



is a simple but powerful language to generate textual displays using Spectrum. It uses range from advertising to education, or just for own absorbing interest.

commands are entered as RDM statements, so that all the facilities of natural Spectrum editor are available. DLAN itself is pure machine code, invoked using a USR function. The commands are generally interpreted after another in a cyclic fashion, so that if left to itself DLAN gives the ever-ending sequence of effects.

central concept is the WINDOW, defined as any rectangle of whole lines and columns within the 24 x 32 character display. Any number of windows can be defined, overlapping or otherwise. Effects such as text display scrolling (four directions to choose from) operate within the currently defined window, leaving the rest of the screen unchanged.

Another major feature of DLAN is its repertoire of print styles or fonts, many of them large size and even some with proportional spacing. (I.e. the letters 'M', 'A', 'P' can be of different widths in a font.) The AEX version of DLAN has a bigger choice of fonts than the 16K version.

DLAN is also a word processor: it maintains a clean left-hand margin when printing text, and minimises word breaks. Thus a professional display of text is possible with the minimum of effort.

Command Structure

DLAN commands are single letters or symbols, optionally prefixed with a number up to 255 which is a repetition factor. For example, 'S' means scroll up 1 line, and '15S' means scroll up 15 lines.

Light commands are followed by text or parameters. For example, 'HELLO' means print (command 'P' symbol) the phrase 'HELLO'. And 'ASD' means scroll (command 'S') down 4 lines. 'D' is the Down parameter.

Commands are entered within Basic RDM statements. You can have any number of commands in a RDM statement, using semi-colon (;) as a delimiter. For example...

```
20 RDM: J=HELLO:USR:P
```

This prints the word 'HELLO' 3 times; scrolls the window 4 columns to the right; and then pauses for 1 second (P) before continuing with the next command. Each command could have written as a separate RDM, with identical result. But combining commands into fewer RDMs can save space.

In general, DLAN interprets commands sequentially, but it also offers a facility equivalent to Basic GOSUB. This allows you to encode a labelled sequence of commands, and execute it from any number of places within the code. And 15: 10 levels of nesting are supported. I.e. you can GOSUB to a routine which in turn has its own GOSUB calls, etc.

The RDM statements will normally get automatically renumbered in terms when DLAN USR code is invoked, so insertions are never a problem.

Alphabetic DLM commands, and alphabetic parameters, may be entered in upper case or lower case interchangeably.

One more general point: DLM never stops because of user errors. Instead, it either ignores a bad parameter or command, or takes some default action. You will not make DLM crash.

We will now describe each of the commands fully.

W: Define or recall a Window

Wrabbccdd
Wabbbccdd
Wr

where 'r' is optional reference letter; you can recall a window defined fully elsewhere just by referring to its letter. Upper case and lower case are differentiated here,

and where 'aa' is start screen line (00 = top, 23 = bottom),
'bb' is number of screen lines (max = 24-aa),
'cc' is left-most screen column (00 to 31),
'dd' is number of screen columns wide (max = 32-cc).

Note that when aa,bb etc are less than 10, you must give a leading zero.

Before interpreting the first command, DLM assumes the current window to be the full screen, as if you had started with a command:

W00240032:

The use of reference letter is highly recommended: you can define all your windows in one REM; and then refer to them where needed with the short 'r' method. And if a window has to be changed (i.e. you want to define it elsewhere) then there is only one change to make to the DLM commands. For example, suppose we have a window at line 5 for 6 lines, column 10 for 15 lines, we can define it as:

W050615:

Now, at any place in the DLM command program, we can recall window "A" just by coding:

WA: (or, wA; but not wE; since the small "a" does not match the big "A".)

