General Plotting Functions

**CLIPUU**
Specifies plot bounds in user units, regardless of plotting mode. Interprets stack values as:

<table>
<thead>
<tr>
<th>T</th>
<th>x-minimum</th>
</tr>
</thead>
<tbody>
<tr>
<td>Z</td>
<td>x-maximum</td>
</tr>
<tr>
<td>Y</td>
<td>y-minimum</td>
</tr>
<tr>
<td>X</td>
<td>y-maximum</td>
</tr>
</tbody>
</table>

**CSIZE**
Sets the character space height in GU's.

**CSIZEO**
Sets the character space height in GU's (X-register), aspect ratio (Y-register), and slant—in current angular mode—(Z-register).

**DGTIZE**
Enteres the x- and y-coordinates of the pen's position into the X- and Y-registers when the plotter's ENTER key is pressed.

**DRAW**
Draws a line from last pen position specified by plotter module to position specified by \((x,y)\).

\[
\begin{align*}
Y & \quad y \\
X & \quad x
\end{align*}
\]

Current Units (UU's or GU's)

**FRAME**
Draws a box around the active plotting boundaries (graphic limits or plotting bounds).

**GCLEAR**
Advances page on plotters that have a page feed mechanism.

**IDRAW**
Draws a line from the last pen position specified by a plotter module function to a point \(x\) and \(y\) units from that position.
**MOVE**

Moves the pen from the position specified by the last plotter module function to a point $x$ and $y$ units from that position.

**LABEL**

Prints the contents of the ALPHA register at the last pen position specified by the HP-41.

**LDIR**

Sets the angle of rotation from $0^\circ$ for printing labels. A positive value specifies counterclockwise rotation.

**LIMIT**

Sets graphic limits in millimeters, according to values in stack. Interprets values in stack in same way as shown for [CLIPUU]. Also sets same default conditions as are set by [PINIT].

**LOCATD**

Specifies plotting bounds by digitizing the two opposite corners. When ENTER POINT is displayed, move pen to lower-left point and press plotter's ENTER key. When ENTER POINT re-appears, move pen to upper-right point and press ENTER key again.

**LOCATE**

Specifies plot bounds in graphic units, regardless of current plotting mode. Interprets values in stack in same way as shown for [CLIPUU].

**LORG**

Sets the label origin position (1 through 9).

**LTYPE**

Uses a line type number (integer value 1 through 8) to select one of eight line types.

**LTYPEO**

Operates same as [LTYPE] except that the number in the Y-register specifies the length of the repeat pattern as a percentage of the length of the diagonal between P1 and P2.

**LXAXIS**

Draws $x$-axis between specified minimum and maximum $x$-values and at the specified $y$-intercept. Labels tic marks below the current plot bounds. A negative tic-spacing parameter specifies horizontal labels.

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>T</td>
<td>$x$-maximum</td>
</tr>
<tr>
<td>Z</td>
<td>$x$-minimum</td>
</tr>
<tr>
<td>Y</td>
<td>tic spacing</td>
</tr>
<tr>
<td>X</td>
<td>$y$-intercept</td>
</tr>
</tbody>
</table>
LYAXIS Draws y-axis between specified minimum and maximum y-values and at the specified x-intercept. Labels tic marks to the left of the current plot bounds. Labels are always printed horizontally.

<table>
<thead>
<tr>
<th>T</th>
<th>y-maximum</th>
</tr>
</thead>
<tbody>
<tr>
<td>Z</td>
<td>y-minimum</td>
</tr>
<tr>
<td>Y</td>
<td>tic spacing</td>
</tr>
<tr>
<td>X</td>
<td>x-intercept</td>
</tr>
</tbody>
</table>

MOVE Moves the pen to specified \((x,y)\) coordinate without drawing a line. Interprets stack in same way as shown for DRAW.

PCLBUF Clears I/O buffer created by PINIT and returns 26 memory registers to available program memory.

PDIV Rotates the \(x\)- and \(y\)-axes to the angle specified in the X-register. Used only to rotate plot direction for incremental plotting ([IMOVE], [IDRAW], and [IPLOT]) and relative plotting ([RPLT]). Does not affect pen movement specified in absolute coordinates. PDIV angles are in reference to the original default PDIV angle (0°).

PEN Selects pen specified by integer portion of a value in X-register.

PENDN Lowers pen to plotting surface.

PENUP Lifts pen from plotting surface.

