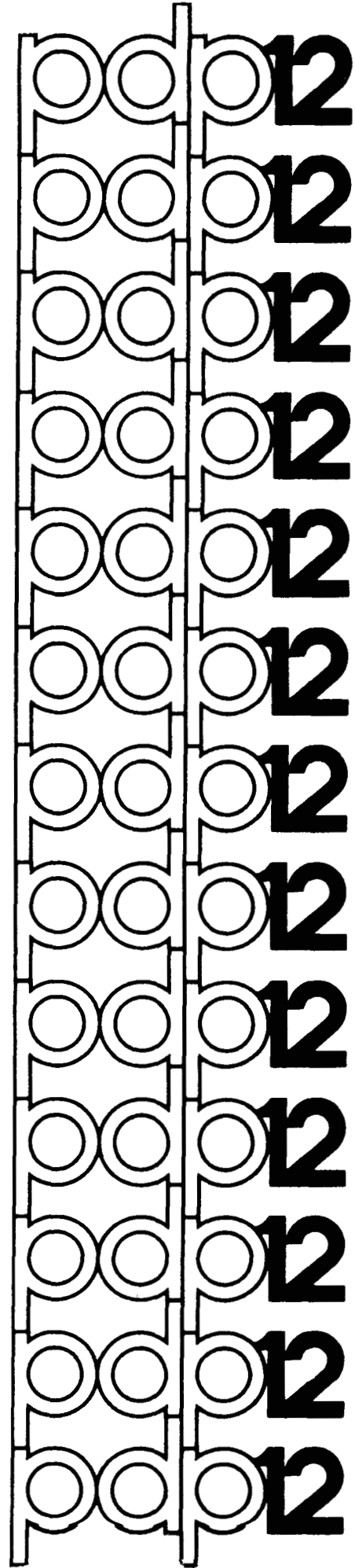


FOCAL-12



FOCAL-12

PROGRAMMING MANUAL

For additional copies, order No. DEC-12-AJAA-D from
Program Library, Digital Equipment Corporation,
Maynard, Massachusetts 01754 Price \$3.00

1st Edition December 1970

Copyright © 1970 by Digital Equipment Corporation

The material in this manual is for information purposes and is subject to change without notice.

The following are trademarks of Digital Equipment Corporation, Maynard, Massachusetts

DEC

FLIP CHIP

DIGITAL

PDP

FOCAL

COMPUTER LAB

TABLE OF CONTENTS

	<u>Page</u>
1.0 INTRODUCTION	1
1.1 FOCAL-12	1
1.2 Hardware Configuration	1
1.3 Required Reading	1
1.4 Loading FOCAL-12	1
2.0 USING THE DISPLAY SCOPE	2
2.1 FDIS	2
2.2 OUTPUT	2
2.2.1 OUTPUT SCOPE	2
2.2.2 OUTPUT TELETYPE	3
2.2.3 OUTPUT DELAY	3
2.2.4 OUTPUT CLEAR	4
2.2.5 OUTPUT ERASE	4
3.0 DATA STORAGE AND RETRIEVAL	4
3.1 LIBRARY MAKE	5
3.2 LIBRARY OPEN	6
3.3 LIBRARY CLOSE	7
4.0 FOCAL-12 PROGRAM STORAGE	8
4.1 LIBRARY SAVE	8
4.2 LIBRARY LOAD	9
4.3 LIBRARY GO	9
5.0 ANALOGUE INPUT AND TIMING	10
5.1 Sampling	10
5.2 Timing Using the OUTPUT INTERVAL Command	10
6.0 DATA ACQUISITION USING FOCAL-12	11
6.1 Memory	11
6.2 LINtape	12
6.3 Disk	12
6.4 \$THRUFØ	12
7.0 IMPLEMENTATION OF USER FUNCTIONS (FNEW,FX,FZ)	14
7.1 Summary	14
7.2 Example of User Implementation of FX	16
7.3 Special Requirements for FOCAL-12	18
7.4 FOCAL-12 Tag Table	19

	<u>Page</u>	
APPENDIX A	COMMAND AND OPERATION SUMMARY	
A.1	Commands	A-1
A.2	Functions	A-3
A.3	FOCAL Operations	A-4
A.3.1	Format	A-4
A.3.2	MODIFY Operations	A-4
A.3.3	The Trace Feature	A-4
A.3.4	Special Characters	A-5
A.4	FOCAL-8 Features Not in FOCAL-12	A-5
APPENDIX B	COMPLETE STARTUP PROCEDURE	B-1
APPENDIX C	DATA FORMAT SPECIFICATION	C-1
APPENDIX D	FOCAL-12 I/O AND THE L C COMMAND	D-1
APPENDIX E	MISCELLANEOUS COMMENTS	E-1
APPENDIX F	FOCAL-12 ERROR DIAGNOSTICS	F-1

FOCAL-12

1.0 INTRODUCTION

1.1 FOCAL-12

FOCAL-12 is an expansion of DEC's FOCAL^(R) language¹ designed to optimize use of the PDP-12 computer and its standard peripheral devices: LINC-tape, disks, VR12 display, A/D channels, KW12A real time clock, console switches and relays. Simple data acquisition and reduction tasks or the analysis of previously generated data may be easily and quickly programmed using FOCAL-12.

1.2 Hardware Configuration

The minimum system configuration for FOCAL-12 is a PDP-12B with 8K of core memory. Supported options include multiple LINCtape units, RSØ8 or RK8 disks, 32 A/D channels and KW12A clock.

1.3 Required Reading

Before continuing with this supplement, the reader should be familiar with the basic FOCAL commands and programming fundamentals, as described in the FOCAL-8 chapter of Programming Languages. Appendix A of this document presents the FOCAL-12 Command and Operation Summary (including FOCAL-8) and lists those features of FOCAL-8 not included in FOCAL-12.

1.4 Loading FOCAL-12

FOCAL-12 is loaded by the LAP6-DIAL-MS² system:

```
→LO FOCAL-12,n )
```

where n is the unit containing FOCAL-12. FOCAL-12 prints an asterisk (*) on the Teletype^(R) and can then be used just as FOCAL-8 (with added features). FOCAL-12 expects a system tape on unit Ø (or unit 1Ø for disk) so that it can read in the system I/O routines when necessary.

¹Specifically, DEC-08-AJAE-PB, FOCAL 1969, Version ZZM.

²Some familiarity with DEC-12-SE2D-D, the LAP6-DIAL Manual, is assumed; however, the complete startup procedure is detailed in Appendix B. Hereafter, in this document, LAP6-DIAL is referred to as DIAL.

2.0 USING THE DISPLAY SCOPE

2.1 FDIS

The function FDIS is used for plotting points on the VR12 display scope. Its general format is

```
SET H=FDIS(X,Y)
where 0<X<1.39 and 0<Y<1.00.
```

FDIS is most commonly used in conjunction with the FOR statement. For example, the program

```
F I=0,.01,1.39;S H=FDIS(I,.5)
```

would plot a line of points across the middle of the display. FOCAL-12 can display about 950 points on the scope at a time.

2.2 OUTPUT

To further facilitate display usage, a command has been added to FOCAL-12. Its forms are:

<u>Full Command</u>	<u>Abbreviation</u>
OUTPUT SCOPE	O S
OUTPUT TELETYPE	O T
OUTPUT DELAY	O D
OUTPUT CLEAR	O C
OUTPUT ERASE	O E

2.2.1 OUTPUT SCOPE

The O S command instructs FOCAL-12 to divert all its "Teletype" output to the display scope, including the echoing of Teletype input, output from "TYPE" and "WRITE" commands, error messages, etc.

When the scope is filled (32 lines or about 500 characters), computation is suspended and the display is maintained until the user types any key on the Teletype.

1. If a line feed is typed, FOCAL-12 clears the display and continues computation, otherwise ignoring the character. The program

```
1.01 O S
1.02 F I=1,35;T I,!
GO
```


generates the display

```
1.0000  
2.0000  
.  
.  
.  
32.0000
```

Pressing line feed will clear the display and continue the computation until the scope again fills or, as in this case, the computation is completed.

```
33.0000  
34.0000  
35.0000
```

2. If any character other than linefeed is typed, the scope is cleared, the character is accepted as legitimate input, and computation continues. Thus, if the screen is filled while editing, the user need not make any special effort to clear it and continue.

2.2.2 OUTPUT TELETYPE

The O T command merely negates the O S command; thus, if typed output is being sent to the scope, an O T command will cause subsequent typed output to be echoed on the Teletype.

