DECUS

PROGRAM LIBRARY

DECUS NO.	8-147
TITLE	Incremental Plotter Printout Subroutines
AUTHOR	Dr. Phillip J. Best
COMPANY	University of Michigan Ann Arbor, Michigan
DATE	April 20, 1968
SOURCE LANGUAGE	

Although this program has been tested by the contributor, no warranty, express or implied, is made by the contributor, Digital Equipment Computer Users Society or Digital Equipment Corporation as to the accuracy or functioning of the program or related program material, and no responsibility is assumed by these parties in connection therewith.

INCREMENTAL PLOTTER PRINTOUT SUBROUTINES

DECUS Program Library Write-up — DECUS No. 8-147

1. IDENTIFICATION

1.2 Incremental Plotter Printout Subroutines

1.3 April 20, 1968

2. ABSTRACT

A group of subroutines providing character-output facilities for the incremental plotter is presented as a package. Virtually all the ASCII characters may be printed in any of eight formats and in 63 sizes. One routine sets a control code to determine the size and orientation of the characters and the direction the lines are to run; another routine prints out a string of characters according to this control code; a third routine prints just one character held in AC 6-11; and a fourth routine prints the signed decimal equivalent of the contents of the accumulator.

3. REQUIREMENTS

3.1 Storage

These routines require 5 memory pages (1177 registers).

3.2 Equipment

Basic PDP-8, Type 350 Plotter Control and Plotter

3.3 Other Subroutines Needed

Digital 8-12-U "Incremental Plotter Subroutine"

4. USAGE

4.1 Loading

The library tape that is supplied is a symbolic tape which has no origin setting and has one undefined variable, PLOTX. These routines may be assembled by preceding the symbolic tape with a tape reading "*XXXX; PLOTX=YYYY; PAUSE;" where

XXXX is the starting address of these routines (which must be the first location of some page of memory) and YYYY is the first location of Digital 8-12-U. Alternatively, these routines may be assembled directly after Digital 8-12-U merely by resetting the origin to the start of some memory page. (((The binary tape supplied contains these routines at 600-1777 and references PLOTX (Digital 8-12-U) to 3000)))

4.2 Calling Sequence

PLTSTG prints a string of characters according to a control code which is set by TYPLOT. PLTSTG is called with an effective JMS XXXX+400 with the address of the string of characters to be printed held in the accumulator.

Sample:

```
                    CLA
                    TAD STGADR    /address of string
                    JMS I RFPLTS
                    Return
                    .
                    .
                    .
        RFPLTS,     PLTSTG
        STGADR,     STRG
        STRG,       0102          /A B
                    0300         /C  end of string
```

TYPLOT is called with an effective JMS XXXX+144 with the address of the string to be printed held in the contents of the accumulator and the control code in the next successive memory location following the call. TYPLOT uses the control code to set the parameters of PLTSTG, and then calls PLTSTG to print the string of characters.

Sample:

```
                    CLA

                    TAD STGADR     /address of string

                    JMS I RFTYPL

                    WXYZ           /control code

                    Return
                    .
                    .
                    .
          RFTYPL,   TYPLOT
```

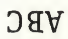
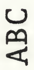
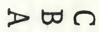
where WXYZ is the control code to be interpreted as follows:

W: indicates the direction the lines are to run relative to the orientations of the characters. W=0, the next character is to the relative right of the previous character (i.e, a normal printed line); W=1, the next character is printed relatively above the previous character.

X: indicates the orientation of the characters. X=0, characters are right side up on the abscissa; X=1, characters are inverted; X=2, characters are rotated 90 degrees right; X=3, characters are rotated 90 degrees left.

YZ: interpreted as a two-digit octal number indicating the number of multiples of the basic 5 x 7 plotter-step size each character is to be.

Sample:

Control Code	Output of String 0102 0300
0001	ABC
0004	ABC
1001	C B A
0101	
0301	
1201	

DECPLT is called with an effective JMS XXXX with the two's complement octal number whose signed decimal equivalent is to be printed in the contents of the accumulator. This routine suppresses the "+" sign and leading zeroes.

