



Technical Data and Instructions

Standard DataPlot Commands

This sheet describes the command characters and formats that are available for use by the standard DataPlot series 1XXX and 2XXX series printers. Printer control boards all implement these fonts in their standard programs; the only differences relate to the differing number of printhead dots in the different models. Special application programs may alter the command set substantially. If you printer contains such a custom program, refer to its special documentation.

X-PRINTING: This is the most commonly used printing, with character lines printed across the paper width in the currently selected font, size, boldness and orientation. As with other printers, just send printable character codes to the interface. These characters are stored in an x-line buffer that is printed when a CR (0x0D) is received. The line may also be overlaid with following graphics or paper advance commands by omitting the CR. A number of characters that may be entered into the buffer dependent on the size of the printhead, the font and size selections. The following table shows the maximum number of size 1 characters per line for each of the available print head sizes:

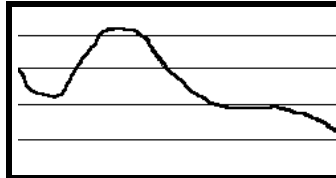
	224	320	416	448	832	1728
5x7	37	53	69	72	138	284
5x9	37	53	69	72	138	284
5x9d	37	53	69	72	138	284
7x9	28	40	52	54	104	213
11x21d	17	13	32	33	64	131

A TAB function is also available for use in x-printing.

Y-PRINTING: DataPlot printers enable character printing in the y-direction, i.e., with lines of print extending along the length of the paper. The YLOAD and YPRINT commands used for this purpose are described in the Command List. Each separate message in this y-line buffer requires N+6 bytes, where N is the number of bytes in the actual stored message. The table below indicates the size of the y-line buffer for each printhead size. As with stored xlines, stored y-lines may be printed separately or overlaid with plotting actions. Y-direction printing is often used for labels on graph axes or other notations on graphic output. It may also be used for printing out pure text, however, such as for printing a table that requires more characters per line than are available in the x-direction. Any of the fonts, sizes, boldness and orientations of x-printing may also be selected for y-printing.

Dots	224	320	416	448	832	1728
y-line	1620	1088	556	391	4500	5833

PLOTTING: DataPlot printers have a plot buffer data storage that contains the same number of bits as printing dots in the printhead: 224, 320, 416, 448, 832 and 1728 dots in the respective models 224, 320, 416, 448, 832 and 1728. Individual bits or lines of adjacent bits can be set in the plot buffer using the DOTLOAD and LINELOAD commands. Any pattern thus loaded into the plot buffer can then be printed on the paper using the PLOT or NPLLOT commands, or saved for repeated use by the GRIDLOAD command. The Command List describes these and other commands in detail.



The dot pitch for plotting is 100 dots/inch across the paper width for the models containing 224, 320 and 416 dot printheads and 200 dots/inch for the 448, 832 and 1728 dot printheads. All models have 200 dots/inch along the paper length. The standard DataPlot commands allow printing of any 2 dimensional pattern that can be represented in this matrix. The maximum plotting speed is about 125 dot rows per second, somewhat slower for rows containing more than 64 set dots.

Any stored xprint and y-print lines will automatically be printed as plotting proceeds. This feature permits printed characters to be placed amid graphics, using any desired font, size, boldness and orientation.

PAPER MOTION: For maximum flexibility in moving paper forward or backward, the standard DataPlot program includes ADV and REV commands, each with resolution of one microstep (about 0.005"). These actions are described in detail in the Command List. If x-print and/or y-print lines are stored in their respective buffers, the ADV command will print the stored characters as the paper advances.