2.2.3 OUTPUT DELAY

None of the commands presented thus far directly output to the display scope. Instead, the FDIS function builds up a buffer of display coordinates and the O S command causes "typed" output to be sorted into another buffer. Whenever FOCAL-12 is not computing (e.g., waiting for Teletype input) it refreshes the display using the information from the buffers. Conversely, while FOCAL-12 is computing, the display is not refreshed and is blank. In the example from section 2.1

```
F I=0,.01,1.39;S H=FDIS(I,.5)
```

the screen would be blank until the completed buffer was ready; then the line would appear on the scope.

The O D command interrupts computation to refresh the display. Thus, if the example from section 2.1 is extended to read

```
F I=0,.01,1.39;S H=FDIS(I,.5);O D
```

the display will be refreshed as each point is added to the buffer so that the user will see the line being plotted across the scope (at the cost of a slight delay in computation).

2.2.4 OUTPUT CLEAR

The display scope is cleared of all points and characters whenever the O C command is executed. One very convenient use of this feature is during program preparation. If editing is done while an O S command is in effect, the program on the display will contain rubouts, MODIFY lines, inserted lines, etc. The O C command will clear the scope and a WRITE command can be given to display immediately the edited program with line numbers in order, MODIFY statements erased, etc.

2.2.5 OUTPUT ERASE

The O E command clears the scope of all "typed" output, leaving any FDIS plots.

In summary, then, the program

```
1.01 F I=0,.04,6.25; S A=FDIS(.5+.3*FSIN(I),.5+.3*FCOS(I))
1.02 O S; T "          A CIRCLE          ";O T
1.03 F I=0,.01,1.39;S H=FDIS(I,0);O D
1.04 T "FAST TOO",!
GO
```

will

- (1.01) compute sines and cosines for a short time;
- (1.02) display a circle on the scope with a label over it;
- (1.03) plot a line of points across the bottom of the scope, (a point at a time because of the "O D");
- (1.04) then type "FAST TOO" (on the Teletype due to the "O T" in 1.02).

3.0 DATA STORAGE AND RETRIEVAL

As part of the data collection and reduction task, the typical user will collect data and store it on LINCtape or disk. FOCAL-12 can access such data either as named binary files¹ under the DIAL system,

¹To those familiar with DIAL formats, there is no "header" block.

or by absolute block number addressing, where the location of the data is the user's responsibility. The data set may be in any of three data formats -- signed integers, signed fractions, or floating point numbers. Once the file has been "opened" (refer to section 3.2), any element of the data array may be addressed as a standard subscripted variable and the actual tape or disk operations necessary to access the element will be automatically carried out by FOCAL-12.

3.1 LIBRARY MAKE

When a set of data is to be saved in a named file, the file must already be defined in the DIAL filing system. If such a file does not already exist, it may be created by using the LIBRARY MAKE command

```
L M,length,name,unit
```

where length is the number of blocks required to hold the data¹; name is the name to be assigned to the file for the DIAL index; and unit is the appropriate device unit number, as defined in DIAL-MS.

<u>Device</u>	<u>Device Unit Numbers</u>
8 LINctapes	Ø - 7
4 RSØ8 disks	1Ø - 17
1st RK8 disk	1Ø - 15
2nd RK8 disk	2Ø - 25
3rd RK8 disk	3Ø - 35
4th RK8 disk	4Ø - 45

Thus

```
L M,19,DATA,Ø
```

would create a 19 block file on tape Ø, naming it DATA.

Note that the number of blocks can be specified by a variable.

```
1.Ø1 ASK "HOW MANY BLOCKS FOR DATA?" N
1.Ø2 L M,N,DATA,Ø
```

This program will create a file, DATA, of N blocks where N is specified at run time.

FOCAL-12 will not put two files of the same name on the same tape or disk. If the file name is already used, it will be deleted and the new file created. However, before deleting the existing file, FOCAL-12 will display "REPLACE?" on the scope. To complete the operation, type "R"; striking any other key will cause FOCAL-12 to ignore the L M command.

¹Each block in the file will contain 256 integers, 128 fractions, or 85 floating point numbers (the last word of each block is unused with the floating point format).

If a legal, but nonexistent device (i.e., in the above list but not physically present) is specified, "NO" is displayed on the scope. There is only one option available currently - type carriage return to return to DIAL. Any other character is ignored.

This is actually a convenient procedure for returning to DIAL. However, if the wrong key has been struck by accident and a useful indirect program is in memory, the following procedure will recover it:

1. Stop the computer.
2. Set LSW to 0200 and the MODE switch to 8.
3. Press I/O PRESET and START LS.
4. If output to the scope was in effect at the time the computer was stopped, it will now be displayed in full size characters. Type "O S" to correct to FOCAL-12's normal display.

3.2 LIBRARY OPEN

FOCAL-12 will at any one time process up to eight files of data stored on LINCTape or disk. The files are referenced using the variable names "F0", "F1", ..., "F7". The standard subscripting methods apply: e.g., S F0(7)=0 will clear the eighth entry in file number 0. F I=0,1000; T F7(I),! will type out the contents of the first 1001 entries in file number 7.

It is important to realize that these file number tags do not, of themselves, specify a particular variable on tape or disk: this association between a file number and a specific file name or location is supplied by means of the LIBRARY OPEN command.

L O,file number,format,name,unit

where file number is "F0", "F1", ..., or "F7";

format is "F" for Floating point format
"S" for Signed fraction
"I" for signed Integers

name is the DIAL file name, or the starting block number written as # number, where number is any legitimate (i.e., existing) starting block number (octal);
unit is the device unit number.

The function of the OPEN command is then to associate a file number, Fn, with a data array on tape or disk and to define the type of data. For example,

```
L O,F1,F,DATA,1
```

declares an array of floating point numbers, in a file named DATA, on tape unit 1, is to be referred to as file number "F1". Any piece of data in the array may now be accessed using the standard FOCAL subscripting procedure. The program

```
1.01 L M,1,DATA,0
1.02 L O,F2,I,DATA,0
1.03 F I=0,255;S F2(I)=0
GO
```

creates a one block file named DATA on tape 0 and clears the array of unsigned integers to zeros. This example

```
1.01 L O,F2,I,#100,0
1.02 F I=0,511;S F2(I)=0
GO
```

sets blocks 100_8 and 101_8 of tape unit 0 to all zeros.

3.3 LIBRARY CLOSE

OPENed data files are CLOSED with the LIBRARY CLOSE command

```
L C,filename
```

This command causes FOCAL-12 to clean up any "unfinished" (see Appendix D for a more complete description) transfers for that file and to free the file number. All OPENed files should normally be CLOSED at the end of a program to prevent succeeding programs from accidentally destroying the file.

For example,

```
1.01 L M,19,COPY,1
1.02 L O,F1,F,COPY,1
1.03 L O,F2,F,ORIG,0
1.04 F I=1,1200;S F1(I)=F2(I)
1.05 L C,F1;L C,F2
1.06 Q
GO
```

will create a file on unit 1 that is 19 blocks long called COPY, copy 12000 floating point numbers from file ORIG of LINCtape 0 to LINCtape 1 file COPY, and QUIT after closing both files.

To reassign a file number to another file requires only another L O command; e.g.,

```
1.01 L O,F2,I,FILEA,0
      .
      .
      .
```

All references to "F2" are executed as FILEA references by FOCAL-12.

```
      .
      .
3.21 L O,F2,F,FILEB,10
      .
      .
      .
```

FILEA is CLOSED and all references to "F2" are now executed as FILEB references on disk unit 0.

Thus, a second L O command to a particular file number will negate or "CLOSE" the first L O command.

4.0 FOCAL-12 PROGRAM STORAGE

4.1 LIBRARY SAVE

FOCAL-12 programs may be saved on LINCtape or disk for later use. The LIBRARY SAVE command is used to store the current (just edited) FOCAL-12 program. Its format is

```
L S,name,unit
```

where name is the binary file¹ name to be inserted in the DIAL index and unit is device unit number. For example,

```
L S,$NEWPRGM,7
```

will save the program (text and variables) just typed as a binary file named \$NEWPRGM on tape unit 7. Since the program is saved as a binary DIAL program, and listed as such in the DIAL index, it is suggested that FOCAL-12 programs be filed under some standard notation, such as

¹Again, no "header" block.

dollar sign for the first character. Note that a copy of the program just saved remains in core after an L S command and may be executed using the GO command. Finally, if a file with this name already exists on the device, the "REPLACE?" message is displayed: type "R" to complete the operation; type anything else to cancel the command.

4.2 LIBRARY LOAD

A FOCAL-12 program that has been saved by a LIBRARY SAVE command can be retrieved from the tape or disk by a LIBRARY LOAD command in the format

```
L L,name,unit
```

where name and unit are as previously described in section 4.1. For example,

```
L L,$NEWPRGM,Ø
```

will retrieve the program saved in the example in section 4.1 (assuming the tape had been put on tape transport Ø). Once loading is complete, FOCAL-12 prints an asterisk to indicate editing may continue or the program may be started.

