ORNL/TM-6918

User and System Considerations for the TCSTEK Software Library

W. H. Gray

OAK RIDGE NATIONAL LABORATORY
OPERATED BY UNION CARBIDE CORPORATION · FOR THE DEPARTMENT OF ENERGY

DISCLAIMER

Printed in the United States of America. Available from
National Technical Information Service
U.S. Department of Commerce
5285 Port Royal Road, Springfield, Virginia 22161
Price: Printed Copy $4.50; Microfiche $3.00

ORNL/TM-6918
Dist. Category UC-20

Contract No. W-7405-eng-26

FUSION ENERGY DIVISION

USER AND SYSTEM CONSIDERATIONS FOR THE TCSTEK SOFTWARE LIBRARY

W. H. Gray

Date Published - August 1979

NOTICE This document contains information of a preliminary nature. It is subject to revision or correction and therefore does not represent a final report.

Prepared by the
OAK RIDGE NATIONAL LABORATORY
Oak Ridge, Tennessee 37830
operated by
UNION CARBIDE CORPORATION
for the
DEPARTMENT OF ENERGY

THIS PAGE

WAS INTENTIONALLY

LEFT BLANK

CONTENTS

THIS PAGE

WAS INTENTIONALLY

LEFT BLANK

PREFACE

This work was sponsored by the ORNL Fusion Energy Division Large Coil Program and was done in conjunction with system development on the GIFTS 4B[1] structural analysis system.

THIS PAGE

WAS INTENTIONALLY

LEFT BLANK

ABSTRACT

This report documents the idiosyncrasies of the Tektronix PLOT 10 Terminal Control System level 3.3 software as it currently exists on the ORNL Fusion Energy Division DECsystem-10 computer. It is intended to serve as a reference for future Terminal Control System updates in order that continuity between releases of Terminal Control System PLOT 10 software may be maintained.

1. INTRODUCTION

This report documents the implementation of the Tektronix PLOT 10 Terminal Control System (TCS) software into the Fusion Energy Division (FED) DECsystem-10 computer. This report supersedes all previous reports on this subject and specifically applies to TCS level 3.3 software. TCS level 3.3 documentation may be found in Ref. 2, and a system guide may be found in Ref. 3.

This report pertains to the software library on REL: called TCSTEK.REL and is divided into the following sections:

a) In Sect. 2 the reader may find out how to access the TCSTEK software library.

b) Section 3 discusses several of the system considerations which prompted the writing of this document.

c) Section 4 presents the locally written character manipulating software for TCS.

d) Section 5 presents the locally written terminal I/O software for TCS and specifically enumerates its advantages and disadvantages.

e) Section 6 discusses the local modifications to the TCS software — in particular the program which sets the transmit speed code for all teletypes connected to the FED DC10 communications interface and the corresponding subroutine modifications in the TCS source program necessary to access the transmit speed code.

If the reader of this report is uninterested in the system implementation of these subroutines, then probably the only section which should be required reading is the description of the subroutine ADEIN (see Sect. 5.1). This subroutine is called every time the user

requests terminal input through TCS and is of particular interest if the user wishes to use the graphic cross-hair cursor as an input mechanism. Otherwise, all sections of this report are intended for readers who want further information about the locally written system structure of TCS.

2. ACCESSING THE TCSTEK SOFTWARE LIBRARY

The TCSTEK library's relocatable object module (.REL file), which for loading efficiency has been indexed by program MAKLIB, may be found on REL:TCSTEK.REL. As a user convenience, an indirect command file (.CMD file) has been created on device PUB: to simplify use of the Tektronix terminal control software. An example of accessing the TCSTEK library follows:

```
.EXECUTE yourpg,@PUB:TCSTEK
```

and the TCSTEK software library will be loaded in the library search mode in order that the global external references made by yourpg will be defined.