PLT1 is called with an effective JMS XXXX+100, and prints a single character whose 6-bit code is held in AC 6-11.

<u>NOTE</u>: TYPLOT is the only routine which changes the control code. All other routines print according to the control code of the previous call of TYPLOT. To change control codes without printing anything, call TYPLOT with a null string (0000) to be printed.

<u>Sample</u>:

```
                        CLA

                        TAD NULSTR

                        JMS I RFTYPL

                        WXYZ          /the new control code
                        .
                        .
                        .
        NULSTR,         .+1

                        0000

        RFTYPL,         TYPLOT
```

5. RESTRICTIONS

These routines have no provision to prevent the output of a string too long or with characters too large to be written on the plotter page. Strings which extend beyond the 4096 plotter-step limit of Digital 8-12-U will exit with the incorrect (by a multiple of $10,000_8$) values of PLOTNX,Y (in Digital 8-12-U) for the current pen position.

6. DESCRIPTION

6.1 <u>Discussion</u>

Each character is represented with a 6-bit code (similar to the code of TYPSTG (Digital 8-18-U-Sym) consisting of the six low-order bits of the 8-bit ACSII code.

PLTSTG takes this code and translates it to the address of the first of five successive words which contain the movement codes for the specified character. Each movement code word contains two movement codes. Movement codes correspond to positions on a 5 x 7 X-Y grid, with X codes of 6 and 7 representing 'pen up' and 'pen down' respectively. The letter "A" for example is coded as follows: 70 (pen down), 13 (⁄), 33 (/‾), 13 (/‾), 26 (/-), 40 (A), 40 ..., which is packed into words from right to left:

```
1370    /A
1333
4026
4040
4040
```

The current pen position (PLOTNX,Y of Digital 8-12-U) is modified during the execution of PLTSTG; however, at exit of PLTSTG the pen position is correctly recorded in PLOTNX,Y with the respect to the position at the call of PLTSTG.

7. METHODS - (Not Applicable)

8. FORMAT

The legal characters, their 6-bit packed codes, and the corresponding 8-bit ACSII codes are best illustrated by the following table.

	String Terminator	
01	301	A
02	302	B
03	303	C
04	304	D
05	305	E
06	306	F
07	307	G
10	310	H
11	311	I
12	312	J
13	313	K
14	314	L
15	315	M
16	316	N
17	317	O
20	320	P
21	321	Q
22	322	R
23	323	S
24	324	T
25	325	U
26	326	V
27	327	W
30	330	X
31	331	Y
32	332	Z
40	240	space
44	244	$
45	245	%
46	246	&
47	247	'
50	250	(
51	251	)
52	252	*
53	253	+
54	254	,
55	255	-
56	256	.
57	257	/
60	260	0
61	261	1
62	262	2
63	263	3
64	264	4
65	265	5
66	266	6
67	267	7
70	270	8
71	271	9
72	272	:
73	273	;
74	274	<
75	275	=
76	276	>
77	277	?

Strings are coded 2 characters per word, read from left to right. "00" code serves to terminate the string and causes exit from the PLTSTG routine.

9. EXECUTION TIME

These subroutines are device-speed limited.