E: Edge the current window

En

where 'n' is a digit 1 to 9 denoting one of 9 styles in which to make a frame around the window currently defined. Further, after making the frame DLM then shrinks the window by 1 all round, so that the frame is then not affected by subsequent scrolls etc. within the window. (If the window is already too small in one direction, this shrink is suppressed.) Rather than list all the styles, we leave it to you to explore them. Note that the use of repetition factor causes progressive shrinkage, e.g.:

528:

Cn: or Cnm:

where 'n' and 'm' are colour codes 0-7 as defined on your Spectrum keyboard. The first or only colour code is deemed to be PAPER colour. The second code, if present, is INK colour. If INK colour is not given, it is left unchanged.

This command does not have immediate effect, but rather affects subsequent commands such as FILL and Print (q.v.)

P: Fill the current window

Px: or P:

where 'x' is the character or graphic which DLM then fills the window with. If no fill character is given, then the only effect is to repaint the window in whatever are the currently defined colours, as last given via 'C' command.

Note that by alternately filling and changing colour you can get very pleasing effects. And by mixing in colours where INK and PAPER are the same, you can create an on-off flashing effect, as opposed to the FLASH swap-the-colours effect. e.g.:

C17:P:C77:P:C17:P

The fast way to clear a window is to fill with SPACE character.

=: Print text

=Any text you like....:

This is the way to display text, scrolling upwards from the lowest part of the current window. A fresh line is started for each '=' command. DLM word-processes to keep the left margin tidy and to minimise word breaks at end of line. Any strings of 2 or more spaces are reduced to a single space. The current type font is used ('?' q.v.) and where appropriate DLM employs proportional spacing.

The use of a repetition factor causes the whole paragraph to be repeated, for example:

5=Hello everybody:

produces the effect:

Hello everybody
Hello everybody
Hello everybody
Hello everybody
Hello everybody

(And not: Hello everybodyHello everybody...)

The text can include graphic characters and UDCs. But DLM has five of its own UDCs as follows:

UDC Graphics: Q R and S are styles of Grey (try them)
UDC Graphic F is a colon (:)
UDC Graphic U is a semi-colon (;)
The reason for the last two is that direct/colon confuses the basic edit.
And the semi-colon confuses DLM!

Do not use direct colour codes, INT VIDEO, or AT and EAS tokens within your text, because DLAN simply converts these codes into spaces. But this does provide a way to show leading spaces, as in a paragraph start. Try it.

By using the 'grey' patterns of UPC graphics Q/R/S with various combinations of PAPER and IRK colour, you can get some 'new' Spectrum colours. Note especially that although Graphic Q and R are almost identical pixel stripes, they give quite different colour effects.

<: Print Right-to-Left

<Any text you like.....;

This is the other way to display text, on a 'moving belt' system. After an initial scroll of the current window, text is built and scrolled right-to-left along the bottom of the window, until the last character of text is printed. Any over-run is simply dropped off at the left-hand boundary of the window. Word-processing logic is not used except with respect to proportional spacing of letters in some fonts.

S: Scroll the window contents

S: or SU; or SD; or SL; or SR;

The contents of the current window are scrolled by one line of column in the direction indicated. To scroll by more lines or columns, use the repetition factor, for example:

10SR;

which scrolls 10 columns to the right.

'f' by itself is taken to mean 'SU;' (Scroll Up)

B: Set Border Colour

Ba;

where 'n' is colour code 0-7 being the Spectrum code of the colour to which the TV screen border is set. The effect is immediate. If no 'B' command is given, DLAN assumes J. (Magenta)

A: Set other Spectrum Attributes

A....;

One or more of the following parameters may be given:

B = Bright D = Dull P = Plash S = steady

For example, to set bright and flash:

ABF; or AFB; or AF:AB;

DLAN starts off by assuming 'ABS', i.e. bright and steady. The effect of 'A' command is delayed in the same way as 'C' command.

F: Pause For 1 second

P: (no parameters)

For longer pauses, use the repetition factor, e.g. to pause 15 seconds:

T: Type Font Select

Tn;

DLAN has a repertoire of different print styles and sizes, for use with the 'n' and '<' commands. The 16K version of DLAN has four styles, selected by T1; T2; T3; T4; respectively - in addition to the standard Spectrum font. The 48K version has these plus a further seven via codes 5,6,7,8,9,A,B.