3. TCS SYSTEM SOFTWARE

The TCS level 3.3 software has been implemented without any user-discernible differences between it and the vendor-supplied installation instructions,[4] as it successfully passes the vendor test program.[5] Sections 4 and 5 present the six subroutines, KA12AS, KAM2AS, KAS2A1, KAS2AM, ADEIN, and ADEOUT, which are required by TCS in order for it to perform correctly.

There are two major differences between the TCS software as received and that which is available in relocatable object module form. They are:

a) The call to subroutine IOWAIT has been replaced with a call to subroutine SLP in subroutines HDCOPY, ERASE, and NEWPAG.

b) Subroutine INITT has been modified to attempt to access the transmission speed which is available in the monitor table for any teletype line number.

The reasons for these modifications are discussed below.

Subroutine IOWAIT's function is to transmit synchronous (SYN) ASCII characters (ASCII character code 37_8; numbers subscripted by 8 in this report refer to octal, or base 8, notation) to the terminal to prevent loss of output while the terminal is busy doing a previous instruction, such as a page erase. To perform this task it multiplies the ITIME (the only argument to subroutine IOWAIT) by the transmission baud rate divided by 10 and uses this number as the number of SYN characters to transmit to the terminal. (The SYN ASCII character is a no-op instruction for the terminal.) This method is at best inaccurate and at worst a tremendous unnecessary overhead on the system. Consider, for example, a call to IOWAIT with ITIME equal to 180 (such a call used to be made from HDCOPY). If a user is transmitting at 960 characters per second, then 180 * 960/10 or 17,280 SYN characters were sent to the terminal — clearly a wasteful operation. By empirical determination it was found that a pause of 2 sec was sufficient for terminals during a page erase and 7 sec was sufficient for terminals during a hardcopy operation. Therefore, in subroutines NEWPAG and ERASE the call to IOWAIT was replaced with a call to SLP with an argument of 2 sec, and in

subroutine HDCOPY the call to IOWAIT was replaced with a call to SLP with an argument of 7 sec. Also, before the call to SLP in the above three subroutines, a call to the TCS subroutine TSEND is included to ensure beginning the terminal operation at the prescribed time. (Subroutine TSEND forces out to the terminal the internal TCS buffer.)

In order that the TCS software be initialized, a call to subroutine INITT with a single argument representing the terminal transmission baud rate (number of characters per second) is required. However, the transmission speed is known by the computer; how else could the computer communicate with a teletype? Therefore, a modification was made to subroutine INITT to have it call a function subprogram named ITRMBD. Subroutine ITRMBD accepts one argument, the job number, and it returns the value of the transmission baud rate. This frees the user from having to worry about this value, especially since an application program usually is executed from many terminals with different transmission speeds. The user need only call INITT with an argument of 0 to have the transmission speed determined by subprogram ITRMBD.

The above solution did not completely solve the problem of TCS internally determining transmission speeds. Numerous teletype lines are hardwired into the FED DC10 communications interface, which does not have the hardware/software to set transmission speed codes in the monitor table for a job connected to it. To alleviate this difficulty a program (SETSPD) was written to software-set the transmission speed codes of all the hardwired lines connected to the DC10. The program is run from the system AUTO file when the system is initialized. This program and the other subprograms mentioned in this section are discussed further in Sect. 6.

4. TCS CHARACTER MANIPULATION SOFTWARE

This section describes four machine-dependent subroutines for translating ASCII character strings required by the TCS software. Their names are constructs of the following mnemonics: convert, K; left-justified ASCII one character per word, A1; to, 2; right-justified ASCII one character per word, AS; left-justified ASCII five characters per word, AM. Therefore, to convert a left-justified ASCII text string to a right-justified array of characters in ASCII decimal equivalent (ADE) format, the appropriate subroutine to call would be KAM2AS. The next four subsections describe the argument list and calling sequence of subroutines KA12AS, KAM2AS, KAS2A1, and KAS2AM.