COMMAND hex, decimal keyboard	Description (All argument digits are in ASCII)	COMMAN D hex, decimal keyboard	Description (All argument digits are in ASCII)
CROSS 02, 02 cntrl B	Prints a small cross starting on the current plot row. The x-position of the center of the cross is specified in dots by the next 3 digits. The valid range for the x-position is 0 thru N-1 where N is the number of dots in the printhead. The cross pattern will automatically be merged into the normal print and plot data as the paper advances.	BITLOAD 12, 18 cntrl R	Loads a pattern of dots into the plot buffer, using the bit pattern in a succession of input bytes. The first byte becomes the first 8 bits in the plot buffer, etc. This is a fast way to load a pattern that is stored externally, for use as a logo, etc. The number of bytes required is the number of printhead dots divided by 8.
TEE 03, 03 cntrl C	Prints a small 'T' starting on the current plot row. The x-position of the center of the cross is specified in dots by the next 3 digits. The valid range for the x-position is 0 thru N-1 where N is the number of dots in the printhead. The cross pattern will automatically be merged into the normal print and plot data as the paper advances.	REV 13, 19 cntrl S	Reverses the paper the number of steps given by the next 3 digits, up to 999. To eliminate backlash effects, the paper is stepped past the desired end point, then brought forward.
CLEARX 07, 07 cntrl G	The x-line is buffer cleared.	GRIDLOAD 14, 20 cntrl T	Stores the plot buffer pattern into one of ten grid buffers for later repeated use. The next digit sent (0-9) determines which buffer.
FONTSWAP 08, 08 cntrl H	Switches between the 5x9 and 11x21 fonts. This command exists to maintain backward compatability. Note that font changes are allowed only between x or y-lines.	GRID 15, 21 cntrl U	Places the pattern stored in a grid buffer into the plot buffer as a logical OR with the current plot buffer contents. The next digit sent determines which grid buffer will be used. This is a way to merge grids with data patterns for plotting.
YPRINT 0A, 10 cntrl J	Prints all currently stored y-lines, merging the stored x-line, if any, and clearing both buffers.	PLOT 16, 22 cntrl V	Plots the dot pattern stored in the plot buffer, advances the paper one step and clears the plot buffer. Any stored x and y-lines will be printed as the paper advances.
ADV 0B, 11 cntrl K	Advances the paper the number of steps determined by the next 3 digits, up to 999. Each step is approximately 0.005" so that you would send ADV followed by 200 to move one inch. As paper advances, any stored x and y-lines will be printed.	NPLOT 17, 23 cntrl W	Plots the pattern stored in the plot buffer repeatedly, the number of times determined by the following 3 digits, and clears the plot buffer. Stored x and y-lines will be printed as the paper advances.
PRINT 0D, 13 CR	Causes x-buffer characters to be printed across the paper. If you omit the PRINT command, the characters will be printed as the paper advances for succeeding operations, such as PLOT or ADV.	GET 18, 24 cntrl X	Applies only to printers with one or more analog input channels. In such printers, it causes analog input N to be sampled, where N may be from 1 thru 8, as determined by the next digit sent. The 12 bit ADC value is sent out to the host through the serial interface.
YLOAD 0E, 14 cntrl N	Begins a message to be sotred in the y-buffer for subsequent printing in the y-direction. The 3 digits (4 digits for 1728 dot printhead) immediately following YLOAD determine which dot number contains the center of the y-line. Next follows the literal message in ASCII, concluded by a CR (0x0D). You may enter several y-lines, as long as the y-buffer size is not exceeded. The actual printing does not occur with the YLOAD command, but with YPRINT or as paper advances for another operation. Y-printing is normally oriented to be upright when the graphics is upright but can be inverted. (see ESCAPE)	TAB 19, 25 cntrl Y	Tabs to the position determined by the following 2 digits (3 digits for 1728 dot printhead). If the argument is smaller than current position on the line, TAB is ignored. TAB works only for x-printing.
CLEARY 0F, 15 cntrl O	The y-line is buffer cleared.	ESCAPE 1B, 25 cntrl [Starts an escape sequence. Refer to table 2, 'Escape Code Selections' for details.