4.3 LIBRARY GO

A FOCAL-12 program that has been saved by a LIBRARY SAVE command can be retrieved from a DIAL binary file on tape or disk and started automatically by a LIBRARY (load and) GO command in the format

```
L G,name,unit
```

where name and unit are as specified for LIBRARY LOAD. For example,

```
L G,$NEWPRGM,Ø
```

will not only load the program, \$NEWPRGM, into memory but will start it automatically.

Note that this feature can be used in conjunction with FOCAL-12's data file handling to enable the operation of large programs by "segmenting" or "chaining". For example, one segment could set up an experiment, acquire data, store it into a file, and load and start a second segment using the LIBRARY GO command.

```

1.05  L M,10,DATA,0
1.10  L O,F0,I,DATA,0
      .
      .
2.75  L G,$2NDSGMT,0

```

The second segment could then process the data, put up a display of results, etc., call a third segment,...

The processing of the L G command requires the use of a portion of the display buffer; however, up to approximately 450 points may be retained. If the display is to be cleared, the second segment should merely start with an "O C" command.

Note that files opened in one segment need not be opened in succeeding segments. Of course, the final segment should CLOSE any files still open.

5.0 ANALOGUE INPUT AND TIMING

5.1 Sampling

FOCAL-12 can sample from 1 to 32 analog channels for digital conversion via the FADC command

```
FADC(n)
```

where n is the channel to be sampled. The voltage across the AD-12 channel at the moment of sampling is the value recorded.¹ Readings collected by this function can be placed in a data file for later analysis or can be processed immediately. For example,

```

1.01  L O,F1,S,DATA,10
1.02  F I=0,1000;S F1(I)=FADC(3)
1.03  F I=0,1000;T F1(I),!
1.04  L C,F1
1.05  Q

```

will take 101 samples from A/D channel 3, store the result in the previously "made" file DATA on the first disk unit, type out the values and QUIT.

5.2 Timing Using the OUTPUT INTERVAL Command

The KW12A clock can be used by FOCAL-12 for user specified interval timing, permitting a delay of known duration to elapse between events.

¹Values from FADC range from -1.000 to .998 with a resolution of 2/1024V.

This interval is established by the O I command

```
O I,n
```

where n, which may be an expression, is the length of the interval in seconds, with $.01 \leq n \leq 40.95$. Thus, an interval of two seconds is specified by

```
O I,2
```

In this case, FOCAL-12 starts the clock so that it will "tic" at two second intervals. If an O I command is now issued in the program without an argument, FOCAL-12 will delay the program until the next tic (up to 2 seconds) occurs, thereby synchronizing the program with the real time clock.

The following program averages samples, taken once per second from A/D channel 1, for 20 seconds.

```
1.01 O I,1;S A=0
1.02 F I=1,20;S A=A+FADC(1);O I
1.03 T A/20,!
1.04 Q
```

6.0 DATA ACQUISITION USING FOCAL-12

FOCAL-12 is not intended to be a replacement for the existing and/or planned data acquisition programs for the PDP-12. However, where low data rates are used, FOCAL-12 will facilitate the data acquisition task. Several methods are discussed in this section (all assume storing of the data into an "S" file of signed fractions).

1. Storing in memory (≤ 50 readings/sec)
2. Storing on LINCtape (≤ 1.2 or $\geq .7$ secs/reading)
3. Storing on disk (≤ 20 readings/sec on RF08; ≤ 12 readings/sec on RK8)
4. Using \$THRUF0 (single channel, ≤ 100 readings/sec to LINCtape)

6.1 Memory

Up to 256 readings may be acquired at a maximum rate of 50 per second using the OUTPUT INTERVAL command and the FADC function. The 257th reading will exceed the capacity of the memory buffer and require output to the auxiliary storage file.

6.2 LINCtape

The 257th reading and every 128th reading thereafter requires a pause of 1.2 seconds to write and check the block on LINCtape. During this pause, there is no sampling so readings are lost. Therefore, the maximum sustainable rate is 1.2 seconds per reading.

The following procedure eliminates the checking feature on the tape write and reduces the pause time to .7 seconds. Note that this is a change to FOCAL-12 itself; i.e., all transfers to tape are made without error checking. Having started DIAL,

```
→ZE )
→AB FOCAL-12,unit )
→AS PATCH,unit )
→SB FOCAL-12,unit )
```

where "PATCH" is

```
PMODE
*3632
5272
*3672
6211
1302
3703
6201
6212
4667
3651
5235
5243
7640
LISTAP -7
```

6.3 Disk

Again, the worst case is assumed as the limit. For the RS08, maximum access time limits maximum throughput to about 20 readings/second. For the RK8, the possibility of crossing a disk track boundary puts the maximum throughput rate at about 12 readings/second.

6.4 \$THRUF0

This version of the user function, FX, (refer to section 7.0) has been implemented to facilitate single channel throughput to LINCtape at up to 100 readings/second. The overlay uses about half of the user space and is called (assuming FOCAL-12 has been loaded) by the command

```
L L,$THRUF0,unit
```

FØ must be opened as a signed fraction file.

```
L O,FØ,S,name,unit
```

or

```
L O,FØ,S,#dddd,unit
```

The function is then initiated by the command

```
S H=FX(channel, no. of samples, sync, rate)
```

where

channel specifies the analogue channel (\emptyset -31₁₀) from which samples are to be taken;
number of samples is limited by file size;
sync specifies the device to initiate the data collection:
 \emptyset -5 for sense switches \emptyset -5
 1 \emptyset -25₁₀ for sense lines \emptyset -17₈;
rate specifies the time between samples in seconds (. \emptyset 1-4 \emptyset .95).

The following sequence

```
L L,$THRUFØ,Ø  
1.Ø1 L M,1Ø,DATA,1  
1.Ø2 L O,FØ,S,DATA,1  
1.Ø3 S A=FX(7,12ØØ,15,.Ø1)  
1.Ø4 L L,$WORK,Ø  
L S,$GETDATA,Ø
```

would save a FOCAL-12 program, which when operated would

```
(1.Ø1) create a file named DATA on tape unit 1  
(1.Ø2) OPEN the file as FØ  
(1.Ø3) after the sync pulse from sense line 5, take 12ØØ  
samples, from analogue channel 7 at a rate of 1ØØ  
samples/second and store them in file DATA  
(1.Ø4) call a processing routine, $WORK.
```

There are several possible error messages from \$THRUFØ:

```
?17.31 FØ was not opened as an "S" file  
?17.53 The file size is insufficient for the number of samples  
?18.Ø3 Format error in specifying parameters  
?18.Ø5 Illegal sync device
```

7.0 IMPLEMENTATION OF USER FUNCTIONS (FNEW,FX,FZ)

FOCAL-12 provides the traditional FOCAL capability for encoding user functions to satisfy those users with special requirements not covered by FOCAL-12. The FNTABF table has three entries - PFNEW, PFX, and PFZ - for this purpose. For internal specs, FOCAL-12 is equivalent to FOCAL-8 and information obtained from DEC-08-AJAE-PB is valid except for addresses, which are presented in the tag table in section 7.4.¹ For convenience, a summary is presented.

7.1 SUMMARY

Text Handling Subroutines

GETC = Get next character from the text; store into CHAR.

SORTC = Sort AC (if non-zero) or CHAR against LIST.

Calling sequence: SORTC /call
LIST-1 /address of LIST-1
XXX /return if in LIST
XXX /return if not in LIST

NOTE: Lists are terminated by negative numbers.

PRINTC = Print the AC; if the AC= \emptyset , print the contents of CHAR.

READC = Read and echo a character from the keyboard and put it into CHAR.

SPNOR = Ignore spaces in text; exit with the first character that is not a space in CHAR.

ERROR = Transfer control to the command mode and terminate execution; print error message.

TESTN = This subroutine is actually a series of SORTC's with various returns:

CALL: TESTN /call
return1 /return if a period
return2 /return if not a period or a number
return3 /return if a number; SORTCN is set
/to the binary equivalent.

The routine tests only CHAR. AC must be \emptyset .

TESTC = This subroutine is again a series of SORTC's with various returns:

CALL: TESTC /call
return1 /terminator; SORTCN set according
/to TERMS
return2 /number; SORTCN sets as in TESTN
return3 /function; (CHAR=F)
return4 /alphabetic character

¹Another excellent source of information is DECUS No. FOCAL-17; indeed, much of the information in this section is taken from that document.

