DECUS
PROGRAM LIBRARY

DECUS NO.	FOCAL8-154
TITLE	8K FOCAL DISPLAY
AUTHOR	Floor Anthony
COMPANY	Medical Biological Laboratory National Defence Research Council TNO The Netherlands
DATE	December 1970
SOURCE LANGUAGE	

ERRATA SHEET

The following errors should be corrected in the 8K FOCAL DISPLAY package:(FOCAL8-154)

1) Corrections to the program "LAZY PLOT"

*Page 19, line 8:

"set to 4" becomes " set to 4 + 1 = 5"

"An extra space is used to stabilize the electron beam."

*Page 21, first line of listing:

"01.01 . . . . . ; S XC = 3; . . . . . ." shoula be changed to:

"01.01 . . . . . ; S XC = 4; . . . . . ."

*Page 21, 8th line from bottom:

"29.60 . . . . . ; T XN + 1 * . . . . ." should be changed to:

"29.60 . . . . . ; T " ", XN + 1 * . ."

*Page 21, 2nd line from bottom:

"30.60 . . . . . ; T YN + 1 * . . . . ." should be changed to:

"30.60 . . . . . ; T " ", YN + 1 * . ."

2) Correction to 8K FOCAL DISPLAY MANUAL

Page 9, line 12 from bottom of page:

"200" should read "200 write-through 1 = write-through intensity; Ø = normal"

3) Correction to the program itself

A bug, due to a typing error still exists. It causes a destruction of the program when the kxtbuffer reaches its upper limit. It then destroys one location of the program. The correction is as follows:

*AXOUT D-1

2571 1047 7600-FST + 1

location 02571 (1046) becomes 1047

8K FOCAL DISPLAY

DECUS Program Library Write-up

DECUS NO. FOCAL8-154

8k FOCAL DISPLAY is an overlay for 8k FOCAL which provides the experienced user with expanded plotting capabilities on the KV8/i and VTO 1 storage tube display. Two new functions FDIS, and FX incorporate all the possibilities of vector-plotting (relative, absolute, circles, arcs) and character display. (Several sizes of characters possible). A dashed line feature is also implemented. An EDIT SCREEN command allows characters to be "typed" onto the screen directly, to the left of the cursor which marks the spot where the text has to appear.
Direct control keys control instant switching from "scope mode" to "teletype mode" and vice versa.
Program control by use of the cursor and interrupt bar enables interactive programming.
All existing functions remain untouched, but can be omitted with the key CTRL/V (EXPAND VARIABLES), giving room for 201 variables, instead of 128.
The program occupies loc. 3140-3377 and 16532-17577.

Software required: FOCAL 69 and 8k overlay.

Minimum hardware: 8k PDP8/i with KV8/i and VTO1 storage tube.

1) Introduction

In computer-graphics it is necessary to be able to display both lines and characters for a good picture. "8K FOCAL DISPLAY" will provide these possibilities to the experienced FOCAL user.

Two new functions (FX and FDIS) transfer information from the FOCAL program to the display routines. The function FX is used for the drawing of vectors and arcs. The arguments of this function will set values in the analog computer part of the KV8/I interface. The function will also be used for CURSOR-readout. The Cursor is the small oval spot that can be moved with the joystick.

The function FDIS was used to switch the output of FOCAL to the character-generating routines. The TYPE and ASK commands will then cause characters to be written on the screen.

The interrupt-bar on the joystick panel causes a real program interrupt which sets up conditions to effectuate a "DO 31" at the end of a program line.

2) The Screen

The size of the screen is 16.5 by 21 cm. The mathematical origin (X=O, Y=O) is situated in the centre of the screen. X may assume values between -280 and +280; y between -330 and +330. In most displays a slight de-focus will occur at the edges of the screen (see drawing page 2a).

3) Vectors and Arcs

The KV8/I draws a vector from point A to point B in a fixed time. The farther the two points lie apart, the greater the writing speed, and the lower the writing-intensity. To cope with this problem there are two kinds of vectors: long vectors (4 milliseconds) and short vectors (0.25 milliseconds).

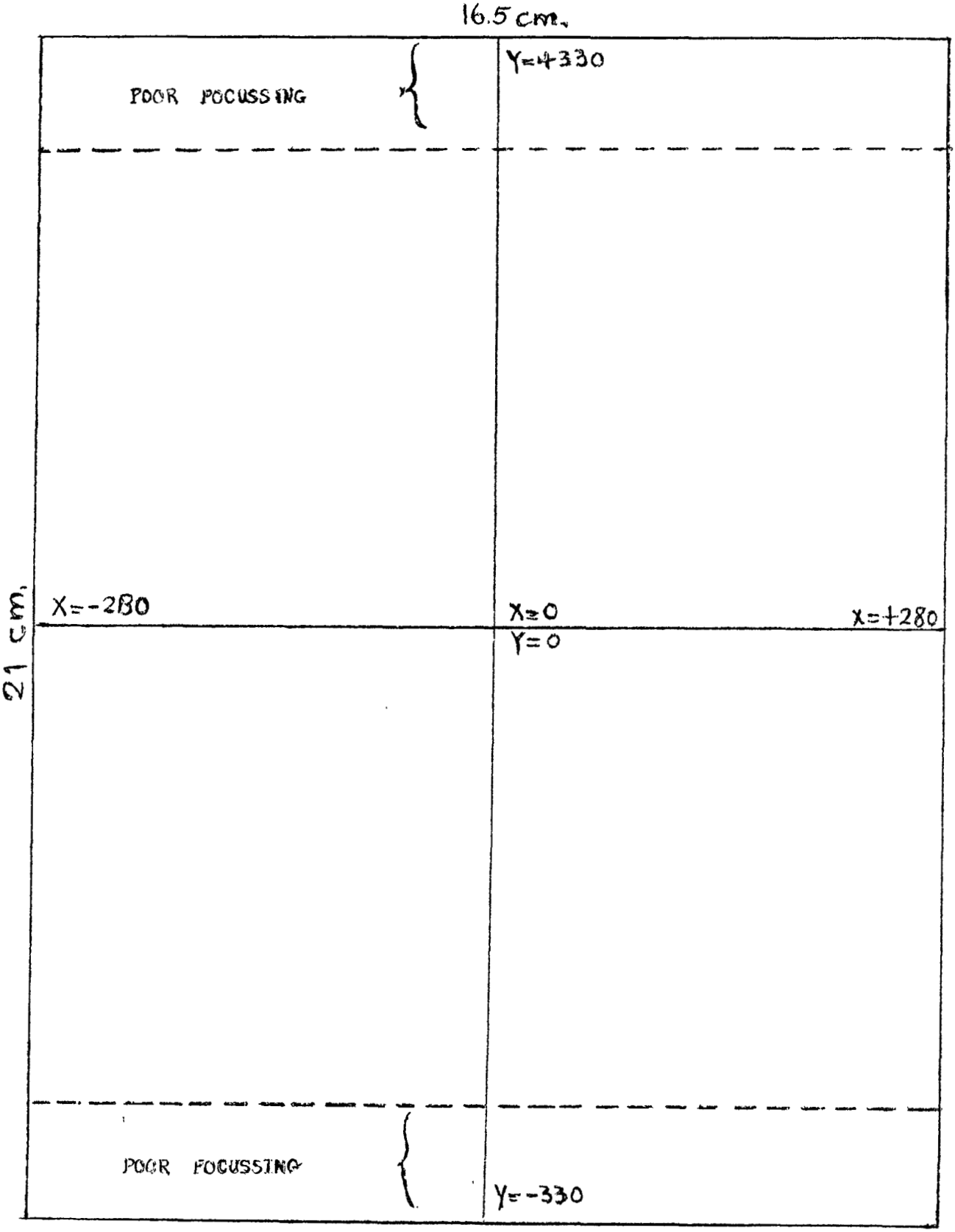
A long vector will be used for lines of which the length is more than 20 points or when undetermined, while short vectors will be used for characters and symbols.

