MASSACHUSETTS INSTITUTE OF TECHNOLOGY

PROJECT MAC

Artificial Intelligence Memo. No. 125. March 1967.
Program Memo.
Replacing Memo. No. 112. MAC-M-334.by Donald Sordillo.

CHAR PLOT

Michael Speciner

CHAR PLOT is a routine which enables one to use the CalComp plotter as a versatile output device. It is presently available as CHPLOT BIN (English CHAR PLOT) on tape MS 3.

The program CHAR PLOT is normally called by a PUSHJ P, PLOTC with a code representing a command or character (as defined in Appendix I) in accumulator C. Upon calling, the routine will either plot a character or line, or perform an internal control function. A 0 code initializes the routine, erasing any unexecuted (buffered) commands.

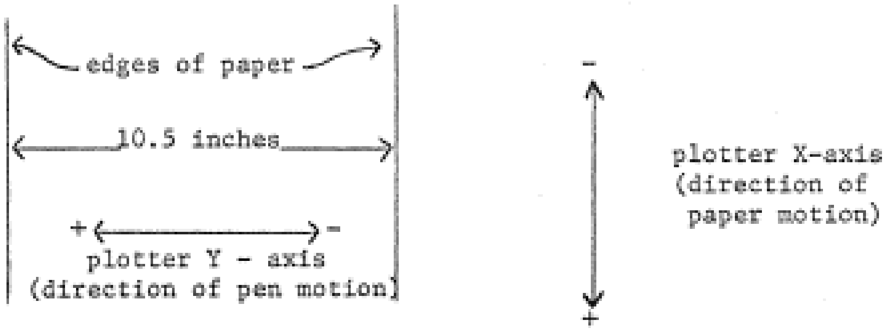
[Starred portions of this memo. explain the items necessary to use the routine for most purposes. Other portions describe more (less needed) options.]

*THE CALCOMP PLOTTER

The CalComp plotter is an incremental xy plotter which is capable of horizontal and vertical incrementing by .005 inch. Incrementing can be done separately (±i, ±j) or together (±i±j), where i and j are unit (.005 inch) vectors in the +X and +Y directions, respectively. Thus 8 directions of motion are possible:

Right	$+i$
Up	$+j$
Left	$-i$
Down	$-j$
UpRight	$+i+j$
UpLeft	$-i+j$
DownLeft	$-i-j$
DownRight	$+i-j$

[Note that this creates a right-hand coordinate system on the paper.]



Note that the plotter has no way of detecting where the pen is with respect to the paper. Pen motion requested past the edge of the paper is ignored by the plotter, but this should not be relied upon for exact positioning.

*CONVENTIONS

Radices. All integers are octal; all other numbers are decimal.

Relative units. 1 relative unit = SCALE * 1 absolute unit (.005 inch). Characters are plotted in relative units and vectors may also be so plotted.

Directions and Apparent Axes. The apparent axes may be set in one of eight possible orientations with respect to the plotter axes (see Appendix I, codes 30 - 37). Directions (R,U,L,D,UR,UL,DL,DR) are with respect to the apparent axes unless otherwise specified.

Parentheses around a symbol mean "contents of".

*GLOBAL SYMBOLS

PLOTC	Normal entry point.
CHARPL	Entry point which treats the 8 low-order bits of accumulator C as a character code regardless of the rest of accumulator C.
UCTAB	Beginning of upper-case table.
LCTAB	Beginning of lower-case table (=200+UCTAB).
CLNGTH	Routine (called by PUSHJ P, CLNGTH) which returns in C the length (in relative units) of the non-control character whose code was its argument in C. Upper and lower case can be specified using the 200 bit <u>or</u> by calling CLNGTH with 16 and 17 codes to change case.
X} Y}	Current coordinates of pen in absolute units with respect to the apparent axes. Initially (X)=(Y)=0. Not initialized by 0 code.
SCALE	Number of absolute units (.005 inch) in a relative unit. Characters are always plotted in relative units, and vectors may be so plotted.

	CRKBRK	Location for return from crock interrupt.
	NEWC	See DEFINE CHAR command (With the transfer bit 1....).
	SETC	
	SETW	
	SUBPLT	
user-defined	PP	
user-defined	CRKCHN	Crock PI channel. ($1 \leq$ CRKCHN≤ 7)
user-defined	LBUFF	Length of data buffers. Unless otherwise specified they are both 1000. However, they may be as little as 1 each. 4*LBUFF$\geq$LWBUFF, otherwise storage is wasted. LBUFF<200000.
user-defined	LWBUFF	
user-defined	P	Push-down pointer. ($1 \leq P \leq 17$)
user-defined	C	Argument accumulator. ($1 \leq C \leq 17$)

*CHARACTERS

For plotting a character, the input to the routine is a word whose 8 low-order bits contain an ASCII-type code (see Appendix I) and whose sign bit must be 0. For a normal character, the pen is assumed to be raised and is left raised; the character is plotted with its lower left corner at the initial position of the pen, and the pen is left ready for plotting the next character.