SORTJ = This subroutine is used as a multiple sort and branch subroutine. CHAR (or the AC if nonzero) is compared to a list. If it is in the list, an address is looked up and an effective JMP ADDRESS is executed. If a match is not in the list, then return is to CALL+3.

```
CALL:  SORTJ
      LIST1-1      /ADDRESS of character list
      LIST2-LIST1 /difference in the addresses
              /of lists
      RETURN      /return here if not in LIST1
```

PUSHA = Put the contents of the AC on the PDL; clear the AC.

POPA = Get the top entry on the PDL and put it in the AC.

PUSHF = This is essentially three PUSHA's and is used for storage of floating point data.

```
CALL:  PUSHF
      ADDRESS      /address of first location of
              /three word floating point number
```

POPF = The inverse of the PUSHF routine.

```
CALL:  POPF
      ADDRESS      /address of where to put data
```

PUSHJ = This is the recursive subroutine call. The subroutine return is put on the PDL and a JMP to the subroutine address is executed.

```
CALL:  PUSHJ
      SUBROUTINE  /address of SUBROUTINE
      XXX         /address of this location is stored
              /on the PDL
```

POPJ = Recursive subroutine return; the top element of the PDL is used as the effective address of the return.

Other Subroutines

INTEGER = Enter via a JMS I INTEGER. This routine makes an integer out of the floating accumulator (FLAC). The low order part is in FLAC+2 and in the accumulator; the high order part is in FLAC+1.

EFUN3I = This routine is the return from a function routine. It checks for a right parenthesis ()) in CHAR and normalizes the FLAC. Enter via a JMP I EFUN3I with the function result, if any, in FLAC.

EVAL = This subroutine evaluates an arithmetic expression. Because it is recursive, it must be called via:

```
PUSHJ
EVAL
XXX          /return
```

The subroutine return is to CALL+2 with the floating point value of the expression it evaluated in the FLAC.

Links to FOCAL

The general form of a function is FUNC(ARG1,ARG2,---). The function coding is entered via a SORTJ where the address is designated in the table:

FNTABF=.	/(374) in FOCAL-12
XABS	/address of FABS coding
XSGN	/FSGN
XINT	/etc.
XDISP	
XRAN	
XADC	
ARTN	
FEXP	
FLOG	
FSIN	
FLOS	
XSQRT	
PFNEW,	ERROR5
PFX,	ERROR5
PFZ,	ERROR5

To add a user coded function, put the entry point of the function coding in the appropriate location in the above table. FOCAL-12 will then branch to that location after the function name is decoded, and ARG1 is evaluated in the FLAC. To delete a function from the list, replace the current contents with 2725.

When the function evaluation is complete, the answer must be left in the FLAC, and a JMP I EFUN3I executed. The EFUN3I routine will check to see if there is a right parenthesis in CHAR, and normalize the FLAC, before returning to the appropriate place in FOCAL-12.

In general, user functions will use part of the text-variables storage area by changing location BOTTOM which contains the address of the last location to be used for storage - initially 4617.

If BOTTOM is made to contain 4277, for example, then the user has from 4300 through 4617 for storage of the function processor. The theoretical limit for BOTTOM is 3216, however, this would not allow any user space for indirect statements.

Note that if LINC mode coding is used, interrupts must be off.

7.2 Example of User Implementation of FX

```

0000          *20
0001          /OVERLAY FOR FOCAL-12
0002          /MAKES "FX" RETURN THE RIGHT SWITCHES
0003          /
0004          /          DIAL COMMANDS FOR USING
0005          /          -----
0006          /
0007          /          ZE
0010          /          AB FOCAL12,U
0011          /          AS OVERLAY,U
0012          /          SB FOCLTFMP,U,P
0013          /          LO FOCLTFMP,U
0014          /          ERASE ALL
0015          /          LIBR SAVE,$NULLPRG,U
0016          /
0017          /          NOW YOU HAVE A NULL PROGRAM
0020          /          SAVED AS "$NULLPRG" WHICH WHEN
0021          /          LOADED BY FOCAL-12 BRINGS IN THE
0022          /          OVERLAY OF "FX"=RIGHT SWITCHES
0023          /          EG-IF SWITCHES STILL 7310(OCTAL)
0024          /          S A=FX(0);T A TYPES 3784.0000(DEC)
0025          PMODE
0026          ROTMOM=05
0027          FEXP=4620
0030          FLAC=44
0031          EFUN3I=136
0032          PFZ=412
0033          PFX=PFZ-1
0034          PFNEW=PFZ-2
0035          /
0036          /
0037          /
0040          *ROTTOM
0041          0035 4607          FX-1
0042          *PFX
0043          0411 4610          FX
0044          *FEXP-10
0045          4610 7604          FX, LAS
0046          4611 3046          DCA      FLAC+2
0047          4612 3045          DCA      FLAC+1
0050          4613 1216          TAD      027
0051          4614 3044          DCA      FLAC
0052          4615 5536          JMP I   EFUN3I
0053          4616 0027          027,   27
0054          /
0055          LISTAP -7

```

NO ERRORS

```

ROTTOM 0035
EFUN3I 0136
FEXP   4620
FLAC   0044
FX     4610
027    4616
PFNEW  0410
PFX    0411
PFZ    0412

```

7.3 Special Requirements for FOCAL-12

The FOCAL-12 facility of saving (L S) and/or loading (L L or L G) programs adds another dimension to the problem of processing user function overlays in the storage area; e.g., on a given DIAL tape, program A might use overlay FX; programs B and C, no overlay; program D, FZ; etc. The solution chosen is to have any necessary overlay in memory when the program is typed in and saved with the L S command. FOCAL-12 will save the program as well as the overlay and will load both when the program is requested via the L L or L G commands.

The remainder of this section explains how this is done with the above example and assumes a knowledge of DIAL on the part of the reader.

Having started DIAL-MS, type

1. →ZE ↵
Clears the binary working area.
2. →AB FOCAL-12,unit ↵
Adds FOCAL-12 to the binary working area.
3. →AS OVERLAY,unit ↵
Assembles the source for OVERLAY on unit, adding the binary to the FOCAL-12 binary in the binary working area.

N.B. The LISTAP-7 instruction must be included in the source for the overlay in order to have the assembler add to the binary working area, which already contains FOCAL-12.
4. →SB FOCLTEMP,unit,P ↵
Saves a "temporary" version of FOCAL-12, which contains the user function, FX.
5. →LO FOCLTEMP,unit ↵
Loads and starts FOCAL-12 with the overlay for FX.
6. FOCAL-12 commands
*E A
FOCAL-12 clears storage of all but the overlay.
7. *L S,\$NULLPRG,unit ↵
You now have a null FOCAL-12 program saved on unit and named \$NULPROG¹. When loaded by FOCAL-12, it brings in the overlay for the FX function. The program requiring this function may now be typed in and saved via the L S command. Whenever it is recalled

¹The temporary program FOCLTEMP is no longer needed and may be deleted from unit using the →DX,unit ↵ function in DIAL.

via the L L of L G commands, the new function will also be loaded. For example, if the switches were still set to 7310₈

```
1.01 IF(FX(0))2.1,3.1,4.1
.
.
.
.
GO
```

would transfer to line 2.1.

On the other hand, programs not needing this function can be typed in without first calling \$NULLPRG via the L L command and so would have the complete storage space available.

7.4 FOCAL-12 Tag Table.

A	0045	C140	2554	DATUM	7102
ABSOL	6751	C144	6140	DATUMA	7252
ABSOL2	6153	C200	0123	DCONP	6303
ABSOL3	7375	C260	0113	DCONT	0471
ABSOLV	5571	C3	5345	DCOUNT	6143
AC1H	0041	C5	5341	DDTJR	0004
AC1L	0042	C7	5335	DEBGSW	0026
ACMINS	6605	C9	5331	DECON	5627
ADDR	0040	CALLIN	7576	DECONV	5600
ADONE	6673	CCR	0077	DECP	5533
AF	4677	CDF	7000	DECR	5521
ALF1	4760	CEX1	6506	DEJUMP	1306
ALF2	4763	CEXP	6505	DELETE	4565
ALFZ	4755	CF	4705	DF	4710
ALGN	6572	CFRS	0133	DGRP	0425
ALIGN	6623	CFRSX	0137	DGRP1	0441
ALIST	1370	CGET	1133	DIG	5543
ALPHA	0016	CGETRE	1137	DIGIT	5713
AMOUNT	6722	CGETX	2564	DIGITS	0006
ARCALG	4732	CHAR	0066	DIV1	5754
ARCRTN	5024	CHARTA	0200	DIV2	6757
ARGNXT	1723	CHECK	1751	DIVDIV	1632
ARTN	5000	CHFLAG	0147	DIVIDE	7150
ASHFT	6665	CHIN	2157	DIVLUP	1635
ASK	1200	CHRCNT	0006	DMDONE	7063
ATLIST	1570	CHREND	0056	DMPSW	0100
ATSW	0056	CHRLUP	0033	DMULT	7004
AXIN	0010	CHRT	6133	DMULT4	7036
AXOUT	0017	CLCU	7427	DNORM	7335
B	0046	CLEAR	7672	DNUMBR	5714
B1BLK	2130	CLF	0076	DO	0420
B1FLG	2125	CLKFLG	2661	DOK	2113
B1UNIT	2126	CNTR	0057	DONE	2131
B2BLK	2134	COL	1253	DOONE	0463
B2FLG	2131	COMBOT	0226	DOUBLE	0127
B2UNIT	2132	COMBUF	0132	DPCVPT	6302
BACK	5503	COMEIN	3140	DPN	6305
BEGIN	3601	COMEQU	3206	DPT	6145
BET1	4771	COMGO	1161	DSAVE	5640
BET2	4774	COMLST	0774	DTST	5647
BETA	0017	COMMEN	0614	DUBDIV	7261
BETZ	4766	COMMON	1600	DUBLAD	5733
BF	4702	COMSUB	1502	DV3	7267
BLK2	0011	CON1	5037	E	0042
BLOCK	2121	CRETLD	1551	ECALL	1601
BOTTOM	0035	CRETST	2072	ECHOLS	1624
BUFBEG	3216	CRLF	7505	EFOP	0056
BUFFER	7470	CRUDDY	1155	EFUN	1743
BUFR	0060	CSTAR	0225	EFUN2	1755
BUFST	5531	D	0041	EFUN3	2021
C	0047	D256	0002	EFUN3I	0136
C100	0006	D85	0004	ELPAR	1764

