBINABLE WITH NUMERIC ARRAY ARGUMENT)
 04: NUMERIC SIMPLE RESULT (COMBINABLE WITH NUMERIC ARRAY ARGUMENT)
 02: IMAGE STRING RESULT (COMBINABLE WITH IMAGE STRING ARGUMENT)
 01: CHARACTER STRING RESULT (COMBINABLE WITH CHARACTER STRING ARGUMENT)
 00: NO CALL PARAMETERS (NEVER USED IN FAST GRAPHICS)

CALL SYNTAX FORMS:

CALL PARAMETER TYPES:

CALL "BOUNDS", I\$, X0, Y0, X1, Y1	20 04 04 04 04
CALL "LOCATE", C, X, Y, K\$	44 04 04 01
CALL "AGIN", X, Y	04 04
CALL "RGIN", X, Y	44 44
CALL "GGIN", X, Y	44 44
CALL "JGIN", T, X, Y, K\$	44 04 04 01
CALL "APOINT", I\$, N, X, Y	20 44 44 44
CALL "RPOINT", I\$, N, X, Y	20 44 44 44
CALL "GPOINT", I\$, N, X, Y	20 44 44 44
CALL "JPOINT", I\$, C, N, X, Y, K\$	20 44 44 04 04 01
CALL "DASHED", H, V, X0, Y0, X1, Y1	40 40 40 40 40 40
CALL "DOTTED", H, V, X0, Y0, X1, Y1	40 40 40 40 40 40
CALL "KABOOM", C, X, Y	40 40 40
CALL "RUBBER", C, X, Y, K\$	44 40 40 01
CALL "VERTEX", I\$, C, X, Y, K\$	20 44 40 40 01
CALL "ACROSS", C, X, Y	40 40 40
CALL "RCROSS", C, X, Y	40 40 40
CALL "GCROSS", C, X, Y	40 40 40
CALL "JCROSS", C, X, Y, K\$	44 04 04 01
CALL "ADRAW", I\$, C, X, Y	20 40 40 40
CALL "RDRAW", I\$, C, X, Y	20 40 40 40
CALL "GDRAW", I\$, C, X, Y	20 40 40 40
CALL "JDRAW", I\$, C, X, Y, K\$	20 44 04 04 01
CALL "ADOTS", I\$, C, X, Y	20 40 40 40
CALL "RDOTS", I\$, C, X, Y	20 40 40 40

CALL "GDOTS", I\$, C, X, Y	20	40	40	40
CALL "JDOTS", I\$, C, X, Y, K\$	20	44	04	04 01
CALL "APRINT", C\$, C, X, Y	10	40	40	40
CALL "RPRINT", C\$, C, X, Y	10	40	40	40
CALL "GPRINT", C\$, C, X, Y	10	40	40	40
CALL "JPRINT", C\$, C, X, Y, K\$	10	44	04	04 01
CALL "AINPUT", C\$, C, X, Y, K\$	10	44	40	40 01
CALL "RINPUT", C\$, C, X, Y, K\$	10	44	40	40 01
CALL "GINPUT", C\$, C, X, Y, K\$	10	44	40	40 01
CALL "JINPUT", C\$, C, X, Y, K\$	10	44	04	04 01
CALL "IMAGES", I\$	02			
CALL "CHANGE", F, I\$	80	02		
CALL "CHANGE", I\$, F	20	08		
CALL "DEFINE", I\$, N, X, Y	22	44	40	40
CALL "POINTS", I\$, N, X, Y	20	44	04	04
CALL "TOGGLE", I\$, N	22	44		
CALL "AMOVE", I\$, X, Y	22	40	40	
CALL "RMOVE", I\$, X, Y	22	40	40	
CALL "GMOVE", I\$, X, Y	22	40	40	
CALL "JMOVE", I\$, C, X, Y, K\$	22	44	04	04 01
CALL "ASCALE", I\$, H, V, X, Y	22	40	40	40 40
CALL "RSCALE", I\$, H, V, X, Y	22	40	40	40 40
CALL "GSCALE", I\$, H, V, X, Y	22	40	40	40 40
CALL "JSCALE", I\$, C, H, V, X, Y, K\$	22	44	40	40 04 04 01
CALL "ASHEAR", I\$, H, V, X, Y	22	40	40	40 40
CALL "RSHEAR", I\$, H, V, X, Y	22	40	40	40 40
CALL "GSHEAR", I\$, H, V, X, Y	22	40	40	40 40
CALL "JSHEAR", I\$, C, H, V, X, Y, K\$	22	44	40	40 04 04 01
CALL "ATAPER", I\$, H, V, X, Y	22	40	40	40 40
CALL "RTAPER", I\$, H, V, X, Y	22	40	40	40 40
CALL "GTAPER", I\$, H, V, X, Y	22	40	40	40 40
CALL "JTAPER", I\$, C, H, V, X, Y, K\$	22	44	40	40 04 04 01
CALL "AROTAT", I\$, A, X, Y	22	40	40	40
CALL "RROTAT", I\$, A, X, Y	22	40	40	40
CALL "GROTAT", I\$, A, X, Y	22	40	40	40
CALL "JROTAT", I\$, C, A, X, Y, K\$	22	44	40	04 04 01
CALL "PRINTS", C\$	10			
CALL "INPUTS", C\$	10			
CALL "STRING", F, C\$	40	01		
CALL "STRING", C\$, F	10	04		
CALL "SOUNDS", C\$	10			
CALL "MUZAKT", C\$	10			

BENCHMARKS:

1,500 VECTOR IMAGE:

VECTORS/SECOND SPEED (X)

CHANGE FLOATING POINT TO STRING:	2.8 SECONDS	536	...
CHANGE STRING TO FLOATING POINT:	28.0 SECONDS	54	...