Characters may be in upper or lower case. The case of a character is specified in one of two ways: codes 16 and 17 may be used to shift case, character codes being interpreted in the current case; or while the routine is in upper case, the 200 bit of a character code may be 0 or 1 to specify upper or lower case, respectively. Note that when in lower case, character codes should have the 200 bit 0. The program starts in upper case (initialized by 0 code) and it is suggested that the 200 bit be used to specify case rather than shifting case with the 16 and 17 codes.

*VECTORS AND CONTROL FUNCTIONS

For plotting vectors and for performing control functions, the input to the routine is a word whose sign bit is 1. The high-order bits of this word are interpreted as follows:

bit 0↴

10	xy-format.
110	Define CHAR/CHAR SET.
1110	Go to effective address at process time with all accumulators restored to their value before the last interrupt. (To return to CHAR PLOT use JRST 12,@CRKBRK".)
1111	Half-word format.

*XY-FORMAT

```
                  012      22   24     43
        bits:     |10|  x  |code|   y   |
```

x and y are interpreted as 20-bit 2's complement numbers.

The code is interpreted as follows:

00	Set (X) and (Y). Set (X) to x and (Y) to y. Does <u>not</u> move pen.
01	Increment (x,y) with scale. Move the pen as specified by the vector SCALE* (x,y), i.e. x relative units R and y relative units U.
10	Increment (x,y) without scale. Move the pen as specified by the vector (x,y), i.e. x absolute units R and y absolute units R and y absolute units U.
11	Go to (x,y). Move the pen to the point (x,y).

$\left.\begin{array}{l}01\\10\\11\end{array}\right\}$ with pen down vect is plot

DEFINE CHAR/CHAR SET

```
               0123           11               12                22      43
     bits:     |110 | pointer |  transfer bit  | character code | address |
```

If the transfer bit is 0, the address and pointer refer to a word and position within the word of the first four-bit byte of the character to be specified by the character code. They are placed in the proper positions of a byte pointer which then defines the character. The byte-length is set to four automatically. The word or words containing the bytes are set up as follows:

|Δx|Δy| | |... |0 |

Δx and Δy are respectively the changes in (X) and in (Y) produced by plotting the character, measured in relative units. They are interpreted as positive 4-bit quantities. Either or both may be 0. The first 0 after Δx and Δy terminates the string. The codes for bytes are as follows:

2	Pen Up
4	Pen Down
5	Dot Down (plots a dot at current pen position)
10	Right one relative unit
11	Up one relative unit
12	Left one relative unit
13	Down one relative unit
14	UR
15	UL
16	DL
17	DR

With the transfer bit 1, the pointer bits are placed as in a byte pointer, bit 14 is set to 1, the address remains, and resulting word defines the character to be specified by the character code. When the character is used, the routine transfers tothe address (without dismissing the interrupt) with the character-defining word remaining in accumulator C. Returns to CHAR PLOT supplied are NEWC (continue), SETC (which plots the character whose defining word is in accumulator C, then continues), and SETW (which interprets the word in accumulator C as if it were an argument to PLOTC with bit 0 = 1, then continues). SCALE, X, and Y are global and so can be modified and/or tested before return. In addition, if the user defines a global symbol PP (a location containing a special push-down pointer), he may call a subroutine SUBPLT which works like PLOTC, only in interrupt mode, and thus performs the command which is its argument immediately. An example of its use would be:

```
EXCH P, PP
MOVE C, [WORD1]
PUSHJ P, SUBPLT
MOVE C, [WORD2]
PUSHJ P, SUBPLT
EXCH P, PP
JRST NEWC
```

The EXCH P, PP instructions are necessary since SUBPLT also executes these instructions before dismissing the interrupt and before returning to the calling program.

Thus a new character may be defined. Once defined, the character may be used in the normal manner.

The Define CHAR SET feature enables the user to define a complete set of characters. The 8 bits for character code being 0 specify this feature. The address is that location which is the beginning of a character table of words defining characters, with the first location defining the character with code 0, the second defining code 1, etc. Once a set is so defined, it is the only one addressable. In order to return to the original set, one may use a Define CHAR SET with UCTAB as the address.

