

USER INTERFACE TO
A FLEXIBLE CHARACTER GENERATOR

BY

UDAY G. GUJAR

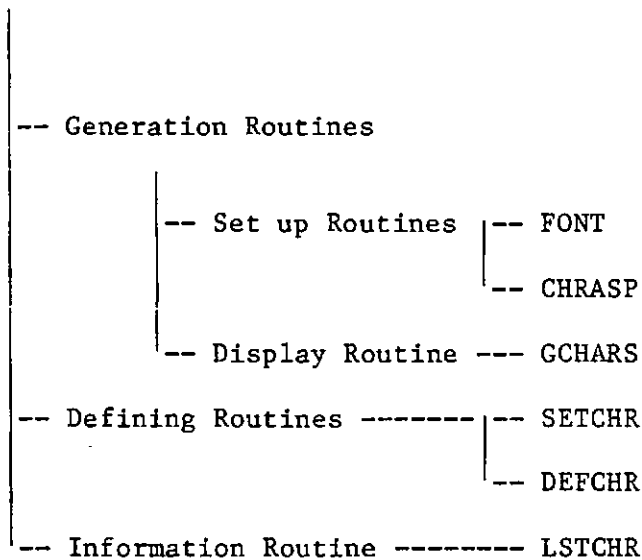
TR86-036, OCTOBER 1986

1.0 INTRODUCTION

This document contains the user interface for a flexible character generator described in [1,2,3,4]. This generator works in the environment of a device independent computer plotting system [5,6]. Several examples have appeared in [1,2,3] which are not repeated in this document; access to either [1], [2] or [3] is highly recommended. An application of this generator has appeared in [7].

The user conception can be broken down as

Generator



The usage of these routines is described in the remaining pages. All the routines have variable number of arguments. The optional arguments are enclosed in square brackets which, obviously, should not appear in the actual statement. The user may omit, consecutively from the right hand side, as many of these optional arguments as desired. The argument &N, where N represents the error return statement, may appear anywhere in the calling sequence.

2.0 SET UP ROUTINES

2.1 FONT

Purpose: To select a character font

Usage: CALL FONT (FTNAME [,POST,UNDEF,&N])

FTNAME: APLCHR APL characters
DBLCHR Double line characters
DEFONT Default characters (as in CHARS, see [5,6])
FINCHR More refined "fine" characters
FIXWID Fixed width characters
SPESYM Special symbols
SCRIPT Script letters
MARATI Devnagari script (for Marathi, Hindi, etc.)
MUSME Indian music in English notation (Ravi Shankar)
MUSMM Indian music in Hindi notation (Bhatkhande)

Must be defined in an EXTERNAL statement

POST: 'CENT' Centered characters
'UNCE' Uncentered characters

UNDEF: 'DELI' treat undefined characters as a delimitar
'IGNO' ignore undefined characters
'BLOB' plot a blob for an undefined character

&N: Error return statement number

2.2 CHRASP

Purpose: To specify height, width and types of transformations for the characters to be plotted.

Usage: CALL CHRASP ([HEIGHT, WIDTH, TRAN, DEG, ----, TRAN, DEG, &N])

HEIGHT: +ve → height in inches
 -ve → absolute value gives height in Y user units
WIDTH: +ve → width in inches
 -ve → absolute value gives width in X user units

TRAN: 'ROTA' Rotation
 'REVO' Revolve
 'SKEW' Skew
 'FALL' Fall
 'STRE' Stretch

DEG: Angle in degrees (REAL*4)

&N: Error return statement number.

Notes:

- a) Stretch is always considered the first one since it is non-linear.
- b) If any transformation is specified more than once, only the first one is taken.
- c) Fall and skew are mutually exclusive, first one is taken.
- d) Pair TRAN, DEG may be repeated as many times as required.

3.0 DISPLAY ROUTINE - GCHARS

Purpose: To plot a specified string at a specified starting point.