3a) Straight Vectors

SET Z=FX(ARC, MODE, XO, YO, X1, Y1 . . . . . .XN, YN)
is the general format of the function FX.

Z is an arbitrary variable which will be set at zero after evaluation of the function. ARC is used for the timing of vector-continue vectors. The timing of this kind of vectors is such that a full circle corresponds to a value of ARC of 360 (16 millisec.). If ARC=90 a long vector will be timed or an arc of 90 degrees.

MODE must have a value corresponding to the kind of vector wanted to be drawn. (See chapter 8) XO and YO are the coordinates of the beginning of the vector, X1 and Y1 those of the end of the first segment. Segments are chained head-to-tail until the right parenthesis is encountered. All arguments may be constants, variables or expressions. An FX may not be nested within an FX.


16.5 cm.
21 cm.
POOR FOCUSSING
Y=+330
X=-280
X=0
X=+280
Y=0
POOR FOCUSSING
Y=-330

Example: SET Z=FX (,289,0,0,100,100) draws a long vector (code 289) from (0,0 to (100,100).

Example: 1.04 FOR X=-200,50,200; SET Z=FX(,289,X,0,X+50,0,X+50,4) draws an X-axis with 9 vertical marks.

3b) Dashed Vectors

As dashed vectors are not hardware-implemented in the KV8/I interface, a software routine turns the intensity on and off at timing intervals given by the value of ARC.

Example: Set Z=FX(3,369,0,0,200,0) draws a dashed line of 15 visible and 15 invisible segments of 3 counts each (total 90 counts=one long vector) between (0,0) and (200,0), beginning with a visible segment. SET Z=FX(3,368,0,0,200,0) does the same but starts with a blank segment (see chapter 10 for explanation). In the case of an odd number of segments (90/6=15, 90/10=9, 90/2=45) the vector will start and end with the same type segment. It is necessary to select arc so that 90 can be divided by it (1,2,3,5,6,9,10,15,30,45).

3c) Thick and Thin Points

A point will be written when the beginning and the end of the vector coincide.

Example: SET Z=FX(,MODE,XO,YO,XO,YO)

If MODE=289 (long vector) a thick spot will be written at (XO,YO)
If MODE=257 (short vector) a thin spot will be written at (XO,YO)
If MODE=513 (point plot) a nearly invisible spot will be written

In "point plot"-mode no vectors are drawn. Instead 20 microsec. points are written at the ends of each segment.

3d) Erasing the Screen

The command string SET Z=FX (,6,,,,) will cause erasure of the screen. Values 4,6,36 for MODE are also valid (see chapter 10).

3e) Circles and Arcs

Example: 1.04 SET Z=FX (180,281,-100,0,0,0) writes an arc of 180 degrees from the point (-100,0) clockwise around the centre (0,0).

Circles and arcs will always write clockwise as this is a property of the analog part of the KV8/I interface.

Example: 1.08 SET Z=FX (180,281,-100,0,0,0,50,0,25,0,37,0) writes a spiral of head-to-tail circle segments of 180 degrees each. It starts at (-100,0) and turns around the centres (0,0) (50,0) (25,0) and (37,0) consecutively.

Example: SET Z=FX (180,313,X,Y,X+10,Y,X+30,Y,X+50,Y,X+70,Y) draws the electronic solenoid symbol. 313 is also a valid circle mode.

3f) Visualizing the Cursor

As a vector and the cursor-oval cannot be written simultaneously, the cursor will disappear during the execution of the function FX (and FDIS). Although not implemented by hardware it will reappear after each FX (not FDIS).

The cursor can also be visualized with MODE=786.

Example: SET Z=FX(,786,,,,)

After writing on the screen the cursor will disappear. This can be used to obscure the cursor: SET Z=FDIS(1,,) (see chapter 4).

4) Text

4a) FOCAL Output to the Scope

FOCAL generates strings of text with the TYPE and ASK commands. The function FDIS controls the software switches to conduct the output to the TELETYPE or to the scope routines. The general format of FDIS is:

SET Z=FDIS(SCALE,XO,YO), in which SCALE controls character size and XO, YO the position of the beginning of the character string on the screen (origin).