PINI When I/O buffer does not exist, uses 26 memory registers to create it. Also performs the following:
  
  - Sets plot bounds equal to graphic limits currently maintained by plotter.
  - Sets UU's equal to GU's and sets to UU mode.
  - Selects pen 1; sets line type 1.
  - Sets 3 GU character space height.
  - Sets label origin to 1 and label direction to 0°.
- Sets x- and y-axes tic lengths to default values for your plotter.
- Sets plotting rotation to 0°.
- If I/O buffer did not exist before executing \texttt{PINIT}, sets bar code plotting parameters to default values. Otherwise does not affect current bar code plotting parameters.

\texttt{PRCL} \hspace{1cm} \text{Recalls contents of the buffer register specified by a number in the X-register.}

\texttt{RATIO} \hspace{1cm} \text{Calculates for the current graphic limits the ratio of the x-axis length to the y-axis length.}

\texttt{SCALE} \hspace{1cm} \text{Sets user scale for plot bounds and sets plotter module to UU mode. Interprets values in stack in same way as shown for \texttt{CLIPUU}.}

\texttt{SETGU} \hspace{1cm} \text{Switches plotter module to Graphic Units (GU’s) mode.}

\texttt{SETUU} \hspace{1cm} \text{Switches plotter module to User Units (UU’s) mode.}

\texttt{TICLEN} \hspace{1cm} \text{Sets vertical and horizontal tic lengths to specified percentage of corresponding dimensions of graphic limits (GU’s only).}

\texttt{UNCLIP} \hspace{1cm} \text{Resets the plot bounds to the graphic limits without changing the current user scale.}

\texttt{WHERE} \hspace{1cm} \text{Enters in the X- and Y-registers the corresponding coordinates (in current units) of the point specified by the last HP-41 pen movement function. Places current pen status in Z-register.}

\texttt{XAXIS} \hspace{1cm} \text{Draws a horizontal axis at the specified y-intercept (UU’s or GU’s).}

\texttt{XAXISO} \hspace{1cm} \text{Draws x-axis between the specified minimum and maximum x-values and at the specified y-intercept, with tic marks at specified intervals. Interprets parameters in current units and in same way as shown under \texttt{LXAXIS}.}

\texttt{YAXIS} \hspace{1cm} \text{Draws a vertical axis at the specified x-intercept for current plotting mode.}

\texttt{YAXISO} \hspace{1cm} \text{Draws a y-axis between minimum and maximum y-values at the specified x-intercept, with}
tic marks at specified intervals. Interprets parameters in current units and in same way as shown under LXAXIS.

**Plot-Option Functions**

The following four functions: (1) Use values in the X- and Y-registers to define next pen position—in current units (UU's or GU's); (2) move pen from the last position specified by a plotter module function to new position using existing pen status—“up” or “down”; (3) set pen status to “down” when pen reaches new point—except as provided by PLREGX.

**PLREGX** Moves pen or draws line to each point specified by the coordinates in the series of data storage registers R_ii through R_fff, as specified by the number iiii.fff in the X-register. R_ii specifies x_0; R_ii+1 specifies y_0, and so on. If PLREGX finds an Alpha character in either data register of any pair included in R_ii through R_fff, the pen status is switched to “up”. It remains so until a subsequent pair of data registers containing only numeric data is encountered.

**Utility Plotting Program**

**NEWPLOT** Prepares for plotting by prompting for six parameters, automatically assigning six default parameters, and transferring execution to REPLOT. Uses R_00 through R_11—which form the plotting data base.

**REPLOT** Prompts you to either automatically generate a complete plot (by pressing R/S) or edit contents of any register nn (by pressing nn R/S).
PLINIT Initializes your plotter according to parameters in plotting data base.

PLTUXY Plots a function or data specified by parameters in plotting data base.

PLANOT Frames plotting area then annotates plot according to parameters in plotting data base.

Y? Prompts you to key in the next y-coordinate.

X? Prompts you to key in the next x-coordinate.