END	0134	FILTAB	2135	FRST	3206
ENDFI	6243	FINCR	1065	FRSTX	3214
ENDLN	4556	FINDLN	4555	FSIN	5204
ENDT	0135	FINDN	2250	FSSERR	5774
ENREPL	1375	FINFIN	1137	FXIT	0000
ENUM	1732	FINISH	2076	G101	3661
EPAR	1710	FINKP	1133	G5772	3662
EPAR2	1766	FINPUT	0131	G5773	3663
ERASE	2206	FINT	4407	G7200	3664
ERG	2227	FISW	0052	G7773	3665
ERL	2224	FIVHUN	7653	G7774	3666
ERR2	2726	FIX	6724	G7775	3667
ERRFIL	2571	FIXM	6753	G7776	3670
ERROR2	4566	FLAC	0044	G7777	3671
ERROR3	4566	FLAD	6510	GAMMA	0005
ERROR4	4566	FLAG1	5162	GBLOK	3655
ERROR5	2725	FLAG2	4725	GECALL	1463
ERT	2216	FLAGJ	1076	GEND	2334
ERV	2221	FLARG	2032	GERR	0340
ERVX	2241	FLARGP	0125	GET1	2330
ERXIT	1457	FLDV	7107	GET3	2345
ESCA	2532	FLEX	6517	GETARG	1401
ETERM	1647	FLGT	6471	GETC	4545
ETERM1	1627	FLIMIT	1075	GETCX	1563
ETERM2	1655	FLINTP	6200	GETLN	4554
ETERMN	1644	FLIST1	0577	GETRHS	1000
EVAL	1613	FLIST2	0574	GETSGN	1045
EX1	0040	FLMY	6565	GETVAR	1405
EXIT	2646	FLOG	5040	GEXIT	0352
EXIT1	5034	FLOP	1674	GFND1	1510
EXIT2	5301	FLOUT	5556	GINC	0070
EXIT3	7363	FLOUTP	6000	GLIST	1375
EXITJ	2660	FLPT	6467	GO	5021
EXP	0044	FLSU	6507	GONE	0232
EXTR	2313	FLTONE	2405	GOODY	0045
F	0043	FLTXR	0014	GOTO	0603
FCONT	1101	FLTXR2	0015	GRPTST	0744
FCOS	5177	FLTZER	2407	GS1	1435
FCOUNT	5535	FM12	6142	GS2	1464
FEND3	2267	FNEG	5163	GS3	1444
FERROR	0177	FNOR	7000	GS4	1457
FEXP	4620	FNTABF	0374	GSERCH	1424
FEXT	0000	FNTABL	2167	GTEM	0021
FFF	1522	FNUM	6311	GZERR	0362
FG02	6011	FOR	1041	HINBUF	0037
FG03	6027	FORHUN	7651	HISS	0150
FG04	6034	FOUTPU	0130	HORD	0045
FG05	6070	FPAC1	7474	I33	2414
FIG01	6221	FPNT	6400	IBAR	0212
FIG04	6261	FPRNT	5465	IECALL	1037
FILERR	1354	FRETLD	1537	IF	1013
FILSTR	1326	FRETST	2014	IF1	1035

IF3	1025	LGO	6360	MCOM	1136
IGNOR	0217	LINENO	0067	MCR	0116
IGOTIT	1036	LIST3	0077	MD	5526
ILIST	0771	LIST6	0072	MEQ	1135
IN	5513	LIST7	0074	MF	0602
INBUF	0034	LISTGO	1366	MFLT	0117
INCALL	2071	LL	5173	MHUNDR	5375
INDEV	0064	LLENGT	1327	MIF	7260
INDRCT	6465	LLIST	6366	MINCMA	1400
INFIX	2401	LLOAD	1203	MINCOM	6374
INLIST	0570	LM	2572	MINE	5662
INORM	6307	LMAKE	1402	MINSKI	0051
INPUT	0756	LNAME	1172	MINUS2	7153
INPUTX	0271	LNUM	1171	MINUSA	0112
INSUB	0036	LO	5167	MINUSE	6301
INTEGE	0053	LOADIT	6333	MINUSZ	5663
INTRPT	2603	LOADJ	1304	MLDBLK	1165
IOBUF	3120	LOG2	5157	MLIMIT	7647
IPART	1040	LOG5	5142	MMCOM	7656
IRETN	0227	LOG6	5145	MOD	5214
ITABLE	6575	LOG7	5150	MODIFY	1254
ITER1	7470	LOG8	5153	MOO	1262
ITLOAD	1533	LOOP01	6433	MOOEND	1275
ITSAGO	1736	LOPEN	1431	MOOLUP	1266
ITSFF	1450	LORD	0046	MORNUM	1056
ITSOK	7521	LOSS	0151	MOVMOV	1305
ITSSS	1451	LPRTST	2037	MP1	7254
ITSTOR	2000	LS	6176	MP2	7256
ITSUU	1452	LSAVE	1233	MP3	7255
JUMP	6464	LSBLK	1324	MP4	7200
K5	5525	LTAPE	6346	MP5	7253
KINT	2625	LUKUP	1342	MP6	7210
L1	5126	LWETMP	0002	MPER	0115
L2	5131	LXIT	1416	MPLUS	5664
L3	5134	M100	0101	MSPACE	5665
L4	5137	M10PT	6147	MULDIV	7101
LASTLN	0025	M11	0121	MULT	6570
LASTOP	0055	M12	2413	MULT10	5667
LASTV	0031	M137	2357	MULT2	5715
LC	5171	M140	2556	MULTY	4752
LCHAIN	1202	M144	6137	MVCNT	1323
LCLOSE	1520	M2	0111	MVCTR	1200
LCON	0371	M20	0105	MVPTR	1201
LDMILD	1160	M240	0114	MYAC1	0164
LEFLAG	1462	M260	1534	MYAC2	0165
LEFPUT	0172	M272	1544	MYAC3	0166
LEPUT	6163	M4	6141	MYTEMP	0156
LERR	6357	M40	2356	MYTMP2	0157
LESUB2	0170	M43	1077	NAGSW	0065
LESUBS	0173	M5	0120	NCHARS	7566
LG	6375	M77	0103	NCOLS	7564
LG2E	4713	MBREAK	2602	NEGP	4724