3,200 VECTOR IMAGE (3-D DOTS)

VECTORS/SECOND SPEED (X)

BASIC USING "MOVE/DRAW":	175 SECONDS	18	REF
BASIC USING "PRINT AT:":	110 SECONDS	29	1.6
BASIC USING "MOVE/DRAW":	79.0 SECONDS (4052*)	41	2.2
EXTENDED BASIC "QDRAW ":	36 SECONDS	89	4.9 (2.2*)
FAST GRAPHICS "RDRAW ":	12.8 SECONDS	250	13.7 (6.2*)
(4051 RDOTS EQUIVALENT REQUIRES A MOVE AND A DRAW AT EVERY POINT)			
FAST GRAPHICS "RDOTS ":	2.8 SECONDS	1145	125 (55*)
FAST GRAPHICS "RMOVE ":	1.9 SECONDS	1684	...
FAST GRAPHICS "APOINT":	5.9 SECONDS	542	...
FAST GRAPHICS "ASCALE":	7.3 SECONDS	438	...
FAST GRAPHICS "BOUNDS":	1.3 SECONDS	2462	...

3,300 VECTOR IMAGE (DEATH STAR) VECTORS/SECOND SPEED(X)

BASIC USING "MOVE/DRAW":	185 SECONDS	18	REF
BASIC USING "PRINT AT:":	110 SECONDS	25	1.7
BASIC USING "MOVE/DRAW":	79.0 SECONDS (4052*)	37	2.2
EXTENDED BASIC "QDRAW ":	36 SECONDS	135	4.9
FAST GRAPHICS "RDRAW ":	12.8 SECONDS	260	13.7 (7.0*)
(4051 RDOTS EQUIVALENT REQUIRES A MOVE AND A DRAW AT EVERY POINT)			
FAST GRAPHICS "RDOTS ":	2.8 SECONDS	1132	127 (61*)
FAST GRAPHICS "RMOVE ":	1.9 SECONDS	1650	...
FAST GRAPHICS "APOINT":	5.9 SECONDS	541	...
FAST GRAPHICS "ASCALE":	7.3 SECONDS	446	...
FAST GRAPHICS "BOUNDS":	1.3 SECONDS	2538	...

4,400 VECTOR IMAGE (GOTHIC FONT) VECTORS/SECOND SPEED(X)

(IMAGE TOO LARGE FOR 4051 FLOATING POINT)

BASIC USING "MOVE/DRAW":	80 SECONDS (4052*)	55	REF
EXTENDED BASIC "QDRAW ":	26.4 SECONDS	165	(3.0*)
FAST GRAPHICS "RDRAW ":	17.8 SECONDS	246	(4.5*)
(4052 RDOTS EQUIVALENT REQUIRES A MOVE AND A DRAW AT EVERY POINT)			
FAST GRAPHICS "RDOTS ":	3.25 SECONDS	1112	(40*)

10,000 VECTOR IMAGE (TIGER) VECTORS/SECOND SPEED(X)

(IMAGE TOO LARGE FOR 4051 FLOATING POINT)

EXTENDED BASIC "QDRAW ":	50.0 SECONDS	200	4.9
FAST GRAPHICS "RDRAW ":	36.6 SECONDS	273	13.7 (7.0*)
FAST GRAPHICS "RDOTS ":	8.5 SECONDS	1177	127 (61*)
FAST GRAPHICS "RMOVE ":	5.5 SECONDS	1818	...
FAST GRAPHICS "APOINT":	18.0 SECONDS	556	...
FAST GRAPHICS "ASCALE":	21.5 SECONDS	465	...
FAST GRAPHICS "BOUNDS":	3.4 SECONDS	2941	...

ASCII CHARACTER STRINGS (3200 CHARACTERS) CHARACTERS/SECOND

CHARACTER PRINTING RATE:	>250	...
CHANGE ASCII TO CHARACTER:	2.1 SECONDS	1524
CHANGE CHARACTER TO ASCII:	4.1 SECONDS	750

ACKNOWLEDGEMENTS:

KURT KRUEGER WROTE THE ORIGINAL VERSION OF MUZAKT IN 1978.
THE FAST GRAPHICS VERSION HAS ADDITIONAL FEATURES THAT CARL HOVEY WANTED.
I HAVE ALWAYS BEEN PARTIALLY HEARING IMPAIRED AND HAVE NO MUSICAL ABILITY.

CARL HOVEY DEMONSTRATED THE FEASIBILITY OF 4051 WRITE-THRU (NON-STORE)
GRAPHICS IN A PINBALL GAME IN 1978.