CHAR PLOT
Donald Sordillo

CHAR PLOT is a routine which enables one to use the Calcomp plotter as an output typewriter.
CHAR PLOT

This program is stored as CHPLOT BIN [English CHAR PLOT]. In use a code, representing a character or command as defined in Appendix I, is placed into accumulator C. Upon calling the routine the plotter will, either print a character, or set itself into one of several modes.

CHARACTER MODE

The input to the routine is a word whose 8 low order bits contain a code and whose sign bit must be 0. The routine is entered by

MOVE C, [WORD]
PUSHJ P, PLOT.

A word = 0, stops everything and initiates the system.

NOTE: The program starts off in lower case mode. While it is in this mode any attempt to issue a lower-case code causes the computer to hang up.

It is suggested that the first call be used to set the routine to upper case and the 8th bit in the code used to shift between upper and lower cases.

The symbols P, C, and CRKCHN are global and user-defined. Other symbols are:

PLOT C Normal entry point.
UCTAB Beginning of upper case table.
LCTAB Beginning of lower case table.
CLNGTH Routine which returns length of the character which was its argument in Acc. C.

\[
x \}
\text{Current position of pen. At initialization } x = y = 0.
\]
LBUFF

LWBUFF

Each 1000g words. Can be reduced to one word to gain more storage for other routines.

CRKBRK

Location for return from interrupt.

CONTROL MODE

The input to the routine is a word whose sign bit is a one. It is entered by:

MOVE C, [WORD]

PUSHJ P, PLOTC.

The high-order bits of the word are decoded as follows:

\[2^5\]

<table>
<thead>
<tr>
<th>10</th>
<th>xy Format mode</th>
</tr>
</thead>
<tbody>
<tr>
<td>110</td>
<td>Define CHAR/CHAR SET mode</td>
</tr>
<tr>
<td>1110</td>
<td>Go to eff. address at process time</td>
</tr>
<tr>
<td>1111</td>
<td>Half-word mode.</td>
</tr>
</tbody>
</table>

XY Format Mode

Bits:

\[
\begin{array}{c|c|c|c}
\hline
2 & 16 & 2 & 16 \\
\hline
10 & x & code & y \\
\hline
\end{array}
\]

The x and y words are in 2's complement. The 2-bit code is as follows:

00 set x and y, i.e., tell program the x and y coordinates of the plotter pen. This does not move the plotter pen.
01  Increment x,y with scale--move the plotter in an x and y direction the number of units equal to the product of the specified number times any scale factor.

10  Increment x,y without scale--move the plotter in an x and y direction the number of units specified.

11  go to x,y--position plotter at the given x and y coordinates.

Define CHAR/CHAR SET mode

Bits:

<table>
<thead>
<tr>
<th>18</th>
<th>12</th>
<th>6</th>
<th>3</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>110</td>
<td>Pointer</td>
<td>Character</td>
</tr>
</tbody>
</table>

Used in conjunction with byte manipulation instructions. The address and pointer refer to a word and position within the word of the first 4-bit byte. [The Byte length is set to 4 automatically].

The 8-bit octal code for the desired character is in the 'character' part of the word. The word or words containing the bytes are set up as follows:

$$[\Delta x | \Delta y | \therefore | \ldots | \#]$$

$\Delta x$ and $\Delta y$ may be zero. The first $\#$ after $\Delta x$ and $\Delta y$ terminates the string.

The codes for bytes are:

<table>
<thead>
<tr>
<th>code</th>
<th>action</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>Pen up</td>
</tr>
<tr>
<td>4</td>
<td>Pen down</td>
</tr>
<tr>
<td>5</td>
<td>Dot down--prints a dot at current point position.</td>
</tr>
<tr>
<td>10</td>
<td>Right</td>
</tr>
</tbody>
</table>
11  Up
12  Left
13  Down
14  Up Right
15  Up Left
16  Down Left
17  Down Right

This is used to specify a new character. Once done, the character may be called by the programmer in the normal manner.

The CHAR SET feature enables the use to define a complete set of characters. The 8-bits for 'character' must be 0 to use this mode. The address part is that location which is the beginning of the character table.

Once entered the new set is the only one addressable. To return to the old set use a DEFINE CHAR SET with UCTAB as the address.

N.B.: If lower case is not defined in the new set, any code with a 2^8 bit set to one will cause the program to hang up.

HALF-WORD MODE

Bits:  

<table>
<thead>
<tr>
<th>4</th>
<th>10</th>
<th>3</th>
<th>18</th>
</tr>
</thead>
<tbody>
<tr>
<td>1111</td>
<td>Not used</td>
<td>code</td>
<td>Number</td>
</tr>
</tbody>
</table>

In all of the below, the function is set to NUMBER. The 3-bit code is:

0  set scale
1  set horizontal tab. This sets tabs at n, 2n, 3n, etc.
set line feed [single space = 9 units]
set vertical tab. Sets tabs at n, 2n, 3n, etc.
set x
\[\text{set y}\] \text{number must be in}
go to x \text{2's complement}
go to y
APPENDIX I

CHARACTER SET (For lower case, add 200\textsubscript{8})

Octal Code

0  illegal character
1  pen down
2  subscript
3  pen up
4  superscript
5  leave mode without pen controls
6  enter mode with no pen controls executed
7  \textbf{(Bell; control C)}
10  set scale to zero
11  horizontal tab
12  line feed
13  vertical tab (will not go past a page boundary)
14  form feed (pages are 2000\textsubscript{8} units long)
15  carriage return
16  shift into upper case
17  shift into lower case
20  add 2\textsuperscript{0} to scale [for lower case subtract]
21  add 2\textsuperscript{1} to scale
22  add 2\textsuperscript{2} to scale
23  add 2\textsuperscript{3} to scale
24  add 2\textsuperscript{4} to scale
add $2^5$ to scale
add $2^6$ to scale
add $2^7$ to scale

direct along x-axis
direct along y-axis
upside down x-axis
upside down y-axis
backwards x-axis
backwards y-axis
upside down and backwards x-axis
upside down and backwards y-axis

upper case  lower case (if no entry, same as upper case)

4 unit space 3 unit space

!"

$ 

¢

'(apostrophe)

* (asterisk)  * (superscript asterisk)
(tilde)
| 103 | C | c   |
| 104 | D | d   |
| 105 | E | e   |
| 106 | F | f   |
| 107 | G | g   |
| 110 | H | h   |
| 111 | I | i   |
| 112 | J | j   |
| 113 | K | k   |
| 114 | L | l   |
| 115 | M | m   |
| 116 | N | n   |
| 117 | O | o   |
| 120 | P | p   |
| 121 | Q | q   |
| 122 | R | r   |
| 123 | S | s   |
| 124 | T | t   |
| 125 | U | u   |
| 126 | V | v   |
| 127 | W | w   |
| 130 | X | x   |
| 131 | Y | y   |
| 132 | Z | z   |
\[ \text{(no horizontal spacing)} \]

\[ \]