NFEEDS	7565	ODISSP	7704	P7200	1402
NLINES	7561	OE	7753	P7600	0104
NOASCI	0061	OERROR	7713	P77	0122
NOCLK	2653	OEXIT	7731	P7700	0101
NOCRLF	7510	OGO	7714	P7740	0372
NOHANG	7556	OI	7734	PA1	2524
NORF	6515	OLIST	7722	PACBUF	2502
NORM	6571	OM12	5530	PACKC	4546
NORMF	7147	ONE	4716	PACKST	0027
NORMLE	2031	0010	1425	PACX	2530
NOTSAV	1314	0012	1430	PALG	5260
NOX	6675	002	1456	PARTES	2051
NOX1	6711	006377	7730	PASS	6335
NOX2	6704	OP	3115	PB1FLG	0163
NUMSGN	1061	OPMINS	6567	PC	0022
01	3600	OPNEXT	1622	PC1	0614
010	1123	OPTABL	1731	PCHAR	1401
012	1545	OPTR	6002	PCHECK	5244
015	1434	OPTR0	2663	PCHK	0510
0200	0003	OPTRI	2665	PCK1	2535
0215	1157	OPTR0	2664	PCLEAR	0175
027	1565	OPUT	5532	PCLKFL	7745
0360	0007	OS	7763	PCOMMO	0154
037	1360	OSAMP	1357	PD2	0534
04377	0076	OT	7771	PD3	0554
04600	5374	OUT	2465	PDLXR	0013
056	1156	OUTA	5536	PECALL	6334
06000	0173	OUTCR	2476	PEQ	6135
06377	7570	OUTDEV	0063	PER	0102
07	1776	OUTDG	6154	PFILTA	0152
07000	7415	OUTPUT	7706	PFINIS	0160
07400	7650	OUTX	2475	PFNEW	0410
07420	0174	OVER1	0043	PFNUM	1771
07453	1426	OVER2	0047	PFX	0411
07472	1501	P	0000	PFZ	0412
07506	1427	P13	0005	PGETC	1422
07510	1424	P17	0107	PGETRH	0143
07524	1154	P177	0106	PI	5311
07566	7572	P1FLAC	0167	PI2	5036
0760	0015	P2000	0373	PIOT	5315
07655	7571	P27	6750	PLCE	5536
077	1124	P277	0110	PLDMIL	0144
07710	1125	P2FLAC	0170	PLEFLA	1075
07716	7573	P3	2036	PLESUB	0101
07761	1155	P337	0075	PLLP1	1006
07763	7567	P377	2553	PLLP2	1016
07764	0172	P3FLAC	0171	PLLP3	1044
07770	1126	P40	2552	PLLP4	1102
07774	1127	P4000	0124	PLNAME	1122
OC	7752	P43	6310	PLNUM	0142
OCTNUM	1101	P5LNAM	0145	PLOOKU	0153
OD	7761	P6LNAM	0146	PNCHAR	7732

PNCOLS	7776	RESOLV	7173	SRETST	2024
PNFEED	7777	RET	5452	SRNLST	1361
POPA	1413	RETRN	1563	START	0177
POPF	4544	RETURN	5536	STARTL	5064
POPJ	5541	REVIT	7146	STARTV	0060
POPTR	7733	RHSERR	1130	STEMP	7750
PP43	1100	RITEOU	3651	STEMP2	7751
PPASS	7705	RND2	5527	STOKOK	2044
PPROC	1421	ROOTGO	7461	STO0BG	2051
PPTEN	6144	ROT	2557	STORIT	6175
PPTR	7574	ROUND	6151	SUBR	0102
PREDIV	1622	RTL6	4557	SUBS	0171
PREPLA	0155	RUB1	3004	SUBS2	0167
PRHSER	1423	RUB2	3042	SWITCH	0161
PRINTC	4551	RUB3	3030	SWTMP	0162
PRNT	2442	RUB4	3037	T	0000
PRNT2	3114	RUB5	3041	T1	0032
PRNTI	6132	RUBIT	2555	T12	3611
PRNTLN	4553	SADR	6150	T2	0071
PROC	0611	SAMEN	1372	T3	0033
PROCES	0610	SAVAC	2600	TABLE	6466
PSCOPO	7775	SAVE	3751	TAG1	6723
PSETCL	7746	SAVLK	2601	TASK	1202
PSIN	0165	SBAR	1300	TASK4	1250
PSTART	1322	SCHAR	1271	TCRLF	1246
PSUBS	0100	SCONT	1266	TCRLF2	1243
PT1	0030	SCOPOU	7500	TDUMP	3052
PTBL	1330	SCOUNT	5534	TELSW	0016
PTCH	0126	SETCLK	5351	TEM	5156
PTEEN	6275	SETT	1041	TEMP	4726
PTEST	1462	SEX	1336	TEN	6271
PUSHA	4542	SEXC	0740	TENPT	6152
PUSHF	4543	SFOUND	1304	TERMS	1772
PUSHJ	4540	SGOT	1310	TEST2	6736
PWAIT	0174	SIGN	7124	TEST4	7366
PXOUTL	7774	SIGNF	0050	TESTA	0322
QADD	0061	SIN	2662	TESTC	4564
R6	5441	SMIN	6136	TESTN	4561
RANMUL	6160	SMP	6101	TEXTP	0017
RANO	1142	SMSP	6134	TGO	5400
RAR1	6573	SORTB	1312	THIR	7257
RAR2	6574	SORTC	4550	THISLN	0023
RDIV	0152	SORTCN	0054	THISOP	0024
READC	4552	SORTJ	4547	TINTR	1236
RECOVR	2740	SPECIA	6777	TLIST	1376
RECOVX	2761	SPLAT	3051	TLIST2	1532
REMAIN	5712	SPNOR	4560	TLIST3	2377
REPLAC	1361	SPTR	7671	TQUOT	1227
REPT	6146	SQCON1	7467	TRAD	6575
RESOL	6752	SQEND	7465	TSTGRP	4563
RESOL3	7376	SRETLD	1541	TSTLPR	4562
RESOL5	6304	SRETN	0261	TWO	4721

TWOPI	5305	XRAN	1145
TYPE	1201	XRAR2	7365
TYPE2	1223	XRT	0011
URETLD	1546	XRT2	0012
URETST	2066	XRTL6	0413
UTE	2276	XSGN	2012
UTQ	2305	XSORTC	0721
UTRA	2274	XSPNOR	1535
UTX	2316	XSQ2	4676
UZERST	2064	XSQR	5325
VAL	0032	XSQRT	7400
WAIT	7657	XT3	0717
WAITER	0020	XTESTC	0700
WAITLP	0115	XTESTN	1546
WALL	0664	XYZ	2451
WEXIT	0072	Y	0077
WORDS	0003	ZERO	6522
WRITE	0635		
WTEST2	0653		
WTESTG	0667		
WX	0673		
X	5321		
X1	5035		
X2	4675		
X7774	0140		
X7775	0141		
XABS	2016		
XADC	1341		
XCT	0020		
XCTIN	0062		
XDELET	2064		
XDISP	7602		
XENDLN	2360		
XFIND	2244		
XGETLN	0302		
XGETOU	1254		
XI33	2666		
XIN	6306		
XINPUT	5666		
XINT	1156		
XLC	0130		
XLG	0136		
XLL	0132		
XLO	0126		
XLS	0134		
XOUTL	2676		
XPOPJ	1565		
XPRNT	2425		
XPUSHA	0477		
XPUSHJ	0521		
XQ	0001		
XR1	0010		

COMMAND AND OPERATION SUMMARY

A.1 Commands

<u>Command</u>	<u>Form</u> ¹	<u>Explanation</u>
ASK	A "X,Y,Z" X,Y,Z	Types "X,Y,Z" and then a colon for each variable; the user types a value to define each variable.
COMMENT	C	Ignores any line beginning with C.
DO	D 4.1	Executes line 4.1; returns to command following DO command.
	D 4.0	Executes all group 4 lines, or until a RETURN is encountered; returns to command following DO command.
ERASE	E	Erases the symbol table.
	E 2.0	Erases all group 2 lines.
	E 2.1	Deletes line 2.1.
	E A	Deletes all user input.
FOR	F I=x,y,z;command(s)	Executes the command for all values of I where x is the initial value, y is the increment to be added to x, and z is the limiting value for I.
	F I=x,z;command(s)	If y is not specified, an increment of 1 is assumed.
GO	G	Starts indirect program at lowest numbered line.
	G 3.4	Starts indirect program (transfers control to line 3.4).
	G ?	Starts at lowest numbered line and traces entire indirect program until another ? is encountered, or until an error is encountered, or until completion of program.
IF	IF (X)Ln,Ln,Ln	Where X is a defined identifier, a value, or an expression followed by up to three line numbers.
		If X is less than zero, control is transferred to the first line number.
		If X is equal to zero, control is transferred to the second line number if present; otherwise to the next command.

¹Short form is presented. Of course, the complete spelling of each command may be used to improve readability.