Usage: CALL GCHARS (X,Y, IARRAY)

X,Y - (X,Y) coordinates of the starting point, REAL*4.

IARRAY - String of characters to be plotted; may contain control characters. Control characters are two characters - control symbol followed by a control character. The default values for the control characters are:

\$E Delimiter
\$U Upper case shift
\$L Lower case shift
\$N Carriage return
\$B Subscript
\$A Superscript
\$F Line feed

These defaults can be changed by calling SETCHR. For carriage return, subscript, superscript and line feed, the plotting pen moves to (XN,YN) given by:

$$\begin{aligned} \$N \rightarrow XN &= XS + ATBX * w * \cos\alpha - ATBY * h * \sin\alpha \\ YN &= YS + ATBX * w * \sin\alpha + ATBY * h * \cos\alpha \end{aligned}$$

$$\begin{aligned} \$B \rightarrow XN &= x + ATBX * w * \cos\alpha - ATBY * h * \sin\alpha \\ YN &= y + ATBX * w * \sin\alpha + ATBY * h * \cos\alpha \end{aligned}$$

w and h are modified as: $w = w * \text{RATIOX}$, $h = h * \text{RATIOY}$ for $c < 0$
 $w = w / \text{RATIOX}$, $h = h / \text{RATIOY}$ for $c > 0$

\$A → same as in \$B, except h and w are modified as

$w = w * \text{RATIOX}$, $h = h * \text{RATIOY}$ for $c > 0$
 $w = w / \text{RATIOX}$, $h = h / \text{RATIOY}$ for $c < 0$

$$\begin{aligned} \$F \rightarrow XN &= x + w * ATBX * \cos\alpha - h * ATBY * \sin\alpha \\ YN &= y + w * ATBX * \sin\alpha + h * ATBY * \cos\alpha \end{aligned}$$

where,

XS,YS - starting position of the character string
ATBX,ATBY - attribute values
 α - angle of rotation
(x,y) - current position
h - height of the characters
w - width of the characters
RATIOX, RATIOY - size factors for super/subscripted characters.
c - an internal counter, zero initially, modified as
c = c+1 when superscript is encountered
c = c-1 when subscript is encountered

GCHARS (continued)

Notes:

- a) For \$A and \$B, if the absolute values of ATBY are same, the effects can be cancelled to move back to the original line. In addition if RATIOX and RATIOY for \$A are the same as those for \$B respectively, the original size of the characters can be restored.
- b) The defaults for various variables in most fonts are:

control symbol: \$

Delimiter: \$E

\$N → ATBX = 0, ATBY = -1.7

\$B → ATBX = 0, ATBY = -.65, RATIOX = 1.0, RATIOY = 1.0

\$A → ATBX = 0, ATBY = .65, RATIOX = 1.0, RATIOY = 1.0

\$F → ATBX = 0, ATBY = -1.7

These can be changed by calling SETCHR.

4.0 DEFINING ROUTINES

4.1 SETCHR

Purpose: Used to change the values of the attributes of a control character or a font parameter or to define a new control character.

Usage: CALL SETCHR (NAME, SYMBOL [,ATBX, ATBY, RATIOX, RATIOY, ICODE, &N])

NAME:

- 'FXSP' Fixed part of spacing between characters (0, ' ', .02)
- 'NOMY' Nominal values (4, ' ', 10, 10)
- 'CYSM' Control symbol (8, '\$')
- 'DELI' Delimiter (12, 'E')
- 'UPSH' Upper case shift (16, 'U')
- 'LOSH' Lower case shift (16, 'L')
- 'CRRT' Carriage return (20, 'N', 0., -1.7)
- 'SUBS' Subscript (24, 'B', 0., -.65, 1., 1.)
- 'SUPS' Superscript (24, 'A', 0., .65, 1., 1.)
- 'LINF' Line feed (28, 'F', 0., -1.7)

SYMBOL: LOGICAL*1 Symbol for a control character (excluding digits)

ATBX,ATBY: X and Y attributes, REAL*4 or INTEGER*4

RATIOX, RATIOY: X and Y ratios (REAL*4 or INTEGER*4), +ve; currently used for only SUBS & SUPS

ICODE: Code of a new control character; must be 0 to 28 in multiple of 4 to use the existing codes to obtain the corresponding actions.