10. PROGRAM

10.4 Program Listing: (attached)

11. DIAGRAMS: (Not applicable)

12. REFERENCES: (Not applicable)

```
/
/MICHAEL P STRYKER
/PHILLIP J BEST
/BRAIN RESEARCH LAB
/UNIVERSITY OF MICHIGAN
/ANN ARBOR, MICHIGAN
/
/INCREMENTAL PLOTTER PRINTOUT SUBROUTINES
/
/THESE ROUTINES, WHICH OCCUPY FIVE PAGES OF MEMORY, MUST
/BE USED WITH DIGITAL 8-12-U PLOT SUBROUTINE.
/
/THE FIVE PAGES MAY BE ASSEMBLED STARTING AT THE FIRST
/LOCATION OF ANY PAGE OF MEMORY.  TO ASSEMBLE, FIRST  MAKE
/A TAPE AS FOLLOWS TO PRECEDE THE TAPE OF THE PRINTOUT SUB-
/ROUTINES:  "*XXXX; PLOTX=YYYY; PAUSE; ", WHERE XXXX IS
/THE FIRST LOCATION OF THE PAGE WHERE THE PRINTOUT
/ROUTINES ARE TO START, AND YYYY IS THE FIRST LOCATION
/OF "PLOTX" (DIGITAL 8-12-U).
/
/ALL THE PRINTOUT ROUTINES EXIT WITH CURRENT PEN LOCATION
/CORRECTLY HELD IN "PLOTNX,Y" OF "PLOTX" (8-12-U).
/


/DECPLT  --  A SIGNED DECIMAL PRINTOUT ROUTINE FOR
/INCREMENTAL PLOTTER.  THIS ROUTINE USES PLTSTG; DIGITS
/ARE PRINTED OUT IN THE SAME FORMAT AS THAT OF THE PREVIOUS
/CALL OF TYPLOT.  LEADING ZEROS AND THE "+" SIGN ARE
/SUPPRESSED.
/THIS ROUTINE IS A MODIFICATION OF DIGITAL 8-22-U-SYM.
/
/CALL WITH NUMBER TO BE PRINTED OUT IN C(AC)

DECPLT,   0
          SMA
          JMP PLUS
          CIA
          DCA VALUE
          TAD L55     /"-" SIGN
          JMP .+3
PLUS,     DCA VALUE
          TAD L40     /SPACE
          JMS PLT1
          DCA SKPF
          TAD M4
          DCA CNT4
          DCA DIGIT           /CLEAR
          TAD CNTRZA
          DCA CNTRZB          /SET COUNTER TO FOUR
          TAD ADDRZA
          DCA ARROW           /SET TABLE POINTER
          SKP
          DCA VALUE           /SAVE
          CLL
          TAD VALUE
```

```
ARROW,    TAD TENPWR          /SUBTRACT POWER OF TEN
          SZL
          ISZ DIGIT           /DEVELOP BCD DIGIT
          SZL
          JMP ARROW-3         /LOOP
          CLA                 /HAVE BCD DIGIT
          TAD DIGIT           /GET DIGIT
          ISZ CNT4   /IS THIS THE LAST Ø?
          SKP        /NO
          JMP .+3     /YES, DON'T SUPPRESS IT.
          SNA
          JMP CK
          ISZ SKPF
          TAD K6Ø
OUT,      JMS PLT1   /PLOT OUT THE DIGIT
          DCA DIGIT           /CLEAR
          ISZ ARROW           /UPDATE POINTER
          ISZ CNTRZB          /DONE ALL FOUR?
          JMP ARROW-1         /NO: CONTINUE
          JMP I      DECPLT  /YES:  EXTI
ADDRZA,   TAD TENPWR
CNTRZA,   -4
TENPWR,   -175Ø               /ONE THOUSAND
          -Ø144               /ONE HUNDRED
          -ØØ12               /TEN
          -ØØØ1               /ONE
VALUE,    Ø
DIGIT,    Ø
CNTRZB,   Ø
CK,       TAD SKPF
          SNA CLA
          TAD M2Ø
          JMP OUT-1
SKPF,     Ø
CNT4,     Ø
M4=CNTRZA
L55,      55
M2Ø,      -2Ø
L4Ø,      4Ø
K6Ø,      6Ø

/PLT1  --  PRINTS OUT ONE CHARACTER HELD IN  AC 6-11
/
*DECPLT+1ØØ
PLT1,     Ø          /ENTER WITH 1 CHR IN AC
          RTL