<u>Command</u>	<u>Form</u>	<u>Explanation</u>
		If X is greater than zero, control is transferred to the third line number, if present; otherwise to the next command.
LIBRARY CLOSE	L C,Fn	Closes active file number (F0, F1,..., or F7).
LIBRARY GO	L G,name,unit	Loads and starts program <u>name</u> from <u>unit</u> .
LIBRARY LOAD	L L,name,unit	Loads program <u>name</u> from <u>unit</u> and returns control to the user.
LIBRARY MAKE	L M,length,name,unit	Creates a file of <u>length</u> blocks with this <u>name</u> on <u>unit</u> .
LIBRARY OPEN	L O,Fn,format,name,unit	Declares that file <u>name</u> on <u>unit</u> is to be referenced as Fn (F0,F1,..., or F7); data is to be interpreted as <u>format</u> (Signed Fractions, Signed <u>I</u> ntegers, or <u>F</u> loating Point).
MODIFY	M 1.15	Enables editing of any character on line 1.15 (refer to section A.3.2),
OUTPUT CLEAR	O C	Erases everything from scope.
OUTPUT DELAY	O D	Delays computation to refresh the display scope.
OUTPUT ERASE	O E	Erases typed output from the display scope, leaving FDIS output.
OUTPUT INTERVAL	O I,n	Starts the clock "tic"ing every n seconds ($.01 \leq n \leq 40.95$).
	O I	Delays computation until the next clock tic.
OUTPUT SCOPE	O S	Places all subsequent typed output on scope.
OUTPUT TELETYPE	O T	Places all subsequent typed output on Teletype.
QUIT	Q	Returns control to the user.
RETURN	R	Terminates DO subroutines, returning to the original sequence.
SET	S A=5/B*C	Sets the variable, A, equal to the value of the expression to the right of the equal sign.

<u>Command</u>	<u>Form</u>	<u>Explanation</u>
TYPE	T A+B-C	Evaluates expression and types out result in current output format (refer to section A.3.1).
	T A-B,C/E	Computes and types value of each expression separated by commas.
	T "TEXT STRING"	Types the text enclosed in the quotes. May be followed by ! to generate carriage return-line feed.
WRITE	W W A	Types out the entire indirect program.
	W 1.0	Types out all group 1 lines.
	W 1.1	Types out line 1.1
A.2 Functions		
Square Root	FSQT(x)	Where x is a positive number or expression greater than zero.
Absolute Value	FABS(x)	FOCAL-12 ignores the sign of X.
Sign Part	FSGN(x)	Evaluates the sign part only, with 1.0000 as integer.
Integer Part	FITR(x)	Operates on the integer part of x, ignoring any fractional part.
Random Number Generation	FRAN(x)	Generates a random number.
Exponential Function (e^x)	FEXP(x)	Generates e to the power x (2.71828^x).
Sine	FSIN(x)	Generates the sine of x radians.
Cosine	FCOS(x)	Generates the cosine of x radians.
Arc Tangent	FATN(x)	Generates the arc tangent of x radians.
Logarithm	FLOG(x)	Generates the $\log_e(x)$.
Analog-to-Digital	FADC(n)	Reads analog-to-digital channel n and records the value.
Display	FDIS(x,y)	Displays the X,Y point on the display scope.
User Functions	FNEW FX FZ	User defined machine language subroutines.

A.3 FOCAL OPERATIONS

A.3.1 Format

To set output format, TYPE % x.y where x is the total number of digits, and y is the number of digits to the right of the decimal point.

TYPE % resets output format to floating point.

To type symbol table, TYPE \$ other statements may now follow on this line.

A.3.2 MODIFY Operations

After a MODIFY command, the user types a search character, and FOCAL-12 types out the contents of that line until the search character is typed. The user may then perform any of the following operations.

- a. Type in new characters. FOCAL-12 will add these to the line at the point of insertion.
- b. Type a CTRL/L. FOCAL-12 will proceed to the next occurrence of the search character.
- c. Type a CTRL/BELL. After this, the user may change the search character.
- d. Type RUBOUT. This deletes characters to the left, one character for each time the user strikes the RUBOUT key.
- e. Type ←. Deletes the line over to the left margin but not the line number.
- f. Type RETURN. Terminates the line, deleting characters over to the right margin.
- g. Type LINE FEED. Saves the remainder of the line from the point at which LINE FEED is typed over to the right margin.

A.3.3 The Trace Feature

<u>Special Character</u>	<u>Example of Form</u>	<u>Explanation</u>
?	?...? or ?...	Those parts of the program enclosed in question marks will be printed out as they are executed. If only one ? is inserted, the trace feature becomes operative, and the program is printed out from that point until another ? is encountered, until an error is encountered, or until program completion.

A.3.4 Special Characters

1. Mathematical Operators (in order of precedence)

↑	Exponentiation	
*	Multiplication	
/	Division	
+	Addition	} Same priority
-	Subtraction	

2. Control Characters

%	Output format delimiter	
!	Carriage return and line feed	
#	Carriage return	
\$	Type symbol table contents	
()	Parentheses	
[]	Square brackets	(mathematics)
< >	Angle brackets	
" "	Quotation marks	(text string)
? ?	Question marks	(trace feature)

3. Terminators:

SPACE key (names)	} (nonprinting)
RETURN key (names)	
ALT MODE key (with ASK statement)	
, Comma (expressions)	
; Semicolon (commands and statements)	

A.4 FOCAL-8 Features not in FOCAL-12

1. There is no initial dialogue; the mathematical functions are retained.
2. The CLINE overlay is not included. FDIS provides the display function.
3. The PLOTR routine can be reoriginated, reassembled and added if necessary.
4. The 4WORD, 8K, LIBRA, GRAPH and QUAD overlays do not apply to FOCAL-12.
5. The TYPE command does not type an "=" before typing the value.
6. FOCAL-12 does not support the high speed paper tape reader.

DATA FORMAT SPECIFICATION

1. "F" is the standard FOCAL-8 floating point format¹, providing 6 digit accuracy² with absolute value being 0 or any value between 10^{-615} and 10^{+615} .
2. "S" provides signed fraction with 6 digit accuracy², with absolute value between 0.0 and 1.0 including 0.0.
3. "I" provides for integer values between -2048 and +2047, inclusive. -2048 follows 2047 and vice versa.
4. In "I" and "S" formats not all possible values can be stored in the desired format: in "S" format, numbers outside the range are given the minimum or maximum values; in "I" format, numbers are integerized and stored as noted above.

Consider the following examples.

a) 1.01 L O,F1,I,DEMO,0
 1.02 S F1(0)=2050; T F1(0),!
 GO

would type -2046.0000

b) 1.01 L O,F1,I,DEMO,0
 1.02 S F1(0)=-2049; T F1(0),!
 GO

would type 2047.0000

c) 1.01 L O,F1,S,DEMO,0
 1.03 S F1(0)=6.0E20; T F1(0),!
 GO

would type 1.0000 (stored as .9999998)

d) 1.01 L O,F1,S,DEMO,0
 1.03 S F1(0)=-7.0E20; T F1(0),!
 GO

would type -1.0000 (stored as -.9999998)

¹Indeed, the FOCAL-8 floating point package is used; problems such as exponent overflow remain.

²Actually, 6.8 digit accuracy.

³The I value stored formally is: $I' = [(I + 2048) \bmod 4096] - 2048$.

FOCAL-12 I/O AND THE L C COMMAND

FOCAL-12 does not actually update the file on tape or disk each time a reference to the file is made (indeed, performance would be intolerably slow if it did). In general, the last two blocks referenced by the program are maintained in memory. For example, the program

```

1.01  L O,F0,F,DATA1,0
1.02  L O,F1,F,DATA2,1
1.03  L O,F2,F,DATA3,2
1.04  F I=0,100; S F0(I)=I
1.05  F I=0,100; S F1(I)=I/2
1.06  F I=0,100; S F2(I)=I/4

```

would, at this point, have set the first 100 entries of DATA1 and DATA2 on tape but would not yet have updated file DATA3 on tape 2. These two blocks are still in core memory. The advantage here is that, for line 1.06, FOCAL-12 has executed tape operations only for $I=0$ and $I=85$. For all other references to F2, the required tape block image was already in memory. If the next line in the program were 1.07 S A=F2(1); S B=F2(150) there would be no tape operation required since the image of the first two blocks of DATA3 are still in core memory. This presents no problem to the user as long as FOCAL-12 is operating and tape 2 is not dismounted. At the end of the program, he merely CLOSEs the OPEN files. Further file references, not to the first 170 entries of DATA3, would cause these tape block images to be written on tape 2 to make room for the new tape block images, etc.

MISCELLANEOUS COMMENTS

1) Starting Block Number

The standard procedure for defining the starting block number is via the LIBRARY OPEN command; e.g.

```
1.01 L O,F0,S,#100,0
1.02 F I=0,1,100; S F0(I)=...
```

It is sometimes desirable to define this starting block number at the time the program is run. To do this

```
1.01 L O,F0,S,#0,0
1.02 A "STARTING BLOCK?" N
1.03 F I=0,1,100; S F0(128*N+1)=...
```

Thus, if the operator typed in "10" when starting block was asked, the base index for F0 would be 1280 numbers or 2560 words of 10 blocks, so that effectively the file starts at block 10. Note that the L O command specifies block number in octal while the response to the ASK statement is interpreted as decimal.