&N: Error return statement number

Notes:

- a) Defaults, for most fonts, for each of the NAME are given in parenthesis in the format (ICODE, SYMBOL, ATBX, ATBY, RATIOX, RATIOY)
- b) For 'NOMY', ATBX and ATBY must be integers in the range 0 to 255 inclusive.
- c) ICODE determines the action and how ATBX, ATBY, RATIOX, RATIOY are used.

4.2 DEFCHR

Purpose: Used to change or define the shape of a character.

Usage: CALL DEFCHR (CHAR, INOMX, INOMY, IC, IX, IY, M, &N)

CHAR: LOGICAL*1, EBCDIC code of a character to be changed or defined within the boundary of the defined character set.

The current defined ranges for various fonts are

```

APLCHR:  '&' to ' ' (or 9 to 199)
DBLCHR:  '-' to '9' (or 96 to 249)
DEFONT:  'ç' to '9' (or 74 to 249)
FINCHR:  'ç' to '9' (or 74 to 249)
FIXWID:  'ç' to '9' (or 74 to 249)
SCRIPT:  'A' to 'Z' (or 193 to 233)
SPESYM:  'A' to '9' (or 193 to 249)
MARATI:  '.' to '9' (or 75 to 249)
MUSME:   '|' to 'S' (or 79 to 226)
MUSMM:   '|' to 'S' (or 79 to 226)

```

INOMX, INOMY: Sizes of X and Y nominal values of grids in which new character is defined; +ve.

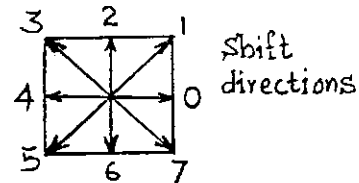
IC,IX,IY Arrays, (control, X, Y) of M points defining strokes. (IX,IY) may contain superimpose and/or shift strokes; corresponding IC is then ignored. IC<0, position at (IX, IY); IC=0, point at (IX, IY); IC>0, line from previous position to (IX,IY).

(IX,IY) values for special strokes are:

```

Superimpose: (15,15) or (255,255) + must be the first stroke
Shift: Direction 0  -(15,0) or (255,240)
              1  -(15,1) or (255,241)
              .
              .
              .
              7  -(15,7) or (255,247)
Reserved: (15,8) or (255,248)
          (15,9) or (255,249)
          .
          .
          .
          (15,14) (255,254)

```



Special stroke values (15,n) for 0 ≤ n ≤ 15 are used for compressed data structure while values (255,n) for 240 ≤ n ≤ 255 are used for non-compressed data structure.

M: Number of points. [IX(M), IY(M)] is the last positioning stroke.

&N: Error return statement number

5.0 INFORMATION ROUTINE - LSTCHR

Purpose: To retrieve the values of the attributes of a control character or a font parameter

Usage: CALL LSTCHR (NAME, SYMBOL [,ATBX,ATBY,RATIOX,RATIOY,ICODE,&N])

NAME:	'FXSP'	Fixed part of spacing between characters
	'NOMV'	Nominal values
	'CYSM'	Control symbol
	'DELI'	Delimiter
	'UPSH'	Upper case shift
	'LOSH'	Lower case shift
	'CRRT'	Carriage return
	'SUBS'	Subscript
	'SUPS'	Superscript
	'LINF'	Line feed

These are existing names. New names may be applicable, if defined through SETCHR.