          RTL
          RTL
          AND L77ØØ  /MASK TO ASSURE EXIT
          DCA STG
            TAD STG+1
          JMS I RPLTST  /JMS PLTSTG
          CLA CLL
          JMP I PLT1
STG,      Ø
          STG
L77ØØ,    77ØØ


*DECPLT+144
```

```
/TYPLOT    ROUTINE TO FETCH CONTROL CHARACTER AND PLOT A
/STRING OF CHARACTERS.  ENTER WITH ADDRESS OF STRING IN AC
/AND CONTROL CHARACTER IN THE NEXT LOCATION FOLLOWING THE
/CALL.    SAMPLE:    TAD STGADD
/                    JMS TYPLOT
/                    WXYZ       (CONTROL CHARACTER--OCTAL)
/CONTROL CHARACTER IS TO BE INTERPRETED AS FOLLOWS:
/W:  THE DIRECTION THE LINE IS TO RUN.  W-Ø; NEXT CHARACTER
/        IS TO LEFT.  W=1; NEXT CHARACTER IS ABOVE PREVIOUS CHR.
/X:  THE ORIENTATION OF THE CHRS.  X=Ø; CHRS ARE RIGHT SIDE UP
/        X=1; CHRS ARE INVERTED.  X=2; CHRS ARE ROTATED 9Ø
/        DEGREES TO THE RIGHT.  X=3; ROTATED 9Ø DEG. LEFT.
/YZ:  TAKEN TOGETHER AS A TWO-DIGIT NUMBER INDICATE THE SIZE
/        OF THE CHRS = THE NUMBER OF MULTIPLES OF THE BASIC
/        6 BY 4 PLOTTER-STEP SIZE.
/
/
/
TYPLOT,   Ø
          DCA S      /STORE ADDRESS OF STRING
          TAD I TYPLOT          /GET CONTROL CHR
          AND L77    /MASK FOR YZ (ABOVE)
          CIA
          DCA I RSCLML          /-YZ => SCLMLT
          TAD I TYPLOT
          RTR
          RTR
          RTR
          AND L7     /MASK FOR X
          DCA I RTRANS          /X => TRANSQ
          TAD I TYPLOT
          RTL
          RTL
          AND L7
          DCA I RLINRU          /W => LINRUN
          ISZ TYPLOT /TO EXIT AFTER CONTROL CHR
          TAD S
          JMS I RPLTST          /CALL PLTSTG
          JMP I TYPLOT          /EXIT

RSCLML,   SCLMLT
RTRANS,   TRANSQ
RLINRU,   LINRUN
RPLTST,   PLTSTG

L77,      77
L7,       7
S,        Ø
```

```
/MOVE & SPC    MOVE MAKES 1Ø PEN MOVEMENTS FOR EACH
/CHARACTER, ACCORDING  TO CODE AT ADDRESS "L" THROUGH
/"L+4".   SPC MOVES PEN TO (5,Ø) OR (Ø,7) ACCORDING TO
/LINRUN.
/
MOVE,     Ø
          TAD         MMM5       /SET WORD COUNTER
          DCA         MMM5GO
          DCA         OLDRX      /ZERO PREVIOUS X-MVMT
TIM1,     TAD         MMM2       /SET MVMT COUNTER  2/WORD
          DCA         MMM2GO
          TAD I       L
          DCA         M          /CONTAINS TWO MOVEMENTS
TIM2,     TAD         M
          AND         LLL7       /MASK  4TH BITE
          DCA         RAWY
          TAD         M
          RTR
          RAR                    /BITE3 =>  BITE4
          AND         LLL7       /MASK AGAIN
          TAD         MMM5       /6,7 - 5 > Ø
          SMA         /IS IT A PNUPDN  COMMAND (EG., 6 OR7)
          JMS I       RPNUPD     /YES, CALL PNUPDN
          TAD         LLL5       /RESTORE RAWX
          DCA         RAWX       /X-COORDINATE OF MVT
          TAD         RAWX       /FOR PNUPDN
          DCA         OLDRX
          JMS         SCLPLT     /TO TRANSFORM, SCALE, & PLOT