If the Type code is not matched, or if just 'T:' is given, then DLAN reverts to standard Spectrum font.

Most styles have upper case only, and one has lower case only. Any text character not matched is replaced with upper/lower case if possible to find a match. If this fails, the individual character is shown in standard Spectrum font. In the case of the normal-size font this may look quite acceptable.

Some fonts have special proportional spaced letters, especially M and which are wider than the rest, and l and i and period which are kept single-column width. The full set of fonts is as follows.

T1: a very pretty 1 x 1 Serif, upper case only, 0-9, Et.

T2: a clear 2 x 2 Serif, upper case only, 0-9, Et.

T3: a highly decorative modern font in 2 x 2, upper case only, 0-9, Et.

T4: a magnetic ink style in 1 x 1, upper case only, 0-9, Et.

T5: a chunky bold 1 x 1, upper case only, 0-9, Et.

T6: full character set 2 tall x 1 wide in Sans Serif

T7: another 2 x 1, upper and lower case, very elegant

T8: Bold 2 x 1 in upper case only, 0-9, Et.

T9: beautifully clear 3 x 1 in lower case only, 0-9, Et.

TA: Gracefully-proportioned 3 x 2 Serif, upper case only, 0-9, Et.

TD: strikingly effective 3 x 2 Shadow face, upper case only, 0-9, Et.

M: Set Command Delay

M: (no parameters)

DLAN waits for about 1/5th second between commands, or between repeats of a command, but you can alter this interval at will to any number of TV frames between one and 255 (0-5 seconds) by giving a repetition factor being the number of TV frames. So for the fastest pace, use just 'P;' and for the slowest pace, use '255M;'. Although the individual effects, such as scrolling, are at a predetermined pace, the use of a affects the overall pace of the sequence of commands. You can change pace with as many M commands as you like. DLAN starts off with an assumed value of 10 - about 1/5th second.

f: COSUB equivalent

fX;

where 'x' is the one-letter label of a DLAN command subroutine. DLAN hunts for a command of 'fX' and transfers its processing to that command if found. 'x' may be upper case or lower, and these are differentiated

'x':
 There 'x' is the one-letter label to which a ↑ command refers. You can have labels of any length, but only the first character is examined by DIAM.

R: Return to ↑ caller

R:
 This is analogous to the Basic RETURN statement. But since DIAM has no "GOTO" equivalent, you must also use 'R:' to separate your top-level DIAM commands from any subroutines which follow. In other words, DIAM restarts at the beginning if there is no ↑ corresponding to an 'R:'.

Notice that DIAM supports up to 10 levels of nesting. This means that a 'x:' routine can itself also call other routines using 'Y:', and these in turn can call others - up to 10 layers deep.

Another noteworthy aspect of DIAM ↑ command is that it can take a repetition factor. DIAM keeps track of the number of repetitions at all levels. An interesting application is to use 'n:n;R:n;' etc. at the beginning, and '...:R:' at the end of the main sequence, so that DIAM very rarely goes back to the exact beginning. This may be of use to prevent DIAM's automatic re-definition of windows and colours at the start of each cycle.

A final note on subroutines: It is easy to forget the 'R:' at the end of the main sequence. If forgotten, the effect is to have DIAM drop into the first subroutine, and exit directly back to the start of the command list. Similar errors happen if 'R:' is omitted at end of subroutines.

Programming Tips

- a) Write and test a little at a time.
- b) If your text is not appearing, you probably forgot the 'a' in front of it!
- c) For a large job, it can get tedious waiting for the whole sequence to reach the bit you've just coded. So, temporarily use a 'Y;R:' at the beginning, and 'X:' just in front of the piece being tested, or some earlier strategic point. You can easily remove these later.
- d) Experiment with all the fonts, edges, colours, etc. and try deliberate overlap of windows - they can be quite pleasing.