SYMBOL:	LOGICAL*1 variable (dummy value ' ')
ATBX, ATBY:	REAL*4 variables (dummy values 999.)
RATIOX, RATIOY:	REAL*4 variables (dummy values 999.)
ICODE:	INTEGER*4 variable (dummy value 999)
&N:	Error return statement number

Note:

- a) Dummy values are returned when the corresponding arguments are not applicable.

6.0 EXAMPLES

Several examples have appeared in [1,2,3]; these are not repeated here. Two examples, one showing the effects of fonts, control characters and transformations and the other illustrating the process of generation of a customer characters and the effects of superimposition of symbols, are included. Complete source listings and the generated graphics images are given. Following these examples, the EBCDIC code for characters in each of the fonts is given in a tabular form; this table has been produced using the flexible character generator.

```
//FIGURE1 JOB
/*JOBPAR4 S=5,L=1,NONEWS,R=2048
// EXEC FORTPLOT,TYPE=FILEPLOT
//FORT.SYSIN DD *
    EXTERNAL FINCHP,SCRIPT,DBLCHP
    CALL APEA ( 5.0, 9.0 )
    CALL SETPLT ( 0.0, 0.0, 500.0, 990.0 )
    CALL RECT ( 0.0, 0.0, 6.0, 9.0 )
    CALL FONT ( FINCHP )
    CALL CHRASP ( 0.35, 0.25 )
1 CALL GCHARS ( 40.0, 755.0, 'UPPER CASE$E')
2 CALL GCHARS ( 40.0, 655.0, 'LOWER CASE$E')
    CALL CHRASP ( .1,.1)
    CALL GCHARS ( 50.,20.,
*Fig. 1: Effects of fonts, control characters and transforms$E')
    CALL FONT ( DBLCHP )
    CALL CHRASP ( 0.35, 0.25 )
    CALL SETCHR ( 'SUPS','A',999.,999.,.7,.7)
3 CALL GCHARS ( 40.0, 532.0, 'SUPERSCRIPT$E')
    CALL SETCHR ( 'SUBS','B',999.,999.,.7,.7)
4 CALL GCHARS ( 40.0, 427.0, 'SUBSCRIPT$E')
    CALL FONT ( SCRIPT )
    CALL CHRASP ( 0.35, 0.25 )
5 CALL GCHARS ( 40.0, 293.0, 'THIS IS ANCARRIAGE RETURN$E')
    CALL CHRASP ( 0.35, 0.25,'REVO',30. )
6 CALL GCHARS ( 40.0, 133.0, 'HERE IS A$ LINE FEED$E')
    CALL FONT ( DBLCHR )
    CALL CHRASP ( 0.35, 0.25 )
    CALL CHRASP ( 0.35, 0.35 , 'ROTATE'; 20.0 )
7 CALL GCHARS ( 355.0, 745.0, 'ROTATE$E' )
    CALL CHRASP ( 0.35, 0.35, 'REVOLVE', 30.0 )
8 CALL GCHARS ( 330.0, 655.0, 'REVOLVE$E' )
    CALL CHPASP ( 0.35, 0.35, 'SKEW', 40.0 )
9 CALL GCHARS ( 355.0, 565.0, 'SKEW$E' )
    CALL CHRASP ( 0.35, 0.35, 'FALL', 40.0 )
10 CALL GCHARS ( 355.0, 485.0, 'FALL$E' )
    CALL CHRASP ( 0.35, 0.35, 'STPETCH', 17.5 )
11 CALL GCHARS ( 335.0, 337.0, 'STPETCH$E' )
    CALL ENDPLT
    STOP
    END
//LKED.USERLIB DD DJSP=SHR,DSN=A.M4235.GCHARS
```

```

//FIGURE2 JOB
/*JOBPARM S=5,L=1,NONEWS,P=2048
// EXEC FORTPLOT,TYPE=FILEPLOT
//FORT.SYSIN DD *
EXTERNAL FINCHR,SCRIPT
INTEGER QIC(55),QIX(55),QIY(55),AIC(3),AIX(3),AIY(3),BIC(12),
* BIX(12),BIY(12),CIC(5),CIX(5),CIY(5),DIC(12),DIX(12),DIY(12)
CALL AREA ( 5.0, 9.0 )
CALL SETPLT ( 0.0, 0.0, 600.0, 900.0 )
CALL PECT ( 0.0, 0.0, 6.0, 9.0 )
CALL LNSPLT ( (0.0,600.0),(470.0,470.0) )
C ***** GENERATION OF A CUSTOM CHARACTER *****
READ(5,*) (QIC(I), QIX(I), QIY(I),I=1,55)
CALL FONT ( SCRIPT )
CALL CHRASP ( 1.0, 0.7 )
CALL GCHARS ( 132.0, 750.0, 'QUEUE$E' )
CALL DEFCHR ( 'Q', 30, 35, QIC, QIX, QIY, 55 )
CALL GCHARS ( 132.0, 550.0, 'QUEUE$E' )
C ***** SUPERIMPOSITION OF SYMBOLS *****
READ(5,*) (AIC(I), AIX(I), AIY(I),I=1,3)
READ(5,*) (BIC(I), BIX(I), BIY(I),I=1,12)
READ(5,*) (CIC(I), CIX(I), CIY(I),I=1,5)
READ(5,*) (DIC(I), DIX(I), DIY(I),I=1,12)
CALL DEFCHR ( 'A', 20, 20, AIC, AIX, AIY, 3 )
CALL DEFCHR ( 'B', 20, 20, BIC, BIX, BIY, 12 )
CALL DEFCHR ( 'C', 20, 20, CIC, CIX, CIY, 5 )
CALL DEFCHR ( 'D', 20, 20, DIC, DIX, DIY, 12 )
CALL GCHARS ( 36.0, 320.0, 'A B C D$E' )
CALL GCHARS ( 36.0, 128.0, 'AB ABC ABD AD$E' )
C ***** WRITE LABELS *****
CALL FONT ( FINCHR )
CALL CHRASP ( 0.1, 0.1 )
CALL GCHARS ( 132.0, 705.0, '$L(A) $UO$RIGINAL STRING$E' )
CALL GCHARS ( 132.0, 513.0, '$L(B) $UN$LEW STRING$E' )
CALL GCHARS ( 36.,285., '(c) Newly-defined symbols$E' )
CALL GCHARS ( 36.,93., '(d) Combinations of symbols in (c)$E' )
CALL GCHARS ( 100.,40., 'Fig. 2: Generation of custom characters and
*$N Superimposition of symbols$E' )
CALL ENDPLT
STOP
END