Example: 1.20 SET Z=FDIS(2,-200,0); TYPE "ABCD"!"EFGH" effectuates the appearance of the character group ABCD in character size 2 to the left of the centre of the screen.
EFGH

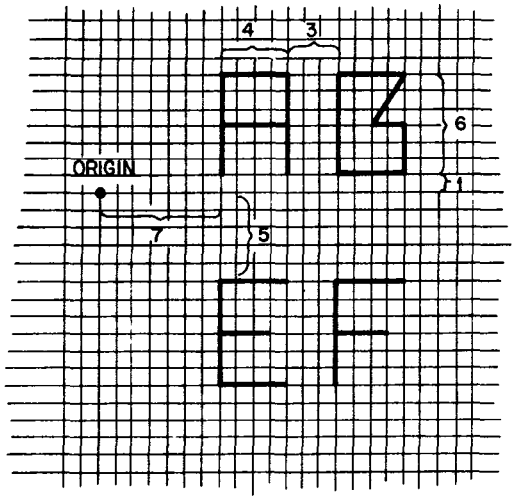
Example: 1.04 FOR X=-200,50,200;SET Z=FX(,289,X,0,X+50,0,X+50,4)
1.08 FOR X=-200,50,200;SET Z=FDIS(2,X-44,-20);TYPE %2,(X+200)/50

writes the values 0,1,2....8 below the X-axis of chapter 3a. It is important to use the correct "%"-format of FOCAL. Note that all FOCAL output will now appear on the screen; also error messages.

FOCAL output can be switched back to the teletype with the function FDIS of format: SET Z=FDIS().

4b) The Character Vector-grid

The smallest character (SCALE=1) is 6 grid elements high and 4 elements wide . Different character sizes are obtained by multiplying the elements in each direction with the scale-factor SCALE. The picture below illustrates the position of the characters with respect to the ORIGIN.



4c) Keyboard-to-scope Editor

Before taking a picture of the screen one may wish to insert comments in the shape of text strings into the picture, text that has not been programmed. The command SET Z=FIDS(-SCALE) enables the user to do so. Now all input from the keyboard will be routed to the scope routines. The user can point with the cursor to the spot where his text has to be inserted. When pressing the key CTRL/O the character-display-ORIGIN will be moved to the position of the cursor. All text, now typed on the keyboard will appear in character size SCALE to the right of the new origin (=cursor position). The following keys can now be used to control size and position of text:

CTRL/O	move the ORIGIN to the cursor
CTRL/Q	increase character size (SCALE=SCALE+1)
CTRL/R	reset character size to size 1 (smallest size)
CTRL/C	normal keyboard interrupt
CTRL/P	PROCEED, go on with the program

The last command is essential because FOCAL will hang in "SCOPE-EDIT"-mode until the CTRL/P key is hit.

5) Cursor and Interrupt-bar

5a) Cursor Read-out

In order to involve the joystick in the program for "interactive" programming, one must be able to read its coordinates. This is achieved with a new format of the function FX:

SET XC=FX(,X); SET YC=FX(,Y). Unlike stated in chapter 3a the function FX will now assume a value and transfer it to any variable: XC and YC will assume the value of the X- and Y-coordinates of the cursor respectively. Note that X and Y in the function are regarded as characters, nat as variables.

Example: Put an asterisk wherever the cursor is.

```
1.04 SET Z=FDIS(2,FX(,X)-18,FX(,Y)-8); TYPE "*"; GOTO 1.04
```

Note that FX may be called within another function, not within itself.

Example: Make a cursor-follower.

```
1.04 SET X=FX(,X); SET Y=FX(,Y); SET Z=FX(,289,XO,YO,X,Y)
1.08 SET XO=X;SET YO=Y;GOTO 1.04
```

For an example of program control with the joystick/interrupt-bar see chapter 5c.

5b) The Interrupt-bar

With the interrupt-bar a FOCAL program can be interrupted at any time to execute a certain group of instructions. Hereafter the program proceeds from where it was interrupted. FOCAL will allow the interrupt to be effectuated between the jump from one statement to the other, by an artificial "DO 31."

Group 31 then contains the statements to be executed in case of an interrupt. Because of the mechanism of the "DO" the program automatically returns to where it came from.

Example: Type the X- and Y-value of the cursor when the interrupt-bar is pressed.

```
01.04 SET Z=FDIS( );SET Z=FX(,790,,,,)
01.08 SET Z=FX(,786,,,,)
01.12 GOTO 1.08
31.04 TYPE %3.00,!,FX(,X)," ",FX(,Y);RETURN
```

1.04 switches the output to the teletype, and erases the screen.
1.08 and 1.12 form a waiting loop, displaying the cursor all the time.
31.04 is the only statement in group 31 and will be executed after a program-interrupt.

Mark that a waiting loop 1.08 SET Z=FX(,786,,,,);GOTO 1.08 cannot be interrupted as FOCAL does not jump from one line to another.

5c) Program Control With Cursor and Interrupt-bar

The possibility to read the position of the cursor can be used for Joystick program control. For example one can examine whether the cursor was in a special area of the screen at the time the interrupt-bar was pressed, and thus render control to that part of the program which was intended to execute the action specified in that area.

Example: A program can be controlled by putting the cursor in the correct segment of a sliced command area with the headings "ERASE SCREEN," "EDIT SCREEN," "A," "B," "C," "D." It then performs the functions so named.

```
01.02 SET Z=FX(,6,,,,)
01.04 SET Z=FX(,289,-300,-310,300,-310)
01.08 FOR X=-300,75,300; SET Z=FX(,289,X,-310,X,-340)
01.10 SET Z=FDIS(1,-280,-330);TYPE "ERASE"!"SCREEN"
01.12 SET Z=FDIS(1,-225,-330);TYPE "EDIT"!"SCREEN"
01.14 SET Z=FDIS(1,-150,-330);TYPE "A"
01.16 SET Z=FDIS(1,-75,-330);TYPE"B"
01.18 SET Z=FDIS(1,0,-330);TYPE "C"
01.20 SET Z=FDIS(1,75,-330);TYPE "D"

02.04 SET Z=FX(,786,,,,)
02.08 GOTO 2.04
02.20 SET Z=FDIS(-1);RETURN
02.40 SET Z=FDIS(1,-260,330);SET Z=FDIS( );QUIT

03.04 SET Z=FDIS( );TYPE "A",!

04.04 SET Z=FDIS( );TYPE "B",!

05.04 SET Z=FDIS( );TYPE "C",!

06.04 SET Z=FDIS( );TYPE "D",!

31.04 SET XC=FX(,X);SET YC=FX(,Y)
31.06 IF (350+YC)2.40;IF (310+YC)31.20;RETURN
31.20 IF (-225-XC)31.22;DO 1;R
31.22 IF (-150-XC)31.24;DO 2.20;R
31.24 IF(-75-XC)31.26;DO 3;R
31.26 IF (-XC) 31.28;DO 4;R
31.28 IF (150-XC)31.30;DO 5;R
31.30 IF (225-XC)31.40;DO 2.40
31.40 DO 2.40
```

