NAME

dl.exe – download files

SYNOPSIS

dl [- h] [-a] [-p PORT] [-c C] [-d DD] [-t TT] [-u UU] [-r] file1 [fileN]

DESCRIPTION

DL is a 'downloader' program used to send data from a PC, via the PC's serial terminal, to some external device. DL can be configured to wait after each transmitted RET character is transmitted until a specific character is received from the remote device, or time-out has expired. These characteristics can be very helpful when downloaded to a microcomputer running a monitor or Basic interpreter.

Options to DL are:

-h	ignores all other options and causes a help message to be printed on the console and then the program terminates.
-a	causes the downloaded file to be prefaced with 5 space characters suitable for autobauding TBAS-52 (and similar) Basic interpreters.
-p PORT	where PORT is of the set 1, 2, 3 or 4. Causes the program to use COMPORT "PORT" as the PC's serial port for downloading combinations. Default is 1.
-s SPEED	where SPEED is of the set 300, 1200, 2400, 4800, 9600, 19200, 38400. Causes the program to set the selected serial port to bit rate SPEED, with 8 bits, one start bit, one stop bit and no parity. On transmit D7 is always set to zero and on receive D7 is ignored (assumed zero). Default is 19200.
-c C	where C is a printable character that DL will wait to receive following sending a RETURN character before DL will send the next line. If the -c option is not used then DL will not wait for any character following transmission of a RETURN character. Instead, it will wait the time specified by -t if non zero. Default is null.
-d DD	where D is a decimal integer value between 33 and 127. The -d option serves the same function as -c, except that using -d provides a means to specify character that have special meaning on the command line (ex., ">"). This option is useful when using DL to down load basic source programs to TBAS-52. For TBAS-52 where the system prompt is ">" use -d 62 as the means to specify ">" as the wait-for character. Default is null.
-r	causes DL to not start looking for the -c specified character until DL has received a RETURN character from the connected device. This is useful when DL is used to download TBAS-52 programs that contain the basic prompt character in a basic statement (e.g., 10 IF A <> B THEN A = 0). Default is not to wait for an echoed RETURN character before accepting the wait-for character.
-t TT	where TT is an integer value between 0 and 32000 and represents the number of system tics (18 tics per second on the PC) while looking for the wait-for character. If TT tics elapse before the wait-for character has been received (assuming it's not null) DL will print an error message at the time of the time-out and again when the program terminates.
-u UU	where UU is an integer value between 0 and 32000 and represents the number of system tics (18 tics per second on the PC) when an END-OF-FILE (EOF) is detected in the source.

EXAMPLES

The two main intended uses for DL are to (1) download BASIC programs to the TUC-52 controller running the TBAS BASIC interpreter and (2) download Intel HEX code programs to the TUC-52 controller running the M51 monitor.

In the case of TBAS the downloader must wait, after sending a line of data to TBAS, until the TBAS interpreter tokenizes the data and outputs a prompt (>) before the downloader sends the next line of data. A batch file to do this is as follows:

```
dl -a -s 19200 -d 62 -t 80 -u 4 tbas.ini
dl -s 19200 -d 62 -t 80 -r -u 4 prog.bas
```

The first line causes DL to send the contents of file "tbas.ini" with the autobaud option enabled at speed 19200 using > as the wait-for-character, with a timeout of 80 tics (4.4 seconds) and an end of file wait time of 4 tics (220 ms). The file "tbas.ini" contains the word "new" on line 1 and no characters on line 2.

The second line causes DL to send the contents of file "prog.bas" with no autobaud, at speed 19200 using > as the wait-for-character but will not see the > until a RET has been received, with a timeout of 80 tics (4.4 seconds) and an end of file wait time of 4 tic (220 ms). The file "prog.bas" contains one or more TBAS statements which begin with line numbers.

In the case of M51 the downloader sends at full speed, without waiting for any response from TUC-52 until the end of file has been transmitted, and then the end of file response from M51 is simply copied to the console display. A batch file to do this is as follows:

```
dl -s 19200 -t 9 -u 9 iload.ini
dl -s 19200 -u 9 load.hex
```

The first line causes DL to send the contents of file "iload.ini" with no autobaud at speed 19200 waiting after each line for 9 tics (500ms) and an end of file wait time of 9 tic (500 ms) after the last character of any response has been received from the downloaded device. The file "iload.ini" contains the word groups "r 0" on line 1 and no characters on line 2. This causes M51 to enter the read intel hex mode with an offset address of zero.

The second line causes DL to send the contents of file "load.hex" with no autobaud at speed 19200 with no wait time after each line transmitted and an end of file wait time of 9 tic (500 ms) after the last character of any response has been received from the downloaded device. The file "load.hex" contains one or more lines of intel hex code.