```

```

//LKED.USERLIB DD DISP=SHR,DSN=A.M4235.GCHARS
//GO.SYSIN DD *
0 11 32 1 7 31 1 4 29 1 2 27 1 1 25 1 0 22 1 0 18
1 1 15 1 2 13 1 4 11 1 7 9 1 9 8 1 12 8 1 6 5
1 6 4 1 7 4 1 8 5 1 10 6 1 11 6 1 13 5 1 16 3
1 18 1 1 20 0 1 23 0 1 25 1 1 26 2 1 26 4 1 25 5
1 23 3 1 21 3 1 15 6 1 14 6 1 11 7 1 13 8 1 11 8
1 8 9 1 5 12 1 4 14 1 3 18 1 3 22 1 4 26 1 5 28
1 8 31 1 11 32 1 13 32 1 17 31 1 20 29 1 22 27 1 23 25
1 24 22 1 24 18 1 23 15 1 22 13 1 20 11 1 17 9 1 13 8
1 16 9 1 19 12 1 20 14 1 21 18 1 21 22 1 20 26 1 19 28
1 16 31 1 13 32 0 30 0 END OF DATA FOR Q
0 0 0 1 0 20 0 6 0 END OF DATA FOR A
0 255 255 0 0 20 1 8 20 1 11 19 1 12 18 1 13 16 1 13 14
1 12 12 1 11 11 1 8 10 1 0 10 0 16 0 END OF DATA FOR B
0 255 255 0 8 10 1 12 1 1 14 0 0 17 0 END OF DATA FOR C
0 255 255 0 0 10 1 8 10 1 11 9 1 12 8 1 13 6 1 13 4
1 12 2 1 11 1 1 8 0 1 0 0 0 16 0 END OF DATA FOR D