6) Special Keys

6a) Keys in FOCAL Command-mode

In order to facilitate scope usage during programming, three extra commands avoid the laborious type-out of FDIS functions:

CTRL/S	switch to scope mode; equivalent to SET Z=FDIS(1,-260,330)
CTRL/T	switch to teletype mode; equivalent to SET Z=FDIS( )
CTRL/U	erase screen, put ORIGIN top left of the screen; equivalent to SET Z=FX(,6,,,,);SET Z=FDIS(1,-260,330)

The commands CTRL/Q and CTRL/R can now also be used to control character size. These control commands are trapped from FOCAL input so that no error messages would occur. For the same reason they cannot be entered into a FOCAL program. They can be issued at any time and in any order when FOCAL is waiting for input.

6b) To Expand the Variables-storage Area

As this display overlay is not written over existing functions, they are still valid (FLOG, FATN, FEXP, FSIN, FCOS), leaving room for 128 variables. Pressing the key CTRL/V will enlarge the variables-area at the cost of the above mentioned functions, giving room for 201 variables. An attempt to use one of the disappeared functions will now cause an error message.

7) Loading and Saving

FOCAL with 8K and DISPLAY is loaded as follows:

1) Load FOCAL 5/69 with initial dialogue; start 200; answer YES,YES.
2) Load 8K overlay
3) Load modified 4-WORD overlay if desired
4) Load "FOCAL 8K DISPLAY" overlay
5) Start 200

Note: The original 4-WORD overlay gives troubles because it writes over the cursor-routines.

Users with a DISK MONITOR system or DECTAPE save the program as follows:

```
. SAVE START ! 4777-7577;200
. SAVE FOCAL ! 0-3377;
. SAVE NULL: 10100,16532-17577;10113
```

Program Calling

```
. FOCAL
. CALL NULL (always)
. CALL PROGRAM (if wished)
. START
```

The 4-WORD overlay can also be loaded after FOCAL has been saved:

```
. FOCAL
. CALL NULL
. CALL PROG
. START
L
```

```
0100
1007          (B)
3425
4617
. LOAD
IN-R:
*
ST=200
↑↑
```

The saving of a FOCAL program has been left unchanged:

. SAVE PROG: 100-(B);10113

8) More Possibilities With the KV8/I

8a) The Structure of MODE

To attain a better insight in the possibilities embedded in the value of MODE, this chapter will explain its structure which in fact is the structure of the microprogrammed instruction issued to the KV8/I interface. Every bit in this one-word instruction has its specific meaning:

Bit (octal)	Meaning	
1	intensify bit	1=visible vector; 0=invisible
2	reset integrators and sample-and-holds	
4	erase	
10	circle mode	1=circle; 0=no circle
20	vector continue	1=software timing necessary; 0=hardware timing
40	timing bit	1=long vector 4 msec; 0=short vector .25 msec
100	dotted vector	1=dotted vector; 0=normal vector
200		
400	vector format	1=absolute vector; 0=relative vector
1000	point mode	1=point plot; 0=vector plot
1400	cursor;display cursor	

Note: Bit 100 is software-implemented. It normally is a spare bit. The combination of 1000 and 400 does not mean "absolute point plot" but "display the cursor." Point plot is never relative.

The following example shows how these bits contribute to the code for MODE:

```
Intensify vector      1
Absolute format     400
Long vector          40

    sum             441   octal=289 decimal which is the code for long vectors.
```

Chapter 10 lists all combinations that make sense.

8b) Relative and Absolute Vectors

In absolute vector mode the arguments XO,YO,.....XN,YN are the exact absolute coordinates of the beginning and end of the segments. In relative vector mode they are treated as an offset from the point where the beam was the last time. It seems as if they are added to the present position of the beam. This can be illustrated by the following example·

```
1.04 SET Z-FX(,289,0,0,0,0)
1.08 SET Z=FX(,33,10,10,-20,0,0,-20,20,0,0,20)
```

1.04 moves the beam in absolute format to the origin; 1.08 writes a square around this point in relative format by adding and subtracting the argument values to and from the current beam position.

NOTE: The "addition" and "subtraction" to and from the current beam position is accomplished by an analogue circuit. It is therefore not recommended to make long chains of relative vectors or with a long waiting period inbetween, as errors will be summed and the circuit is not free from "drift."

Note: Circles can never be drawn in relative format.

9) Practical Examples

9a) Symbols

As symbols have to be written on the screen everywhere, the use of relative format vectors is recommended. The following statements generate symbols which can be scaled with the variable H.

```
01.02 SET Z=FX(,289,,,,); C MOVE BEAM TO ORIGIN
01.04 SET Z=FX(,1,H,H,-2*H,,,-2*H,2*H,,,2*H)
01.08 SET Z=FX(,1,H,H,-2*H,-2*H,H,H,-H,H,2*H,-2*H)
01.10 SET Z-FSQT(3);SET Z=FX(,1,-H,-H* Z/3,2*H,,-H,H*Z,-H,-H*Z)
01.12 SET Z-FX(,1,,H,,-2*H,,H,-H,,2* H,,-H,)
01.14 S Z=FSQT(3);S Z=FX(,1,H,H/Z,-H,-H/Z,-H,H/Z,H,-H/Z,,-2*H/Z)
```

9b) Bold Type Characters

Bold type characters can be written by moving (wiggling) the ORIGIN a little bit and rewriting the character string:

In one direction

```
01.04 FOR I=0,4;SET Z=FDIS(3,-10+I,70);T "ABCD"! "EFGH"
```

In two directions

```
01.08 FOR I=0,4;FOR J=0,4;SET Z=FDIS(3,-10+I,70+J);T "ABCD"! "EFGH"
```

Characters with "depth"

```
01.12 FOR I=0,4;SET Z=FDIS(3,-10+I,70+I);T "ABCD"!"EFGH"
```

9c) Curves

When a curve is approximated by straight line segments, two points (the end of the preceding vector and the end of the next vector) are needed to draw a vector through. It is therefore necessary to assign the value of the previous end of the vector to the beginning of the next vector. The very first vector needs special attention (statement 1.04).