          JMS         PLOTIT     /PLOT IT!
          CLA
          TAD         M
          RTR
          RTR
          RTR         /BITES 1&2  =>  BITES 3&4
          DCA         M
          ISZ         MMM2GO     /GONE THRU TWICE?
          JMP         TIM2       /NO, DO  IT AGAIN.
          ISZ         MMM5GO     /YES, HAVE WE DONE 5 L'S?
          SKP
          JMP         SPC        /YES, SPACE AND EXIT
          ISZ         L          /NO, INCR L & AWAY AGAIN.
          JMP         TIM1
```

```
/ROUTINE TO RAISE PEN & MOVE TO (5,0) OR (0,7).
SPC,        JMS I       RPNUPD      /LIFT PEN
            CLA
            TAD         LINRUN      /=0?
            SNA CLA
            JMP         .+5
            TAD         LLL7        /LINRUN = 1
            DCA         RAWY
            DCA         RAWX
            JMP         .+4
            TAD         LLL5        /LINRUN = 0
            DCA         RAWX
            DCA         RAWY
            JMS         SCLPLT
            JMS         PLOTIT      /SCALE & PLOT THE SPACE
            JMP I       MOVE        /EXIT

RAWX,0
RAWY,0
SCLMLT,0
TRANSQ,0
LINRUN,0
OLDRX,0
L,0
M,0

LLL5,       5
LLL7,       7
MMM5,       -5
MMM2,       -2

MMM2GO,0
MMM5GO,0
CNTR,0

RPNUPD,     PNUPDN


/PLOTIT     ROUTINE TO CALL PLOTX (DIGITAL 8-12-U) WITH
/PROPER PARAMETERS.
PLOTIT,     0
PNSTAT,     CLA IAC     /PEN UP OR DOWN?
            JMS I PLOT
XMVT,       0
YMVT,       0
            JMP I PLOTIT
PLOT,       PLOTX
```

```
/SCLPLT    ROUTINE TO TRANSFORM ACCORDING TO TRANSQ
/AND SCALE ACCORDING TO SCLMLT EACH MOVEMENT.  INPUT IS
/RAWX, RAWY; OUTPUT INTO XMVT, YMVT OF PLOTIT.
SCLPLT,   Ø
          TAD        TRANSQ     /ORIENTATION OF CHRS
          SNA
          JMP        STRAIT     /TRANSQ=Ø; NO TRANSFORM
          TAD        MMM2
          SNA
          JMP        ROTRIT     /TRANSQ=2; ROTATE 9Ø DEG. RIGHT
          SPA
          JMP        INVRT      /TRANSQ=1; INVERT EACH CHR
                                /TRANSQ=3; ROTATE LEFT

ROTLFT,   CLA        /TO ROTATE EACH CHR LEFT 9Ø DEGREES
          TAD        RAWX
          DCA        TRY        /RAWX =>  TRY
          TAD        RAWY
          CIA
          TAD        LLL7
          DCA        TRX        /7 - RAWY => TRX
          JMP        SCALE

ROTRIT,   TAD        RAWX       /TO ROTATE EACG CHR RIGHT 9Ø DEG.
          CIA
          TAD        LLL5
          DCA        TRY        /5-RAWX => TRY
          TAD        RAWY
          DCA        TRX        /RAWY => TRX
          JMP        SCALE

INVRT,    TAD        RAWX       /TO INVERT EACH CHR
          CIA
          TAD        LLL5
          DCA        TRX        /5-RAWX =>  TRX
          TAD        RAWY
          CIA
          TAD        LLL7
          DCA        TRY        /7-RAWY => TRY
          JMP        SCALE

STRAIT,   TAD        RAWX       /NO TRANSFORM
          DCA        TRX
          TAD        RAWY