4.1 SUBROUTINE KA12AS

Subroutine KA12AS performs to Tektronix Terminal Control System specifications and may be used to convert left-justified ASCII characters (stored one character per word) into right-justified, zero-filled characters (stored one character per word).

Calling sequence:

```
CALL KA12AS (NCHAR,KA1,KADE)
```

Argument list:

NCHAR is the number of characters to be translated (unmodified by this subroutine).

KA1 is the left-justified ASCII character array stored one character per word (unmodified by this subroutine).

KADE is the right-justified, zero-filled ASCII character array stored one character per word (returned by this subroutine).

Required subroutines: None

Language: MACRO-10 (see Ref. 6)

Subprogram listing:

```
        TITLE   KA12AS

T0=0
T1=1
T2=2
L=16
P=17

        ENTRY   KA12AS

SIXBIT  /KA12AS/
KA12AS: PUSH    P,T2            ;SAVE T2
        HRRZ    T2,@0(L)        ;GET NUMBER TO TRANSLATE
        JUMPE   T2,BYEBYE       ;IS IT 0
        MOVN    T2,T2           ;NO, MAKE IT NEGATIVE
        HRL     T2,T2           ;IN LEFT HALF
        HRRI    T2,@2(L)        ;NOW PUT IN THE OUTPUT ADDRESS
        HRRZI   T1,@1(L)        ;GET ADDRESS OF A1 CHARACTERS
LOOP:   MOVE    T0,(T1)         ;GET AN A1 CHARACTER
        LSH     T0,-^D29        ;MAKE IT ADE
        MOVEM   T0,(T2)         ;PUT THE ADE CHARACTER AWAY
        AOJ     T1,             ;INCREMENT ADDRESS FOR A1
        AOBJN   T2,LOOP         ;ARE WE FINISHED
        POP     P,T2            ;RESTORE T2
BYEBYE: POPJ    P,              ;WE ARE DONE
        END
```

4.2 SUBROUTINE KAM2AS

Subroutine KAM2AS performs to Tektronix Terminal Control System specifications and may be used to convert left-justified ASCII characters (stored five characters per word) into right-justified, zero-filled characters (stored one character per word).

Calling sequence:

CALL KAM2AS (NCHAR,KAM,KADE)

Argument list:

NCHAR is the number of characters to be translated (unmodified by this subroutine).

KAM is the left-justified ASCII character array stored five characters per word (unmodified by this subroutine).

KADE is the right-justified, zero-filled ASCII character array stored one character per word (returned by this subroutine).

Required subroutines: None

Language: MACRO-10

Subprogram listing:

```
        TITLE   KAM2AS

T0=0
T1=1
T2=2
L=16
P=17

        ENTRY   KAM2AS

SIXBIT  /KAM2AS/
KAM2AS: PUSH    P,T2            ;SAVE T2
        HRRZ    T2,@0(L)        ;GET NUMBER TO TRANSLATE
        JUMPE   T2,BYEBYE       ;IS IT 0
        MOVN    T2,T2           ;NO, MAKE IT NEGATIVE
        HRL     T2,T2           ;IN LEFT HALF
        HRRI    T2,@2(L)        ;NOW PUT IN THE OUTPUT ADDRESS
        HRRZI   T1,@1(L)        ;GET ADDRESS OF AM CHARACTERS
        HRLI    T1,440700       ;GET A GOOD BYTE POINTER
LOOP:   ILDB    T0,T1           ;GET AN AM CHARACTER
        MOVEM   T0,(T2)         ;PUT THE ADE CHARACTER AWAY
        AOBJN   T2,LOOP         ;ARE WE FINISHED
        POP     P,T2            ;RESTORE T2
BYEBYE: POPJ    P,              ;WE ARE DONE
        END
```

4.3 SUBROUTINE KAS2A1

Subroutine KAS2A1 performs to Tektronix Terminal Control System specifications and may be used to convert right-justified ASCII characters (stored one zero-filled character per word) into left-justified, blank-filled characters (stored one character per word).