*HALF-WORD FORMAT

```
                  01234           17      22 43
     bits:        |1111 | unused | code |     n|
```

The code is interpreted as follows:

0	Set scale to n.
1	Set horizontal tab to n. This sets tabs at n, 2n, 3n, etc. (to be interpreted in relative units). Initially 40.
2	Set line feed length to n (to be interpreted in relative units). Initially set for single space at 9.

3	Set vertical tab. (like horizontal tab)	
4	Set (X) to n.	
5	Set (Y) to n.	n interpreted as 22-bit
6	Go to (X)=n.	2's complement number.
7	Go to (Y)=n.	

APPENDIX I.

*INITIAL CHARACTER SET

code	upper case/lower case (add 200 to code if in upper case) (if no entry, same as upper case)
0	illegal character/illegal character if in lower case
1	pen down
2	subscript
3	pen up
4	superscript
5	enter mode with no pen controls executed
6	leave mode with no pen controls executed
7	🔔
10	set scale to zero (1 is first usable scale)
11	horizontal tab
12	line feed
13	vertical tab (will not go past a page boundary)
14	form feed (pages are 4000 absolute units long)
15	carriage return
16	shift into upper case
17	shift into lower case
20	add 2^0 to scale/subtract 2^0 from scale
21	add 2^1 to scale/subtract 2^1 from scale
22	add 2^2 to scale/subtract 2^2 from scale
23	add 2^3 to scale/subtract 2^3 from scale
24	add 2^4 to scale/subtract 2^4 from scale
25	add 2^5 to scale/subtract 2^5 from scale
26	add 2^6 to scale/subtract 2^6 from scale
27	add 2^7 to scale/subtract 2^7 from scale
30	orient apparent x-axis along plotter x-axis ⌊
31	orient apparent x-axis along plotter y-axis ⌋
32	orient apparent x-axis upside down along plotter x-axis ⌉
33	orient apparent x-axis upside down along plotter y-axis F
34	orient apparent x-axis backwards along plotter x-axis ⌈
35	orient apparent x-axis backwards along plotter y-axis ⌊
36	orient apparent x-axis upside down and backwards along plotter x-axis ⅃
37	orient apparent x-axis upside down and backwards along plotter y-axis ꟻ
	[Note that when one of the codes 30-37 is executed, (X) and (Y) are changed. The point (2000,2000) is a fixed point in this transformation.]

code	upper case	lower case
40	4-unit space	3-unit space
41	!	
42	"	
43	#	
44	$	ç
45	%	
46	&	
47	'	
50	(	
51	)	
52	* (asterisk)	* (superscript asterisk)
53	+	~
54	,	
55	- (minus)	- (hyphen)
56	.	
57	/	
60	Ø	
61	1	
62	2	
63	3	
64	4	
65	5	
66	6	
67	7	
70	8	
71	9	
72	:	
73	;	
74	<	^
75	=	⊻
76	>	∨
77	?	
100	@	
101	A	a
102	B	b
103	C	c
104	D	d
105	E	e
106	F	f
107	G	g
110	H	h
111	I	i
112	J	j

code	upper case	lower case
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
133	[	
134	\	/ (no horizontal spacing)
135	]	
136	↑	↓
137	←	→
140-157	not used	
160	R one relative unit	R one absolute unit
161	U one relative unit	U one absolute unit
162	L one relative unit	L one absolute unit
163	D one relative unit	D one absolute unit
164	UR one*relative unit	UR one*absolute unit
165	UL one*relative unit	UL one*absolute unit
166	DL one*relative unit	DL one*absolute unit
167	DR one*relative unit	DR one*absolute unit
170	DD	
171-177	not used	

*actually $\sqrt{2}$ units

APPENDIX II

SUMMARY OF COMMANDS

bit: 0	1	34 43	
0	unused	code	CHARACTER

bit: 01	2	22	24 43	
10	x	00	y	SET (X) AND (Y)
10	Δx	01	Δy	INCREMENT W/ SCALE
10	Δx	10	Δy	INCREMENT WOUT SCALE
10	x	11	y	GO TO (x,y)

bit: 012	3	11	12	22 43	
110	pointer	t.b.	code	address	DEFINE CHARACTER
110	unused		0	address	DEFINE CHAR SET

bit: 01234	4	15	16	22 43	
1110	unused	I	X	address	GO TO EFF. ADDRESS

bit: 0123	4	17	22 43	
1111	unused	000	n	SET SCALE
1111	unused	001	n	SET HOR. TAB
1111	unused	010	n	SET LINE FEED
1111	unused	011	n	SET VERT. TAB
1111	unused	100	n	SET (X)
1111	unused	101	n	SET (Y)
1111	unused	110	n	GO TO (X)=n
1111	unused	111	n	GO TO (Y)=n