2) Deletion of EXP, ARCTAN, LOG

For those users who do not need the above functions, the following patch can be used to delete them from FOCAL-12, increasing the user space by almost 30%. Using LAP6-DIAL-MS

```
→ZE↵
→AB FOCAL-12,unit↵
→AS PATCH,unit↵
→SB NEWFOCAL,unit,P↵
```

where "PATCH" is

```
PMODE
*35
5166
*402
2725
2725
2725
FIELD 1
*1225
1067
*1247
```

1173
*1323
6017
LISTAP -7

Note that this makes a new FOCAL-12! Programs saved under the standard FOCAL-12 will not load under this version and vice versa.

FOCAL-12 ERROR DIAGNOSTICS*

<u>Code</u>	<u>Meaning</u>
?00.00	Manual start given from console.
?01.00	Interrupt from keyboard via CTRL/C.
?01.40	Illegal step or line number used.
?01.78	Group number is too large.
?01.96	Double periods found in a line number.
?01.:4	Group zero is an illegal line number.
?01.:5	Line number is too large.
?02.32	Nonexistent group referenced by 'DO'.
?02.52	Nonexistent line referenced by 'DO'.
?02.79	Storage was filled by push-down list.
?03.05	Nonexistent line used after 'GOTO' or 'IF'.
?03.28	Illegal command used.
?04.39	Left of "=" in error in 'FOR' or 'SET'.
?04.52	Excess right terminators encountered.
?04.60	Illegal terminator in 'FOR' command.
?05.46	Bad argument to 'MODIFY'.
?06.03	Illegal use of function or number.
?06.57	Storage is filled by variables.
?07.22	Operator missing in expression or double 'E'.
?07.38	No operator used before parenthesis.
?07.;0	No argument given after function call.
?07.;7	Illegal function name or double operators.
?08.49	Parentheses do not match.
?09.13	Bad argument in 'ERASE'.
?10.<1	Error in DIAL file reference; name not found or no room left on unit.
?10.:5	Storage was filled by text.
?11.35	Input buffer has overflowed.
?20.34	Logarithm of zero requested.
?20.36	Log of a negative number requested.
?23.<4	Subscript error in file reference or undefined file number or reference.
?23.36	Literal number is too large.
?25.81	File number not OPENed.
?25.;1	Syntax error on LIBRARY command.
?26.:1	Exponent overflow. Too many FDIS points.
?28.73	Division by zero requested.
?30.05	Imaginary square root required.
?31.12	Missing argument in display command.
?31.23	Too many FDIS points.
?31.75	Syntax error on OUTPUT command.
?31.<7	Illegal character, unavailable command, or unavailable function used.

*For FOCAL-12 only.

INDEX

- Addressing Data, 4
- Analogue Input, 10, 12
- A/D Channels, 1,10
- A/D Function 10,11

- Binary Files, 4, 8, 9
- Block Numbers, 5, 6, E-1

- Chaining, 9
- Characters, A-5
- Checking Tape Write Feature, 12
- Clock, 1, 11
- Command Summary, A-1

- Data Acquisition, 11
- Data Files, 5, 6, 7
- Data Storage and Retrieval, 4, 5, 9, 11, 12
- Delays, 10
- Deleting Functions, E-1
- DIAL, 1, 4, 5, 6, 8, 9
- Disks, 1, 4, 5, 6, 8, 12
- Display Function, 2, 3, A-3

- Editing, 4, 9
- Error Messages, 13, F-1

- FADC, 10, 11, A-3
- FDIS, 2, 3, A-3
- File Names, 5, 8
- File Numbers, 6, 7, 8
- Files, Binary, 4
 - Data, 5, 6
- Floating Point Numbers, 5, 6
- FNEW, 14
- FOCAL-8, 1
- Format, Data, 5, 6, C-1
- FX, 12, 14, 16
- FZ, 14

- Hardware Requirements, 1

- Intervals, 10
- Introduction, 1

- LIBRARY CLOSE, 7, A-2, D-1
- LIBRARY GO, 9, 18, A-2
- LIBRARY LOAD, 9, 18, A-2
- LIBRARY MAKE, 5, A-2
- LIBRARY OPEN, 6, 8, A-2, E-1
- LIBRARY SAVE, 8, 18, A-2
- LINctape, 1, 4, 5, 6, 8, 12, D-1

- Line Feed, 3
- Links to FOCAL, 16
- Loading Procedure, 1, B-1

- Memory Buffer, 11, D-1
- MODIFY Operations, A-4

- Nonexistent Device, 6

- Operation Summary, A-4
- OUTPUT CLEAR, 4, A-2
- OUTPUT DELAY, 3, A-2
- OUTPUT ERASE, 4, A-2
- OUTPUT INTERVAL, 11, A-2
- OUTPUT SCOPE, 2, A-2
- OUTPUT TELETYPE, 3, A-2
- Overlays, 18

- Plotting, 2
- Program Preparation, 4
- Program Storage, 8, 16

- Rate, 13
- REPLACE, 5, 9
- Restarting FOCAL, 6
- Returning to DIAL, 6

- Sampling Channels, 10, 12
- Scope Capacity, 2, 10
- Scope Refreshing, 3
- Signed Fractions, 5, 6, C-1
- Signed Integers, 5, 6, C-1
- Startup Procedure, B-1
- Subroutines, 14
- Subscripting, 5, 6
- Sync, 13
- System Tape, 1

- Tag Table, A-4
- Terminators, A-5
- Text Handling, 8, 14
- Throughput - \$THRUF, 12
- Timing, 10
- Tracing, A-4

- Unit Numbers, 5
- User Coded Functions, 16
- User Functions, 12, 14, 18, A-3

- Variables, 5, 6, 16

READER'S COMMENTS

Digital Equipment Corporation maintains a continuous effort to improve the quality and usefulness of its publications. To do this effectively we need user feedback -- your critical evaluation of this manual.

Please comment on this manual's completeness, accuracy, organization, usability, and readability.

Did you find errors in this manual? _____

How can this manual be improved? _____

DEC also strives to keep its customers informed of current DEC software and publications. Thus, the following periodically distributed publications are available upon request. Please check the appropriate boxes for a current issue of the publication(s) desired.

- Software Manual Update, a quarterly collection of revisions to current software manuals.
- User's Bookshelf, a bibliography of current software manuals.
- Program Library Price List, a list of currently available software programs and manuals.

Please describe your position. _____

Name _____ Organization _____

Street _____ Department _____

City _____ State _____ Zip or Country _____

HOW TO OBTAIN SOFTWARE INFORMATION

Announcements for new and revised software, as well as programming notes, software problems, and documentation corrections are published by Software Information Service in the following newsletters.

**Digital Software News for the PDP-8 Family
Digital Software News for the PDP-9/15 Family
PDP-6/PDP-10 Software Bulletin**

These newsletters contain information applicable to software available from Digital's Program Library.

Please complete the card below to place your name on the newsletter mailing list.

Questions or problems concerning DEC Software should be reported to the Software Specialist at your nearest DEC regional or district sales office. In cases where no Software Specialist is available, please send a Software Trouble Report form with details of the problem to:

**Software Information Service
Digital Equipment Corporation
146 Main Street, Bldg. 3-5
Maynard, Massachusetts 01754**

These forms, which are available without charge from the Program Library, should be fully filled out and accompanied by teletype output as well as listings or tapes of the user program to facilitate a complete investigation. An answer will be sent to the individual and appropriate topics of general interest will be printed in the newsletter.

New and revised software and manuals, Software Trouble Report forms, and cumulative Software Manual Updates are available from the Program Library. When ordering, include the document number and a brief description of the program or manual requested. Revisions of programs and documents will be announced in the newsletters and a price list will be included twice yearly. Direct all inquiries and requests to:

**Program Library
Digital Equipment Corporation
146 Main Street, Bldg. 3-5
Maynard, Massachusetts 01754**

Digital Equipment Computer Users Society (DECUS) maintains a user Library and publishes a catalog of programs as well as the DECUSCOPE magazine for its members and non-members who request it. For further information please write to:

**DECUS
Digital Equipment Corporation
146 Main Street
Maynard, Massachusetts 01754**

Send Digital's software newsletters to:

Name _____
Company Name _____
Address _____
(zip code) _____

My computer is a

PDP-8/I	<input type="checkbox"/>	PDP-8/L	<input type="checkbox"/>
LINC-8	<input type="checkbox"/>	PDP-12	<input type="checkbox"/>
PDP-9	<input type="checkbox"/>	PDP-15	<input type="checkbox"/>
PDP-10	<input type="checkbox"/>	Other	<input type="checkbox"/> _____ Please specify

My system serial number is _____ (if known)

**Digital Equipment Corporation
Maynard, Massachusetts**