Calling sequence:

CALL KAS2A1 (NCHAR,KADE,KA1)

Argument list:

NCHAR is the number of characters to be translated (unmodified by this subroutine).

KADE is the right-justified, zero-filled ASCII character array stored one character per word (unmodified by this subroutine).

KA1 is the left-justified ASCII character array stored one character per word (returned by this subroutine).

Required subroutines: None

Language: MACRO-10

Subprogram listing:

```
        TITLE   KAS2A1

T0=0
T1=1
T2=2
L=16
P=17

        ENTRY   KAS2A1

SIXBIT  /KAS2A1/
KAS2A1: PUSH    P,T2            ;SAVE T2
        HRRZ    T2,@0(L)        ;GET NUMBER TO TRANSLATE
        JUMPE   T2,BYEBYE       ;IS IT 0
        MOVN    T2,T2           ;NO, MAKE IT NEGATIVE
        HRL     T2,T2           ;IN LEFT HALF
        HRRI    T2,@2(L)        ;NOW PUT IN THE OUTPUT ADDRESS
        HRRZI   T1,@1(L)        ;GET ADDRESS OF ADE CHARACTERS
LOOP:   MOVE    T0,(T1)         ;GET AN ADE CHARACTER
```

```
        LSH     T0,^D29         ;MAKE IT A1
        MOVEM   T0,(T2)         ;PUT THE A1 CHARACTER AWAY
        AOJ     T1,             ;INCREMENT ADDRESS FOR ADE
        AOBJN   T2,LOOP         ;ARE WE FINISHED
        POP     P,T2            ;RESTORE T2
BYEBYE: POPJ    P,              ;WE ARE DONE
        END
```

4.4 SUBROUTINE KAS2AM

Subroutine KAS2AM performs to Tektronix Terminal Control System specifications and may be used to convert right-justified ASCII characters (stored one zero-filled character per word) into left-justified, blank-filled characters (stored one character per word).

Calling sequence:

CALL KAS2AM (NCHAR,KADE,KAM)

Argument list:

NCHAR is the number of characters to be translated (unmodified by this subroutine).

KADE is the right-justified ASCII character array stored one character per word (unmodified by this subroutine).

KAM is the left-justified, blank-filled ASCII character array stored five characters per word (returned by this subroutine).

Required subroutines: None

Language: MACRO-10

Subprogram listing:

```
        TITLE   KAS2AM

T0=0
T1=1
T2=2
L=16
P=17

        ENTRY   KAS2AM

SIXBIT  /KAS2AM/
KAS2AM: PUSH    P,T2            ;SAVE T2
        HRRZ    T2,@0(L)        ;GET NUMBER TO TRANSLATE
        JUMPE   T2,BYEBYE       ;IS IT 0
        MOVN    T2,T2           ;NO, MAKE IT NEGATIVE
        HRL     T2,T2           ;IN LEFT HALF
        HRRI    T2,@1(L)        ;NOW PUT IN THE OUTPUT ADDRESS
        HRRZI   T1,@2(L)        ;GET ADDRESS OF AM CHARACTERS
        HRLI    T1,440700       ;GET A GOOD BYTE POINTER
LOOP:   MOVE    T0,(T2)         ;GET THE ADE CHARACTER AWAY
        IDPB    T0,T1           ;PUT INTO AM
        AOBJN   T2,LOOP         ;ARE WE FINISHED
        POP     P,T2            ;RESTORE T2
BYEBYE: POPJ    P,              ;WE ARE DONE
        END
```

5. TCS I/O SOFTWARE

This section describes two subroutines for character transmission between a user application program and a Tektronix terminal. They are subroutines ADEIN and ADEOUT, which mnemonically represent ASCII decimal equivalent input and output, respectively. Both subroutines are written in MACRO-10; therefore, the TCS software library does not require the FORTRAN REAL TIME OPERATING SYSTEM[7] (FOROTS) to be in the user's high segment during execution.