CLEAR 10, 16 cntrl P	Clears both x and y-buffers.	DOTLOAD 1C, 28 cntrl L	Sets a single dot in the plot buffer, using the next 3 digits (4 digits for 1728 dot printhead) to identify which dot. Dot number 0 is at the end of the printhead nearest the stepper motor. The opposite end dot is number is N-1 where N is the total number of dots in the printhead. DOTLOAD produces the logical OR of its argument with the current plot buffer contents. Thus, successive DOTLOAD commands can be used to plot several dots in the same row.
LINELOAD 11, 17 cntrl Q	Loads a line of dots into the plot buffer, using the following 6 digits (8 digits for 1728 dot printhead) to identify the dot numbers where the line begins and ends. LINELOAD is a logical OR with the current plot buffer contents.		
COMMAND hex, decimal keyboard	Description (All argument digits are in ASCII)		
PLOTAREA 1D, 29 cntrl]	<p>This command defines the characteristics of a range of printhead dot positions that will be used for subsequent plotting. The complete format for this command is: \$c llluuufojgrrrr...t where \$ represents the PLOTAREA command code, c is the plotting region number (1, 2, 3 or 4), lll and uuu are the 3 digit dot numbers that define the lower and upper plotting limits, fo is a two digit code that controls the appearance of the plotted line, j is a single digit that determines if the plotted points are to be individual dots (j=0) or joined together (j=1), g is a grid identifier character (0..9 or A..E), rrr is the number of plot rows between repeats of this grid (up to 10 such repeating grids may be specified, though none is required) and t is the terminating character (CR or 0x0D). Once the plot area has been defined, the commands PLOTADOT and VECTOR may be used and associated with the region through the channel number.</p> <p>The parameters lll and uuu specify the lower and upper dot position for the plot region. These values must be within the range 0..N-1 where N is the number of dots in the printhead. In addition, the relative size of lll and uuu is used to control the location of the origin of the plot region. If uuu >= lll, the origin of the plot region will be at dot lll, that is, when PLOTADOT or VECTOR is used, an argument of 0 for either of these commands will represent a dot at printhead dot position lll. However, if uuu < lll, the origin of the plot region will be at printhead dot position uuu and increasing argument values will produce dots closer to the lll end of the region.</p> <p>The pair of digits 'fo' are used to control the appearance of the line that is plotted. The 'f' digit control the number of consecutive dots that are 'on' while the 'o' digit specifies the number of open dots between line segments. For example, if the parameters are specified as '10', a solid line will be plotted but if the specification '23' is given, the plotted line will consist of 2 'filled' dots separated by 3 'open' dots, resulting in a dotted line. Magnified many times, these two lines specifications would produce plotted lines that resemble figures 1 and 2. By manipulating the values of the 'fo' parameter, various types of dotted lines can be generated, providing the means of identifying various plotted lines where plot regions overlap or are coincident.</p> <p>If the grid identifier, 'g', is a numeric digit, then the grid used is one of the ten free form grids set up by the GRIDLOAD command. If it is a letter (A..E), then a standard grid is generated, adjusted to fit the specified plotting region. See table 1 for the standard grid types.</p> <p>Figure 3 shows an example of how the PLOTAREA command could be used.</p>		
PLOTADOT 1E, 30 cntrl ^	<p>Causes a point to be added to the specified plotting region. The format for the command is \$cddd where \$ represents the PLOTADOT command code, c is the plotting region number (1..4) and ddd is the data value. The data value maps directly onto the printer dots, with the value 000 mapping to lll (as defined by the PLOTAREA command) and increasing data values stepping toward uuu. Thus, if lll = 100 and uuu = 200, a data value of 025 would map onto dot 125. If however, the plotting direction is reversed (lll = 200, uuu = 100), a data value of 025 would map onto dot 175. Dots that would lie outside the range between lll and uuu will map onto the nearer of lll or uuu. If j=1 (see PLOTAREA command), then the selected line type is filled in between this and the dot corresponding to the immediately preceeding data value in the region. For the first data value entered into a region (e.g. the first PLOTADOT command for a region after the regions defining PLOTAREA command), only the dot that corresponds to the data value is set.</p> <p>PLOTADOT produces the logical OR of it's output with the previous contents of the plot buffer. It does not cause the paper to be marked or moved. It can be used repeatedly to place into the plot buffer, data for all four channels. As before, the programmer would use the PLOT or NLOT commands to actually print onto the paper.</p>		

Figure 5 demonstrates the use of the PLOTADOT command.