          DCA        TRY        /RAWX,Y => TRX,Y
```

```
SCALE,    TAD        SCLMLT
          DCA        CNTR
          TAD        TRX
          ISZ        CNTR       /TAD LOOP TO MULTIPLY
          JMP        .-2
          DCA        XMVT
          TAD        SCLMLT
          DCA        CNTR
          TAD        TRY
          ISZ        CNTR
          JMP        .-2
          DCA        YMVT

          JMP I      SCLPLT     /EXIT

TRX=XMVT
TRY=YMVT
```

```
/PLTSTG    ROUTINE TO PLOT A STRING OF CHARACTERS ACCORDING
/TO CONTROL CHR SET BY TYPLOT.  ENTER WITH ADDRESS OF STRING
/IN AC.  EXITS WITH THE CURRENT VALUES OF PLTNX,Y (DIGITAL
/8-12-U) RELATIVE TO THOSE VALUES AT CALL OF PLTSTG.
/STRING IS PACKED TWO 6-BIT TRIMMED ASCII CHARACTERS TO A
/WORD, WITH 'ØØ' SERVING AS THE TERMINATOR.
/
*MOVE+2ØØ
/
PLTSTG,   Ø
          DCA STRADD
          TAD I OLDX  /=TAD PLTNX - CURRENT X-COORDINATE
          DCA COLDX  /SAVE IT
          TAD I OLDY
          DCA COLDY
RDPACK,   TAD MM2
          DCA MM2GO  /SET PASS COUNTER
          TAD I STRADD
          DCA SAV
RDPK2,    TAD SAV
          RTR; RTR; RTR
RDPK3,    AND LL77   /MASK
          SNA        /ØØ TERMINATES THE STRING
          JMP TYPFIN
```

```
/CHRTR    ROUTINE TO DETERMINE CHARACTER SET
CHRTR,    TAD MM4Ø
          SNA         /IS IT A SPACE (=4Ø)?
          TAD MM5     /YES, SUBRATCT 5
          SMA
          TAD MM1Ø    /CHR > 44: SUBTRACT 1Ø
          TAD LL4Ø    /RESTORE


/BASAD    ROUTINE TO MULTIPLY CODE BY 5 & ADD TO BASE
/TO DEVELOP THE ADDRESS OF MOVEMENT CODE.

BASAD,    DCA CODE
          TAD MM5
          DCA CNTR2
          TAD CODE
          ISZ CNTR2
          JMP .-2
          TAD BASE
          DCA I RL    /DCA L NEXT PAGE



/UPDATE    ROUTINE TO PUT SCALED (Ø,Ø) INTO PLTNX,Y;
/CALL 'MOVE'; AND UPDATE COLDX,Y
/
UPDATE,   JMS SCLØØ  /GET SCALED & TRANSFORMED (Ø,Ø)
          DCA I OLDX    /PUT XØ INTO PLTNX
          TAD I RYMVT
          DCA I OLDY /SAME WITH YØ
          JMS I RMOVE   /CALL "MOVE"
          JMS SCLØØ
          CIA          /- XØ
          TAD I OLDX /+ PLTNX (CURRENT X)  = DELTA X
          TAD COLDX  /+ PREVIOUS X COORDINATE
          DCA COLDX  /UPDATED
          TAD I RYMVT  /YØ
          CIA           /- YØ
          TAD I OLDY /+ PLTNY  =  DELTA Y
          TAD COLDY
          DCA COLDY  /UPDATED
```

```
/RDPK4 & TYPFIN

RDPK4,    TAD SAV
          ISZ MM2GO   /2 TIMES THRU?
          JMP RDPK3   /NO, GET 2ND PACKED CHARACTER
          CLA
          ISZ STRADD  /INCREMENT STRING ADDRESS
          JMP RDPACK  /GET NEXT WORD IN STRING


TYPFIN,   CLA
          TAD COLDX  /CURRENT X-COORDINATE REL TO CALL
          DCA I OLDX /PUT CURRENT X INTO PLTNX, REL.
                     /TO PLTNX AT CALL OF PLTSTG.
          TAD COLDY
          DCA I OLDY /SAME WITH Y-COORD.
          JMP I PLTSTG          /EXIT


/PNUPDN    ROUTINE TO LOWER PEN IF C(AC)=2; RAISE
/PEN OTHERWISE.  EXITS WITH PREVIOUS X-COORD (OLDRX)