5.1 SUBROUTINE ADEIN

Subroutine ADEIN performs to Tektronix Terminal Control System specifications and may be used to input right-justified, zero-filled ASCII characters from the user's terminal. The complete ASCII character set may be transmitted with this subroutine; however, there are certain characters which have a special meaning. These special characters are discussed below.

a) ^U If a ^U (ASCII character code 25_8; numbers which are subscripted by 8 in this report refer to octal, or base 8, notation) is typed, the input line is erased and the input may be started over.

b) ^R If a ^R (ASCII character code 22_8) is typed, the current line is retyped to the user's terminal for visual inspection of mistakes. The contents of the input buffer are not altered by this character.

c) (DEL) If a DEL (ASCII character code 177_8) is typed, the previously input character is thrown away and the terminal input line is backspaced one character.

d) (BS) If a backspace (ASCII character code 10_8) is typed, it has the same effect as typing a DEL character.

e) (CR) If a carriage return (ASCII character code 15_8) is typed, it is ignored. (Remember on most terminals the carriage return key actually transmits two characters to a host computer — both a carriage return and a line feed.)

f) (LF) If a line feed (ASCII character code 12_8) is typed, input is terminated without inserting the line feed character into the input buffer, and program control is passed back to the calling subroutine.

Remember — a maximum of 72 characters may be input by this subroutine during any single call.

5.1.1 Further Considerations

Tektronix 4010 series models have the capability of being used in the graphic input mode (GIN). When an ESC SUB character sequence (ASCII character codes 33_8 and 32_8, respectively) is transmitted to a terminal, the terminal's graphic cross-hair cursor will be turned on as a prompt for GIN mode input. By positioning the graphic cursor (using the thumb wheels located on the terminal) to a user-selected screen position and then striking a character, both the character (in ADE format) and four more characters are transmitted to subroutine ADEIN. The last four characters are decoded by the TCS software to locate the screen position of the graphic cursor.

Subroutine ADEIN is written to accept five and only five characters when the GIN mode switch is set; therefore, only a single character need be typed to transmit all the graphic cursor coordinate position

information to the host computer. The other four characters are generated internally by the Tektronix terminal. This differs from previous implementations of the TCS software for the FED DECsystem-10, but it reflects the author's preference of execution.

One word of caution: do not strike the carriage return key while in the GIN mode. This will confuse the TCS software, since it expects to receive five characters instead of six under this circumstance. If a carriage return is struck during GIN mode input, unpredictable results will occur.

Calling sequence:

```
CALL ADEIN (NCHAR,KADE)
```

Argument list:

NCHAR is the number of characters to be translated (unmodified by this subroutine).

KADE is the right-justified ASCII character array stored one character per word (unmodified by this subroutine).