Figure 1, The line generated by 'fo' = 10 (a continuous line).



Figure 2, The line generated by 'fo' = 23 (the repeat length is marked).

Grid code	Generated grid pattern
A	Every dot from lll through uuu.
B	Only dots lll and uuu.
C	Dots lll, uuu and one dot at the 50% position.
D	Dots lll, uuu and dots at the 25%, 50% and 75% positions.
E	Dots lll, uuu and 9 other dots at the 10% positions.

Table 1, Standard grid patterns.

ESC "A"	selects the 11x15d font.
ESC "B"	selects the 5x9 font.
ESC "C"	selects the 5x7d font.
ESC "D"	turns off bold printing.
ESC "E"	turns on bold printing.
ESC "F"	selects the 5x7 font.
ESC "G"	selects the 7x9 font.
ESC "H"	selects size 1 characters.
ESC "I"	selects size 2 characters.
ESC "J"	selects size 3 characters.
ESC "K"	selects size 4 characters.
ESC "L"	selects normal y-printing.
ESC "M"	selects inverted y-printing.
ESC "N"	selects normal x-printing.
aESC "O"	selects inverted x-printing.

Table 2, Escape code selections

VECTOR 1F, 31 cntrl _	<p>VECTOR is used to produce, with a single command, data vector line segments in one or more of the four plotting regions using the specified line type(s). The command format is \$ssscbbbeecbbbee...t where \$ represents the VECTOR command code, sss defines the number of paper advance steps between the beginning and ending points of the vector and the sequence 'cbbbee' gives the plotting region and the beginning and ending data values for the vector. Up to four vector segments can be specified, one for each channel. 't' is the terminating character for the command (CR or 0x0D).</p> <p>When the terminating character is received, the printer calculates the needed dots for the first row, plots that row, then continues with the second row, etc. until all sss rows are plotted and the vector segments appear on the paper, with whatever grids were previously selected for the region(s). As with the other commands that actually move and mark the paper, any stored x and y-line data will be printed as the paper advances.</p> <p>Refer to figure 7 for an example of the VECTOR command.</p>
-----------------------------	---

This is an example of the PLOTAREA command used to define a plot area that is designated region 1. Up to 4 such regions may be defined and each region's definition is completely independent of the other regions. Regions may overlap or use different portions of the paper as required. The actual text used to create the example plot shown below is given here. These text lines are transmitted to the printer by a program that translates certain character sequences into control characters. For example, the sequence '\[' is converted by the program into the single control character ESC, the escape control code. Each appearance of the backslash character '\ ' tells the transmitting program to convert the character that follows immediately into its equivalent 'control' character code. The text between sets of curly braces '{' and '}' is treated as a comment by the transmitting program and not sent to the printer. The control character sequence 'CR' 'LF' (0x0D 0x0A) is sent to the printer after each line of text is transmitted.

```
\[N\[E\[I\[X050{ Normal X printing, Size 1 characters, Eject = 50 }
GRID A TEST (50 REPEAT){ Display a text annotation }
\]1000200121A050{ Plot region 1 from dot 000 through dot 200, 1 filled, 2 open dots when
plotting, join plotted dots, use grid A with a 50 dot repeat }
\[K100\]1000223100{ Plot 100 rows & turn off plot area }
GRID B TEST (10 REPEAT)
\]1000200121B010
\[K100\]1000223100
GRID C TEST (20 REPEAT)
\]1000200121C020
\[K100\]1000223100
GRID D TEST (30 REPEAT)
\]1000200121D030
\[K100\]1000223100
GRID E TEST (40 REPEAT)
\]1000200121E040
\[K100\]1000223100{ Do final advance and turn off the plot area }
```

Figure 4 shows a magnified view of the plot produced by the above command sequence.