/IN AC.
PNUPDN,   0
          TAD MM2
          SZA CLA
          JMP .+4
          TAD DN
          DCA I RPNSTA          /LOWER PEN
          JMP .+3
          TAD UP
          DCA I RPNSTA
          TAD I ROLDRX
          TAD MM5
          JMP I PNUPDN          /EXIT

DN,       CLA
UP,       CLA IAC
RPNSTA,   PNSTAT
ROLDRX,   OLDRX
```

```
/ROUTINE TO TRANSFORM & SCALE (Ø,Ø).  EXITS WITH XØ
/IN AC; YØ IS IN YMVT OF PLOTIT
SCLØØ,    Ø
          CLA
          DCA I RRAWX           /ZERO RAWX,Y
          DCA I RRAWY
          JMS I RSCLPL
          TAD I RXMVT
          JMP I SCLØØ   /EXIT

RRAWX, RAWX
RRAWY, RAWY
RSCLPL, SCLPLT
RXMVT, XMVT


/VARIABLES FOR PLTSTG

STRADD,Ø
MM2GO,Ø
SAV,Ø
CODE,Ø
CNTR2,Ø

MM2,      -2
LL77,     77
MM5,      -5

COLDX,Ø
COLDY,Ø
OLDX,     PLOTX+162
OLDY,     PLOTX+163
RL,L
RYMVT,    YMVT
RMOVE,    MOVE

BASE,     .-1
MM4Ø,-4Ø
MM1Ø,-1Ø
LL4Ø,4Ø
```

```
/
/MOVEMENT CODES FOR EACH CHARACTER
/
1370; 1333; 4026; 4040; 4040     /A
4676; 3343; 3303; 4042; 4000
4660; 1676; 0105; 4010; 4040
3676; 4145; 0030; 0000; 0000
4676; 0306; 0333; 4000; 4040
4676; 0306; 0333; 0000; 0000
4660; 1676; 0105; 4010; 2242    /G
0376; 4643; 4040; 4040; 4040
7010; 2030; 1626; 3636; 3636
0072; 3630; 4626; 4646; 4646     /J
0276; 4013; 4613; 4646; 4646     /K
0076; 4040; 4040; 4040; 4040     /L
2376; 4046; 4040; 4040; 4040     /M
4076; 4646; 4646; 4646; 4646     /N
7561; 3616; 4145; 1030; 0501     /O
4676; 0343; 4643; 0006; 0000     /P
1645; 4146; 0030; 2262; 4072     /Q
4676; 0343; 4013; 4040; 4040     /R
3070; 4241; 1333; 0504; 4616     /S
7620; 0646; 0606; 0606; 0606     /T
7601; 1001; 4130; 4646; 4646     /U
7606; 4620; 4646; 4646; 4646     /V
7606; 2310; 4630; 4646; 4646     /W
4670; 0666; 4076; 4040; 4040     /X
7320; 2306; 4646; 4646; 4646     /Y
7606; 0046; 4040; 4040; 4040     /Z

0000; 0000; 0000; 0000; 0000     /SPACE
```

```
7521; 3424; 1314; 3233; 1212     /$
7525; 1514; 1135; 3231; 2121     /%
7521; 3424; 2314; 3212; 3232     /&
7323; 2615; 2335; 2323; 2323     /'
7636; 1325; 3021; 3030; 3030     /(
7010; 3321; 1625; 1616; 1616     /)
7521; 1423; 2332; 3412; 2323     /*
7521; 0323; 2343; 2121; 2121     /+
7211; 2122; 2111; 1010; 1010     /,
7313; 6333; 3333; 3333; 3333     /-
7110; 2021; 1110; 0464; 0404     /.
4670; 4066; 4040; 4040; 4040     /SLASH(/)
7501; 3616; 4145; 1030; 4501     /0
7010; 2030; 1526; 2626; 2626     /1
7505; 3616; 4445; 0001; 4140     /2
7606; 1346; 4334; 3041; 0110     /3
7630; 4333; 0603; 0606; 0606     /4
7646; 0306; 3414; 4143; 0030     /5
1372; 4233; 3041; 0500; 3616     /6
7505; 4616; 3222; 2212; 1010     /7
4472; 0646; 4204; 0040; 0000     /8
7540; 1636; 0405; 3313; 4544     /9
7211; 2122; 6411; 2575; 1424     /:
7210; 2122; 6410; 2575; 1424     /;
7545; 4103; 4141; 4141; 4141     /<
7202; 6342; 0373; 0303; 0303     /=
7101; 0543; 0505; 0505; 0505     />
7505; 3616; 4445; 2123; 7060     /?
```