Example: Plot a sinewave from left to right between Y=200 and -200:

```
01.04 SET PI=3.14;SET X0=-60*PI;SET Y0=0
01.08 FOR A=-PI,0.2,PI;SET X=60*A;SET Y=200*FSIN(A);DO 2
01.10 QUIT

02.04 SET Z=FX(,289,X0,Y0,X,Y );SET X0=X;SET Y0=Y
```

The factors 60 and 200 are used for scaling.

When a curve is derived from an array of variables, one can fall back on the previous variable, which is more convenient.

Example: The variables A(1) to A(20) have to be plotted above the X-axis above the values X=1 to 20. The value of A is between 0 and 300; the graph occupies the upper half of the screen.

```
01.10 FOR X=-200,40,200;SET Z=FX(,289,X,10,X+40,10,X+40,16)
01.20 FOR Y=10,100,210;SET Z=FX(,289,-200,Y,-200,Y+100,-194,Y+100)
01.30 FOR I=1,19;SET Z=FX(,289,20*I-220,A(I),20*I-200,A(I+1)
```

1.10 writes an X-axis with 30 graduation marks; 1.20 writes a Y-axis with 3 marks; 1.30 draws the curve.

9d) Dot-Dash-Dot Curves

In some graphs one wishes the connecting vector not to touch its beginning and end, but leave a gap of about 1/10 of its length at both ends. For this purpose one can make effective use of FOCAL's computing power:

The beginning becomes $\left(XO+\frac{XI-XO}{10},\quad YO+\frac{YI-YO}{10}\right)$

The end of the vector: $\left(XI-\frac{XI-XO}{10},\quad YI-\frac{YI-YO}{10}\right)$

9e) Histograms

The histogram presentation is mostly achieved from an array of variables (one variable per histogram block).

Example: Make a histogram of A(1)...A(6) in the first quadrant of the screen.

```
01.04 FOR I=1,6;SET Z=30*I;SET Z=FX(,289,Z-30,,Z-30,A(I),Z,A(I),Z,)
```

A test-set of variables can be created for example by:

```
FOR I=1,6;SET A(I)=180+150*FSIN(I)
```

9f) Hatching Histograms

Horizontal and vertical hatching can easily be done for histograms. Hatching of curves is not so simple.

Example: Hatch the histograms of chapter 9e horizontally. For this purpose we use short vectors (thin lines).

```
01.08 FOR I=1,6;FOR Y=0,3,A(I);SET Z=FX(,257,30*(I-1),Y,30*I,Y)
```

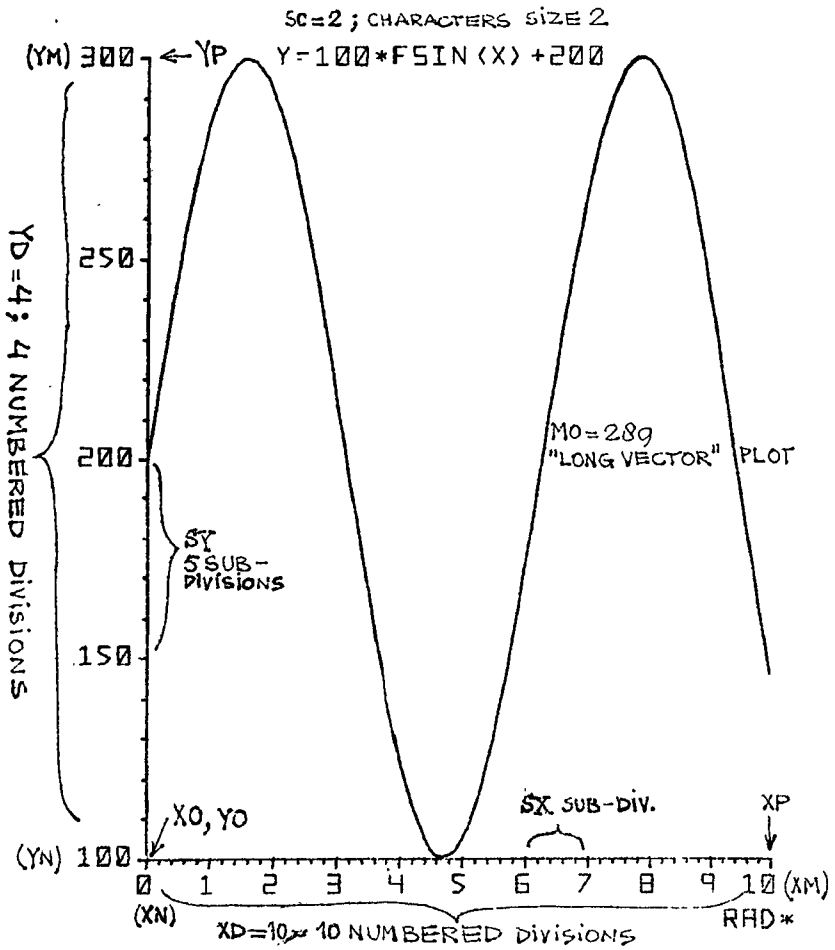
Example: Hatch the histograms of chapter 9e vertically. As the length of the vectors is undetermined, we use long vectors.