LOAD/SAVE/RUN

Let us start with how DIAM sits in RAM: DIAM machine code sits between RAM:TOP (lowered using a CLEAR XXXXX) and the UDG set. Its entry point is some way beyond the start of the machine code, most of which contains font tables. Entry is made via the familiar USR function. DIAM16K and DIAM48K are of different sizes (due to less fonts in DIAM16K), and at different addresses.

Apart from the RAM lines containing your DIAM commands, we suggest the following to complete your programs with SAVE and auto-load facilities.

```
16K.....
    9990 RANDOMIZE USR 30335: STOP
    9995 SAVE "DIAM" LINE 9996: SAVE "DIAMmc" CODE 26346,
    6253: STOP
    9996 CLEAR 26345: LOAD "" CODE: RUN
```

```
48K.....
    9990 RANDOMIZE USR 63103: STOP
    9995 SAVE "DIAM" LINE 9996: SAVE "DIAMmc" CODE 50392,
    14976: STOP
    9996 CLEAR 50391: LOAD "" CODE: RUN
```

Your cassette tape contains an example 16K program which auto-loads the DIAM16K machine code which follows. The third item on the tape is the DIAM48K machine code, which we leave for you to load if you have the 48K version of Spectrum.

To save a program, you can just SAVE "name". But if you want to save DIAM too and have it auto-run, code Basic as shown above and use GOTO 9995.

16K DIAM leaves you about 2K of RDK space. 48K DIAM leaves you with about 26K of RDK space.

To run your DIAM display, just: RUN.

Editing DIAM Commands

If 'r' key is pressed during DIAM processing, DIAM halts after completing the current command. It resumes when you press any key other than 'r' or 'x'. To exit from processing, press the 'x' key. (No need to hold keys down - DIAM remembers but always finishes its current command.) Upon exit, DIAM will have renumbered your RDKS in tens, and also it sets the program cursor to the interrupted RDK. Only lines below 9990 are renumbered - hence the choice of line numbers in previous paragraph. Resume DIAM with RUN at any time. (But it always starts at the beginning.)

User Basic

If you wish to add your own processing to the display, use a GOTO after the RANDOMIZE USR XXXXX, instead of STOP. Then you can make and print your own fancy invitation letters, etc. If you wish to suppress the renamer function, use an entry point 46 hLShet then that given in the USR function.

DLN: Command Summary

Command	Meaning	Parameters
A	Attribute	B (Bright) D (Dim) P (Plash) S (Steady)
B	Screen Border	0-7 colour code
C	Colour	Paper 0-7, or Paper and Ink 0-7
E	Edge window and if room shrink the window size by 1.	1-9 which is the style of border required.
F	Fill the window	Fill character, or space, or if no fill character then only colours are affected. (no parameters)
P	Pause for 1 second	(no parameters)
R	Return from subroutine	(no parameters)
S	Scroll window contents.	U (Up) D (Down) L (Left) R (Right) If no parameter, U (Up) assumed.
T	Select Type Font	1-9 for type size and style, or N to revert to normal Spectrum font. If no parameter, N is assumed.
Y	Window select or define. (Use just Yr: to select window defined elsewhere in the commands.)	raabccdd or nabccdd or r, where r = non-numeric reference label aa = start screen line bb = number of screen lines cc = start screen column dd = number of screen columns
-	Print text	text to be printed, scrolled up from the lower part of the current window, and using word-processing logic.
<	Print text	text to be rolled right-to-left after initial up-scroll, in lower part of the current window.
↑	Subroutine call	One-character name of subroutine.
•	Subroutine entry point	One-character name of subroutine.
#	Tempo control	No parameters, but repetition factor in no. of TV frames pause between commands.

NOTICE

Program DLN is the copyright of Campbell Systems, and reproduction of the program or this booklet requires our written permission.

All the special computer fonts (except TG:) are created by and are the copyright of IMAGE SYSTEMS of 185 Elm Road, New Malden, KTJ 3HX.