Utility Plotting Program User Parameters

<table>
<thead>
<tr>
<th>Storage</th>
<th>Parameter</th>
<th>Format</th>
</tr>
</thead>
<tbody>
<tr>
<td>R08</td>
<td>NAME?</td>
<td>label or bbb.eeet</td>
</tr>
<tr>
<td>R00</td>
<td>XMIN</td>
<td>n</td>
</tr>
<tr>
<td>R01</td>
<td>XMAX</td>
<td>n</td>
</tr>
<tr>
<td>R05</td>
<td>XINC</td>
<td>(-n, \text{int or label})</td>
</tr>
<tr>
<td>R04</td>
<td>YMIN</td>
<td>n</td>
</tr>
<tr>
<td>R07</td>
<td>YMAX</td>
<td>n</td>
</tr>
<tr>
<td>R02</td>
<td>PLTPRM</td>
<td>ccclp.bbbt or label</td>
</tr>
<tr>
<td>R03</td>
<td>ANNOT</td>
<td>(\pm F _XX_xx.F _YY_yy)</td>
</tr>
<tr>
<td>R06</td>
<td>XAXAT</td>
<td>n</td>
</tr>
<tr>
<td>R09</td>
<td>YAXAT</td>
<td>n</td>
</tr>
</tbody>
</table>

Bar Code Functions

BC Uses bit pattern in ALPHA register to plot a single row of HP-41 bar code.

BCA Converts data in ALPHA register to an Alpha-replace bar code bit pattern. Places value in X-register to indicate number of bytes in bit pattern.

BCAA Operates in same way as BC except that bit pattern is for Alpha-Append bar code.

BCCKSM Computes checksum of bit pattern in ALPHA register and places checksum in byte specified by bb (in X-register), counting from the right. For HP-41 bar code:

One-Byte Paper Keyboard: \(bb = 1\)
Two-Byte Paper Keyboard: \(bb = 2\)
Direct Execution: \(bb = \text{no. of bytes in ALPHA} + 1\)
Plotter Option. Operates the same as [BC] except that (1) HP-41 directional bars are not automatically added to row; (2) you specify how many rightmost bits to plot from first and last bytes; (3) enables you to interpret bit patterns in various ways, depending upon current bar code type. (Refer to [Bysize].) Positive number in X-register specifies plotter option.

<table>
<thead>
<tr>
<th>Z</th>
<th>number of bits (Leftmost Byte)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Y</td>
<td>number of bits (Rightmost Byte)</td>
</tr>
<tr>
<td>X</td>
<td>bb Bytes in Row (Includes Leading and Trailing Bytes)</td>
</tr>
</tbody>
</table>

Printer Option. Prints on HP 82162A Thermal Printer a row of HP-41 bar code corresponding to bit pattern in ALPHA register. Negative number (−bb) in X-register specifies printer option; |−bb| specifies number of bytes in row. To use after [BCA], [BCAA], [BCX], or [BCXS], just press [CHS] and execute [BCO]. To use after press RCL[J]TCHS[BCO].

Uses program name in Y-register and row number/bit pattern (rrr.bb) in X-register to place in ALPHA register a bit pattern for specified program row. Negative rrr.bb specifies “private” bar code. Places following values in stack:

<table>
<thead>
<tr>
<th>T</th>
<th>bb Number of Bytes in Row</th>
</tr>
</thead>
<tbody>
<tr>
<td>Z</td>
<td>fff.iii Lines Coded in Row</td>
</tr>
<tr>
<td>Y</td>
<td>name Program Name</td>
</tr>
<tr>
<td>X</td>
<td>±rrr.bb Next Row Number and Length</td>
</tr>
</tbody>
</table>

BCREGX Transforms the decimal values in a block of data registers into a bit pattern in ALPHA register. Each register corresponds to one byte. Uses iii.fff in X-register to specify first and last data registers. RIII corresponds to leftmost byte in bit pattern. Does not compute checksum.
Uses absolute values of numbers in X- and Y-registers to adjust bar/space proportions (in APU’s) and to specify bar code type (0 through 3).

\[
\begin{array}{cccc}
Y: & nn & . & t & ww & ss \\
Narrow Bar Width & Bar Code Type & Wide Bar Width & Space Width \\
(0 = HP-41)
\end{array}
\]

\[
\begin{array}{cccc}
X: & pp & . & hhh & aa \\
Pen Width & Bar Height & Alternate Space Width (Non HP-41 Bar Code Only)
\end{array}
\]

Using zeros in place of any parameter element sets that element to its default value. Default values are:

\[
\begin{array}{ccc}
pp & hhh & aa \\
12 & 350 & 00
\end{array} \quad \begin{array}{ccc}
nn & . & t & ww & ss \\
18 & . & 0 & 30 & 21
\end{array}
\]

Places in ALPHA register a bit pattern for non-sequenced bar code representing data in X-register. Places value in X-register to indicate number of bytes in bit pattern.

Places in ALPHA register a bit pattern for sequenced bar code representing data in X-register and sequence number in Y-register. Places value in X-register to indicate number of bytes in bit pattern.