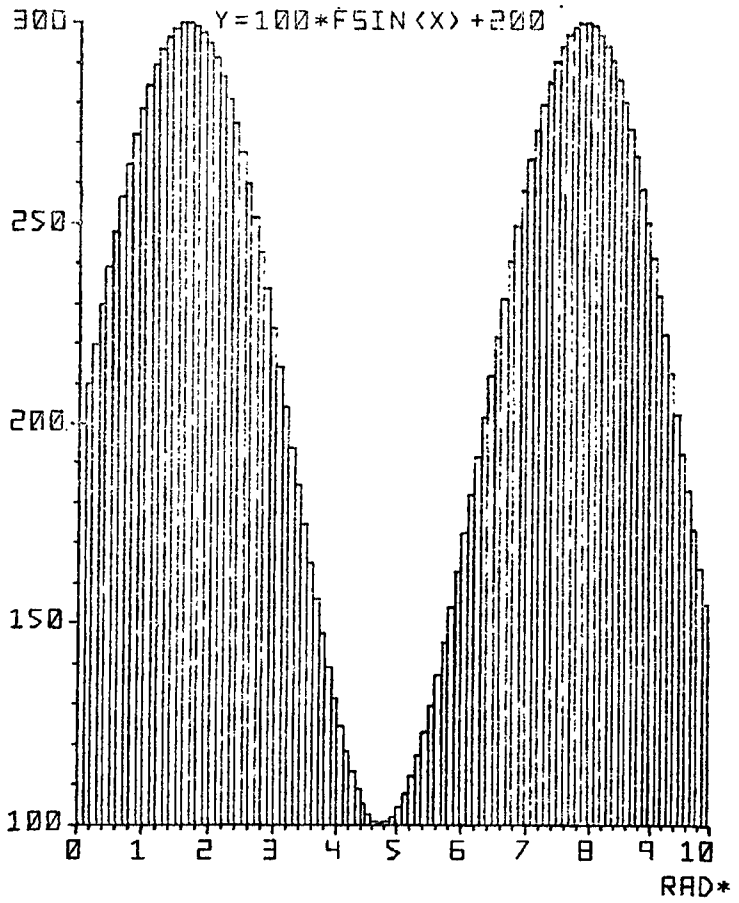
```
01.10 FOR I=1,6;FOR X=30*(I-1),3,30*I,SET Z=FX(,289,X,,X,A(I) )
```

10) Survey of the Possibilities for MODE

OCTAL	ELEMENTS	DECIMAL	DESCRIPTION
2	reset	2	Reset integrators and sample-and-holds
4	erase	4	Erase the screen
6	res.erase	6	2 and 4 together
46	res.erase	38	2 and 4 during 4 msec.
1422	displ.cursor	786	Display cursor
1426	cursor+erase+reset	790	All the above together
401	abs+short	257	Normal short, fast vector
441	abs+long	289	Normal long, slow vector
561	abs+long+dot+vc	369	Normal long dotted vector

OCTAL	ELEMENTS	DECIMAL	DESCRIPTION
451	abs+long+circ	297	Quarter of a circle if ARC greater than 90
411	abs+short+circ	265	1/64 part of a circle
471	abs+long+vc+circ	313	Normal timed circle. ARC in degrees
431	abs+short+vc+circ	281	idem
1	rel+short	1	Normal relative short vector
41	rel+long	33	Normal relative long vector
161	rel+long+dot+vc	113	Relative dotted vector
1001	point	513	thin point


SC=2 ; CHARACTERS SIZE 2
Y=100*FSIN(X)+200
(YM) 300
← YP
250
200
150
(YN) 100
YD=4 ; 4 NUMBERED DIVISIONS
5Y
5 SUB-DIVISIONS
MO=289
"LONG VECTOR" PLOT
XO, YO
SX SUB-DIV.
XP
0 1 2 3 4 5 6 7 8 9 10 (XM)
(XN)
XD=10 10 NUMBERED DIVISIONS
RAD*


Y=100*FSIN(X)+200
300
250
200
150
100
0 1 2 3 4 5 6 7 8 9 10
RAD*

ADDENDUM

A NEW GRAPHICS OVERLAY FOR 8K FOCAL

J. F. Anthoni

Medical Biological Laboratory of the National Defense Research Organisation TNO,

Rijswijk Z. H., The Netherlands

Summary

In order to expand the display capabilities of 8K FOCAL a program has been written to combine the possibilities of the existing display - overlay GRAPH and the character generator program VSCG (Murray Ruben) so that vectors, arcs and characters can be displayed. In addition provisions were made for easy keyboard-to-scope interaction. Two function names were used: FX for vector and arc plotting, and FDIS to switch FOCAL's output to the character generator program.

Introduction

The presentation of results of biological experiments often occurs as graphics to be published in international scientific papers. Such graphics may be produced by a computer memory display of which photographs may be taken as the final step in the processing of experimental data. It is therefore necessary to be able to display characters in several sizes and to draw vectors and arcs. These and other possibilities (to enhance man-machine interaction by the use of a computer display) are implemented in the program presented in this paper.

Character Display

In the character generator program VSCG the tabulation routine is replaced by a scaling routine which multiplies all vector increments with a scale factor. The function FDIS (SCALE, XØ, YØ) transfers a scale factor and an origin (XØ, YØ) to VSCG and switches FOCAL's output from teletype to VSCG.

Example: SET Z=FDIS(2,5Ø, 1ØØ); TYPE"ABCD" will produce the word ABCD written in character size 2 from position (5Ø,1ØØ) to the right of the screen. SET Z=FDIS( ) switches FOCAL's output back to the teletype.

Keyboard-to-scope Editing

Sometimes a user wants to insert text strings that have not been programmed into the generated picture before making a photograph. The function FDIS(-SCALE) causes all characters from the keyboard to be trapped and routed to VSCG. The following characters control position and size of the text to be inserted:

CTRL/O: read the coordinates of the cursor(joystick) and take these as the origin for the next characters to be typed.

CTRL/Q: increase character size (scale factor +1)

CTRL/R: reset character size to the smallest size (scale factor = 1). Being isolated from keyboard input, FOCAL is still alert for CTRL/C (normal keyboard interrupt). It proceeds with the program when CTRL/P is hit, restoring input and output to normal.