Figure 3, An example of PLOTAREA grid commands.

GRID A TEST (50 REPEAT)

GRID B TEST (10 REPEAT)

GRID C TEST (20 REPEAT)

GRID D TEST (30 REPEAT)

GRID E TEST (40 REPEAT)

Figure 4, A magnified image of the plot produced by the data in figure 3.

This is an example of the PLOTAREA command used to define plot areas designated regions 1, 2, 3 and 4. The actual text used to create the example plot shown below is given here. These text lines are transmitted to the printer by a program that translates certain character sequences into control characters. For example, the sequence '\[' is converted by the program into the single control character ESC, the escape control code. Each appearance of the backslash character '\' tells the transmitting program to convert the character that follows immediately into its equivalent 'control' character code. The text between sets of curly braces '{' and '}' is treated as a comment by the transmitting program and not sent to the printer. The control character sequence 'CR' 'LF' (0x0D 0x0A) is sent to the printer after each line of text is transmitted.

```
PLOTDOT COMMAND TEST
\]1000100101B002{ Region 1, dots 000 through 100, 1 filled, 0 open, join, grid B with a
repeat of 2 }
\^1000\V^1010\V^1020\V^1030\V^1040\V^1050\V^1060\V^1070\V^1080\V^1090\V
\^1050\V^1051\V^1052\V^1053\V^1054\V^1055\V^1056\V^1057\V^1058\V^1059\V^1060\V^
1061\V^1062\V^1063\V^1064\V^1065\V^1066\V^1067\V^1068\V^1069\V
\]2110210111B002{ Region 2, dots 110 through 210, 1 filled, 1 open, join, grid B with a
repeat of 2 }
\^2000\V^2010\V^2020\V^2030\V^2040\V^2050\V^2060\V^2070\V^2080\V^2090\V
\^2050\V^2051\V^2052\V^2053\V^2054\V^2055\V^2056\V^2057\V^2058\V^2059\V^2060\V^
2061\V^2062\V^2063\V^2064\V^2065\V^2066\V^2067\V^2068\V^2069\V
\]3000100211B002{ Region 3, dots 000 through 100, 2 filled, 1 open, join, grid B with a
repeat of 2 }
\^3000\V^3010\V^3020\V^3030\V^3040\V^3050\V^3060\V^3070\V^3080\V^3090\V
\^3050\V^3051\V^3052\V^3053\V^3054\V^3055\V^3056\V^3057\V^3058\V^3059\V^3060\V^
3061\V^3062\V^3063\V^3064\V^3065\V^3066\V^3067\V^3068\V^3069\V
\]4110210221B002{ Region 4, dots 110 through 210, 2 filled, 2 open, join, grid B with a
repeat of 2 }
\^4000\V^4010\V^4020\V^4030\V^4040\V^4050\V^4060\V^4070\V^4080\V^4090\V
\^4050\V^4051\V^4052\V^4053\V^4054\V^4055\V^4056\V^4057\V^4058\V^4059\V^4060\V^
4061\V^4062\V^4063\V^4064\V^4065\V^4066\V^4067\V^4068\V^4069\V
\]1000223100
\]2000223100
\]3000223100
\]4000223100{ Turn off the plot areas }
```

Figure 6 shows a magnified view of the plot produced by the above command sequence.

Figure 5, an example of the PLOTDOT command.