```

UPPER CASE

ROTATE

lower case

REVOLVE

SUPER^{SCRIPT}

SKEW

SUB_{SCRIPT}

FALL

STRETCH

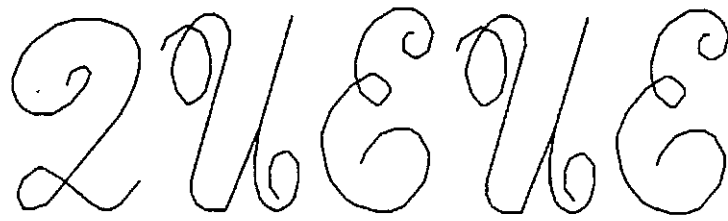
THIS IS A

CARRIAGE RETURN

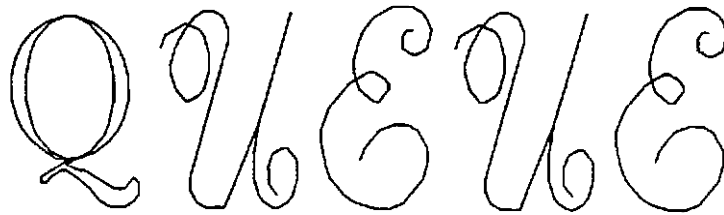
HERE IS A

LINE FEED

Fig. 1• Effects of fonts, control characters and transforms



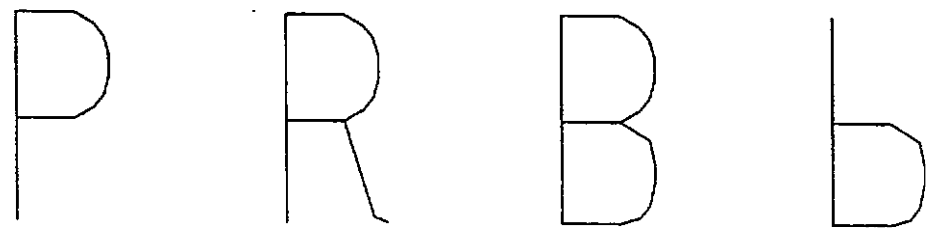
(a) Original string



(b) New string



(c) Newly-defined symbols



(d) Combinations of symbols in (c)

Fig. 2. Generation of custom characters and Superimposition of symbols

REFERENCES

1. F.W.L. So, 'A flexible software character generator', M.Sc.(C.S.) Thesis, School of Computer Science, University of New Brunswick, Fredericton, N.B., Canada, 150 pages, October 1982.
2. U.G. Gujar and F.W.L. So, 'Computer display of characters', TR83-023, School of Computer Science, University of New Brunswick, Fredericton, N.B., Canada, August 1983.
3. U.G. Gujar and F.W.L. So, 'A flexible software character generator', Software-Practice and Experience, Vol. 15, #2, pp. 191-218, Feb. 1985.
4. U.G. Gujar and F.W.L. So, 'Generation and manipulation of fonts for computer graphics', APICS Computer Science Seminar Series, November 1982.
5. U.G. Gujar, 'Computer plotting', University of New Brunswick, Fredericton, N.B., Canada, 1972.
6. U.G. Gujar, 'A device independent computer plotting system', Proceedings of ACM Symposium on Graphics Languages, Computer Graphics, 10(1), 85-100 (1976).
7. U.G. Gujar, 'A character generator and its applications to Devnagari script', IEEE International Conference on Computers, Systems and Signal Processing, Bangalore, India, pp. 1518-1522, Dec. 1984.