Vector Plotting

The function FX(ARC, MODE, XØ, YØ, X1, Y1, ...XN, YN) plots vectors and arcs head-to-tail from (XØ, YØ) to (X1, Y1) consecutively until (XN, YN). In fact an invisible vector is drawn (XØ, YØ) and visible vectors from there on. To establish exact linking to the vector an invisible point is plotted at the end of each vector by analog to digital conversion (software) of the X- and Y-INTEGRATOR voltages in the analog computer part of the KV8/I interface.

All possibilities embedded in this interface such as absolute, relative, long, short timed vectors, circles and arcs and also erase screen and reset integrators are possible by inserting the correct (decimal) code for MODE. The value of ARC determines the length of the vector by timing, thus also controls the length of an arc in degrees around the circumference.

Example: FOR X= -2ØØ, 5Ø, 2ØØ; SET Z= FX(,289,X,Ø,X+5Ø,Ø,X+5Ø,4) plots an X-axis in long vectors (289).

Example: SET Z = FX(18Ø, 313, -1ØØ, Ø, Ø, Ø, 5Ø, Ø, 25, Ø) plots a spiral of half circles (ARC = 18 Ø degr.; MODE = 313) from the startpoint (-1ØØ,Ø) clockwise head-to-tail around the centres (Ø, Ø) (5Ø, Ø) and (25, Ø).

Dotted Vectors

A dotted vector can be drawn by inserting the code 369 for MODE. Now the information in ARC is used as the dot-length; the intensity is turned on and off for each dot length until the end of the vector is reached (90 counts,=time-units = degrees). Example: SET Z = FX(3,369, -1ØØ,Ø, Ø, Ø) will draw a dotted vector of 15 visible and 15 invisible sections with a length of 3 time units each between (-1ØØ, Ø) and (Ø, Ø).

Joystick and Cursor

Each time a vector is displayed, the cursor will reappear, indicating the position "pointed to" by the joystick. The cursor can also be displayed by the command SET Z = FX(,786,,,,).

Read-out of the coordinates of the cursor position can be obtained by the command SET XC = FX(,X); SET YC = FX(,Y). After analog to digital conversion of the corresponding voltages in the KV8/I interface the function FX achieves the integer-value of the X- and Y-cursor position respectively. Note that X and Y are treated as characters and not as variables.

Interrupt-bar

Pressing the interrupt-bar effectuates a "DO 31" as in GRAPH. Between two program

statements FOCAL jumps to a routine which checks a software interrupt flag that can be set by the interrupt service routine. It then sets up locations within FOCAL to effectuate a "DO 31."

Keyboard Interaction

All incoming characters are trapped and tested for the control-characters CTRL/OPQRSTUV. These are decoded in the command decoder of VSCG and are never routed to FOCAL (no echo). By doing so these commands can be issued at any time and in any order.

CTRL/S :switch FOCAL output to SCOPE
CTRL/T :switch FOCAL output to TELETYPE
CTRL/U :erase the screen and start from above again
CTRL/Q :increase character size
CTRL/R :reset character size

Extended Functions, Core Allocation

Because this program is not written over existing functions, the functions FATN, FLOG, FEXP, FSIN, FCOS are all valid, leaving 128 variables for the user's programs. This display overlay occupies loc. 3140-3377 under the variables area and loc. 6532-7577 in FIELD 1. More variables (200) can be stored at the expense of the above-mentioned functions by issuing the command CTRL/V. Attempting to use one of the above-mentioned functions will now cause an error message.

Corrections to FOCAL

Besides correcting two bugs (loc. 7525 and 5333) the print-out of "=" and ":" is suppressed. A provision was found to do a correct carriage-return without linefeed ("#") on teletypes with FORM FEED option. For the same reasons the command CTRL/I (in MODIFY) had to be changed to CTRL/N (NEXT CHARACTER).

Configuration and Programs Needed

Minimum configuration: PDP8/I with 8K of core memory and KV8/I, VTØ1, H3Ø6 programs:

1) Load FOCAL 5/69

2) Answer initial dialogue with "yes"

3) Load 8K overlay

4) Load 4 word overlay if needed

5) Load this program; start 200

ADDENDUM #2

LAZY PLOT ROUTINES FOR 8K FOCAL DISPLAY

by
Floor Anthoni

Medisch Biologisch Laboratorium RVO-TNO, Lange Kleiweg 139, Rijswijk (ZH)

1. Purpose

In order to achieve quick plotting results, a set of routines was developed which enables the user to plot an X- and Y-axis with marks, subdivisions and legends and curves as histograms without worrying about scaling and character size. The routines are especially useful when more than one graph is needed; one can easily change the parameters for the next one.

Of course these routines can be compacted to serve in special purpose plots. This then minimizes the number of variables used. These routines were written for a 1-quadrant plot, but can be used for other quadrants (see section 4).

2. Used Variables

The 21 variables used are:

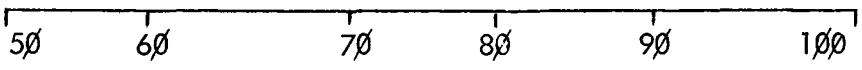
XO, XØ, X, XM, XN, XC, XD, SX
YO, YØ, Y, YM, YN, YC, YD, SY
MO, SC, I, Z, X1,

of these variables X, Y, I, Z, and X1 are used temporarily, and can be shared by other programs.

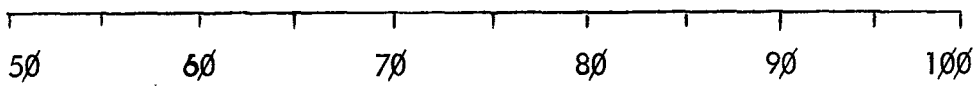
XO, YO, XP, YP are the coordinates of the origin and perimeter of the plot. They determine position and size of the rectangle which encloses the plotting area. These variables are defined in scope-coordinate values (YO, YP between -300 and +300 and XO, XP between -250 and +250). A graph can easily be shifted by changing the values of these variables.

XM, XN, YM, YN give the actual region of interest of the plot. They also determine the legends alon the X- and Y-axis, e.g. when XM = 100 (X MAXIMUM) and XN = 50 (X NULL), an X-axis will be plotted from the value 5Ø to the value 100, independent of its position and size (which are determined by XO and XP). Note: XM and XN, YM and YN may be interchanged.