PLOT00T COMMAND TEST

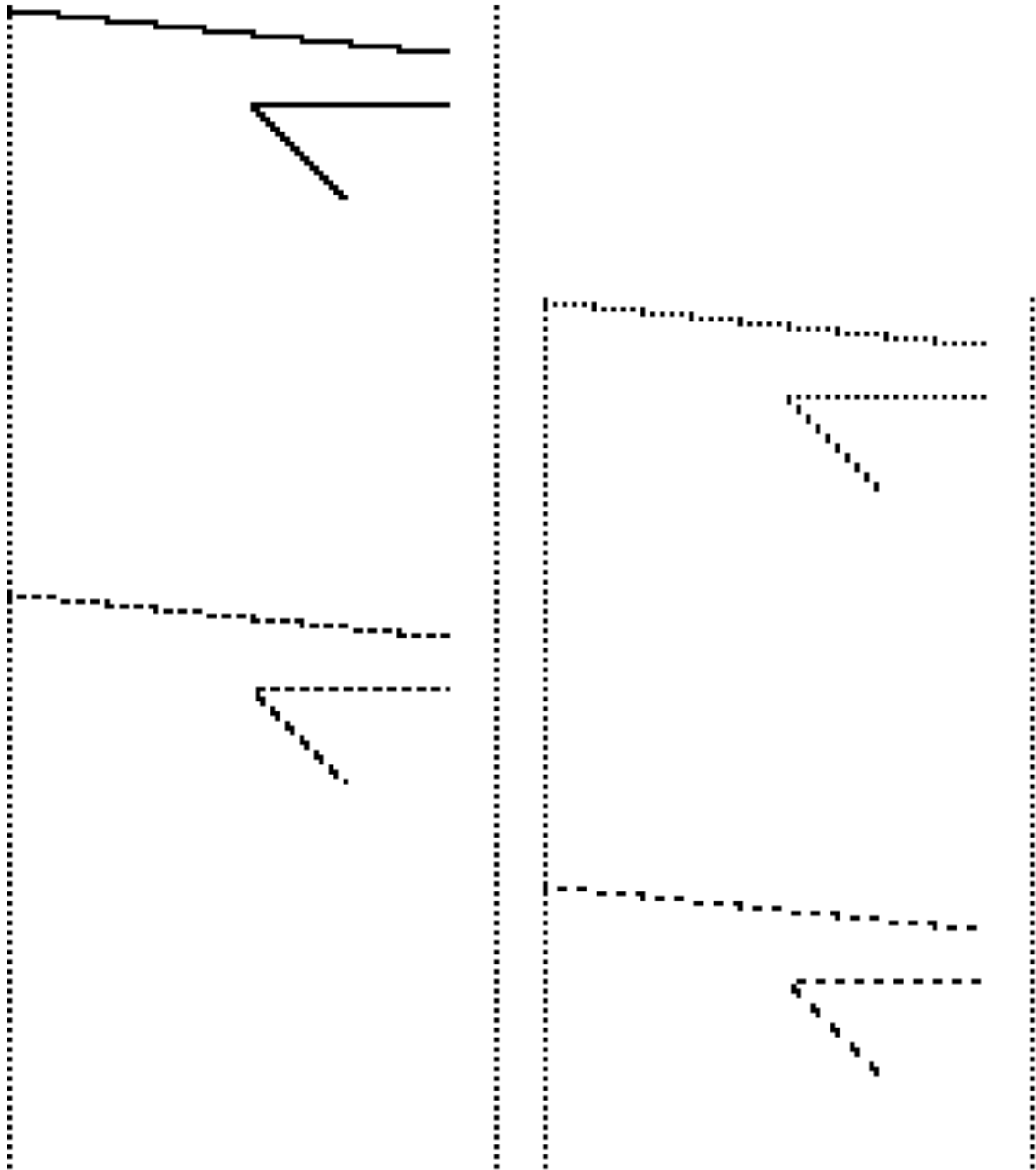


Figure 6, A magnified image of the plot produced by the data in figure 5.