XD, YD are used to divide the X- and Y-axis into numbered divisions. Each DIVISION has its own legend, it is therefore necessary that each division is exact. In the example above the value of XD would be 5 or 10 etc.



SX, SY enable the user to make SUBDIVISIONS if in the above example SX was set to 2, each division would be divided into two subdivisions. These subdivisions bear no legends. SY and SX may not be zero; if no subdivisions are needed, they should be set to 1.



XC, YC refer to the number of CHARACTERS FOCAL types in the type command. Legends will be shifted so that they match the marks on the X- and Y-axis. In the above example one wishes to type 3 digits, so the format used is % 3.00. When issuing the command TYPE % 3.00, 50 FOCAL responds with 4 characters: 50*. In this case XC should be set to 4.

One can at best try which format (and XC-, YC-value) fits best in his specific plot. YC can be chosen negative so that legends appear to the right of the Y-axis (see also section 4).

SC is the SCALE FACTOR for the characters. It enables the user to plot legends in several size characters (size 1, 2, 3, 4 etc.). The position of the characters with respect to the X- or Y-axis is shifted according to their size.

MO sets the vector MODE during curve- or histogram plot (see page 14 of FOCAL 8K DISPLAY MANUAL).

257 = short vector = thin plot (256 = invisible short vector)
289 = long vector = thick plot (288 = invisible long vector)
369 = dashed vector = dotted plot (368 = visible dashed vector)

3. How to Use the Display Routine

Group 29 : X-axis plot
Group 30 : Y-axis plot
Group 31 : Curve or histogram plot

The use of these routines can be illustrated by the following example in GROUP 1, which plots a sine-wave from X = 0 to X = 10 radians.

a) Locate maximum and minimum values for X and Y (XM, XN, YM, YN), and the number of divisions and subdivisions (XD, YD, SX, SY).

c) Determine from the format used, the number of typed characters: (XC, YC). A "." in a number also counts for a full character.

d) Determine the SCALE of the legends and the MODE of the vector plot (SC, MO).

e) Always set the format before issuing a DO 29 or DO 30. The X- and Y-axis will now be plotted.

f) Special attention should be paid to the first vector, which should be invisible. It is therefore necessary to compute X and Y of the beginning of the curve, to set MODE to invisible vector (preferably 288) and then DO 28. MODE should then be restored.

Note: 369 gives a dashed vector; 368 also, it only starts with an invisible segment.

4) Plotting in Other Quadrants

Principally XO, YO, XP and YP determine the orientation (and position and size) of the graph. A plot in any quadrant can be done by setting XO, YO, XP and YP to the appropriate values.

Difficulties arise when the legends have to be moved to the other side of the axis.

An Y-axis with legends appearing at the right of it, can be written when YC = ¯1. The marks will still be at the opposite side. This can only be changed by modifying 30.50 "XØ-3" into "XØ + 3" and 30.55 "XØ-6" into "XØ + 6."

For the X-axis: 29.50 "YØ-3" into YØ + 3" and 29.55 "YØ-6" into YØ + 6."

The only way to produce legends above the X-axis is by modifying 29.60 "Y0-12-7 * SC" into "Y0 + 12."

```
Ø1.Ø1  S     XO=-2ØØ;S XP=25Ø;S XM=1Ø;S XN=Ø;S XC=3;S XD=1Ø;S SX=5
Ø1.Ø2  S     YO=-25Ø;S YP=3ØØ;S YM=3ØØ;S YN=1ØØ;S YD=4;S SY=5;S YC=4
Ø1.Ø3  S     SC=2;S MO=289
Ø1.Ø4  T     %3.ØØ;D 3Ø;T "          Y=1ØØ*FSIN(X)+2ØØ";T %2.ØØ;D 29;T !,"RAD"
Ø1.19  S     X=Ø;S Y=1ØØ*FSIN(X)+2ØØ;S MO=288;D 28;S MO=289
Ø1.2Ø  F     I=Ø,.1,1Ø;S Y=1ØØ*FSIN(I)+2ØØ;S X=I;D 28
Ø1.21  QUIT

28.1Ø  C     LAZY PLOT ROUTINES FOR QUICK PLOTTING;SET VARIABLES FIRST
28.3Ø  S     X=XO+(X-XN) *( (XP-XO)/(XM-XN))
28.4Ø  S     Y=YO+(Y-YN) *( (YP-YO)/(YM-YN))
28.5Ø  S     Z=FX(6,MO,XØ,YØ,X,Y);S XØ=X;S YØ=Y;C FOR CURVES
28.51  C     FOR HISTOGRAMS:S Z=FX(6,MO,XØ,YO,XØ,YØ,X,YØ,X,YO);S XØ=X;S YØ=Y

29.3Ø  S     I=Ø;S X=XO;S XØ=XO;S YØ=YO;D 29.6Ø;D 29.55
29.45  S     I=I+1
29.5Ø  S     X=X+(XP-XO)/XD;F X1=XØ,(X-XØ)/SX,X;S Z=FX(,289,X1,YØ,X1,YØ-3)
29.55  S     Z=FX(,289,XØ,YØ,X,YØ,X,YØ-6)
29.6Ø  S     XØ=X;S Z=FDIS(SC,X-XC*SC*8,YO-12-7*SC);T XN+1*(XM-XN)/XD
29.8Ø  IF    (I-XD)29.45;R

3Ø.3Ø  S     I=Ø;S Y=YO;S YØ=YO;S XØ=XO;D 3Ø.6Ø;D 3Ø.55
3Ø.45  S     I=I+1
3Ø.5Ø  S     Y=Y+(YP-YO)/YD;F X1=YØ,(Y-YØ)/SY,Y;S Z=FX(,289,XØ,X1,XØ-3,X1)
3Ø.55  S     Z=FX(,289,XØ,YØ,XØ,Y,XØ-6,Y)
3Ø.6Ø  S     YØ=Y;S Z=FDIS(SC,XO-(YC+2)*SC*6-5,Y-SC*3);T YN+I*(YM-YN)/YD
3Ø.8Ø  IF    (I-YD)3Ø.45;R
*
```