This is an example of the PLOTAREA command used to define plot areas designated regions 1, 2, 3 and 4. The actual text used to create the example plot shown below is given here. These text lines are transmitted to the printer by a program that translates certain character sequences into control characters. For example, the sequence '\[' is converted by the program into the single control character ESC, the escape control code. Each appearance of the backslash character '\' tells the transmitting program to convert the character that follows immediately into its equivalent 'control' character code. The text between sets of curly braces '{' and '}' is treated as a comment by the transmitting program and not sent to the printer. The control character sequence 'CR' 'LF' (0x0D 0x0A) is sent to the printer after each line of text is transmitted.

```
\[N\[E\[I\[X050{ Normal X printing, Enhanced, Size 1 characters, Eject = 50 }
VECTOR COMMAND TEST
\[1000100101B002{ Region 1, dots 000 through 100, 1 filled, 0 open, join, grid B with a
repeat of 2 }
\_0101010090
\[2110210111B002{ Region 2, dots 110 through 210, 1 filled, 1 open, join, grid B with a
repeat of 2 }
\_0202010090
\[3000100211B002{ Region 3, dots 000 through 100, 2 filled, 1 open, join, grid B with a
repeat of 2 }
\_0303010090
\[4110210221B002{ Region 4, dots 110 through 210, 2 filled, 2 open, join, grid B with a
repeat of 2 }
\_0404010090
\[1000223100
\[2000223100
\[3000223100
\[4000223100{ Turn off the plot areas }
END OF DATA
```

Figure 8 shows a magnified view of the plot produced by the above command sequence.

Figure 7, and example of the VECTOR command.

VECTOR COMMAND TEST

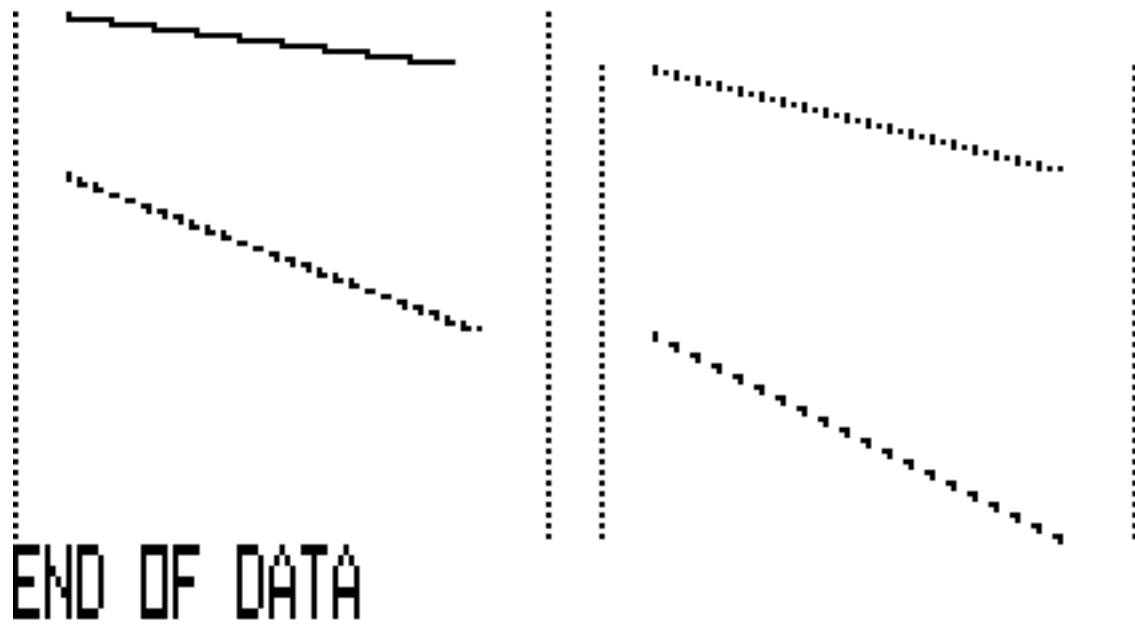


Figure 8, A magnified image of the plot produced by the data in figure 7.

B-G Instruments, Inc
13607 E. Trent Ave.
Spokane, WA 99216 U.S.A.
www.bginstruments.com

Toll Free: (888) 244-4004
Phone: (509) 893-9881
Fax: (509) 893-9803
Email: contact@bginstruments.com