Required subroutines: None

Language: MACRO-10

Subprogram listing:

```
        TITLE   ADEIN
        ENTRY   ADEIN

T0=0
T1=1
T2=2
L=16
P=17

        .COMMON  TKTRNX[^D60]       ;ADDRESS OF TKTRNX COMMON

KGNMOD=TKTRNX+25                    ;ADDRESS OF GIN MODE FLAG

SIXBIT  /ADEIN/
ADEIN:
        PUSH    P,T2                ;SAVE T2
BEGIN:
        HRREI   T1,-^D72            ;WE NEED 72 ADE CHARACTERS
        SKIPE   KGNMOD              ;UNLESS GIN MODE INPUT
        HRREI   T1,-5               ;IN WHICH CASE WE ONLY WANT 5
        HRRZI   T2,@1(L)            ;ADDRESS OF IARRAY
        SETZM   @0(L)               ;ZERO OUT NCHAR
LOOP:   INCHRW  T0                  ;GET A CHARACTER
        SKIPE   KGNMOD              ;ARE WE IN GIN MODE
          JRST  NOCHK               ;THEN THIS COULD BE A COORDINATE
        CAIN    T0,25               ;IS IT A ^U
          JRST  CONTLU              ;YES, START OVER
        CAIN    T0,22               ;IS IT A ^R
          JRST  CONTLR              ;YES, TYPE OUT
        CAIN    T0,177              ;IS IT A RUBOUT
          JRST  RUBOUT              ;YES, GO PROCESS A RUBOUT
        CAIN    T0,10               ;IS IT A BACKSPACE
          JRST  RUBOUT              ;YES, TREAT AS IF RUBOUT
        CAIN    T0,15               ;IS IT A CARRIAGE RETURN
          JRST  LOOP                ;YES,IGNORE IT
        CAIN    T0,12               ;IS IT A LINE FEED
          JRST  LF                  ;THEN INPUT FOR CALL IS DONE
NOCHK:
        MOVEM   T0,(T2)             ;STORE CHARACTER
        AOS     @0(L)               ;BUMP NCHAR
        AOJ     T2,                 ;INCREMENT ADDRESS
          JRST  SKIP                ;GO TO THE END OF THE LOOP
CONTLU:
        OUTSTR  CRLF                ;OUTPUT A CR-LF
          JRST  BEGIN               ;START OVER
CONTLR:
        PUSH    P,T1                ;SAVE T1
        OUTSTR  CRLF                ;OUTPUT A CR-LF
        MOVN    T1,@0(L)            ;GET NEGATIVE NUMBER OF CHARS.
        HRL     T1,T1               ;IN LEFT HALF
        HRRI    T1,@1(L)            ;GET ADDRESS OF CHARS.
LOP:    MOVE    T0,(T1)             ;GET A CHARACTER
        IONEOU  T0                  ;OUTPUT IT
```

```
        AOBJN   T1,LOP          ;ARE WE FINISHED
        POP     P,T1            ;RESTORE T1
          JRST  SKIP
RUBOUT:
        SKIPG   @0(L)           ;HAVE WE GOT ANY CHARACTERS YET
          JRST  LOOP            ;NO, THEN GO BACK TO LOOP
        SOJ     T1,             ;DECREMENT COUNTER
        SOJ     T2,             ;DECREMENT ADDRESS
        SOS     @0(L)           ;DECREMENT NCHAR
        IONEOU  [10]            ;BACKSPACE TERMINAL
SKIP:
        AOJL    T1,LOOP         ;ARE WE FINISHED
        SKIPE   KGNMOD          ;IS GIN MODE SET
        CLRBFI                  ;YES, THEN CLEAR INPUT BUFFER
LF:
BYEBYE:
        POP     P,T2            ;RESTORE T2
        POPJ    P,              ;RETURN
CRLF:   ASCIZ/
/
        END
```

5.2 SUBROUTINE ADEOUT

Subroutine ADEOUT performs to Tektronix Terminal Control System specifications and may be used to output right-justified, zero-filled ASCII characters to the user's terminal. The complete ASCII character set may be transmitted with this subroutine because it uses the IONEOU machine instruction for output of a single Packed Image Mode (PIM) character. This instruction bypasses several monitor level checks; e.g., after every 72 characters, the monitor automatically inserts a carriage return-line feed sequence into the output stream. It is, therefore, an efficient way to transmit characters.

Calling sequence:

CALL ADEOUT (NCHAR,KADE)

Argument list:

NCHAR is the number of characters to be translated (unmodified by this subroutine).

KADE is the right-justified ASCII character array stored one

character per word (unmodified by this subroutine).

Required subroutines: None

Language: MACRO-10

Subprogram listing:

```
        TITLE   ADEOUT
        ENTRY   ADEOUT

T0=0
T1=1
L=16
P=17

SIXBIT  /ADEOUT/
ADEOUT:
        MOVN    T1,@0(L)        ;GET NCHAR
        HRL     T1,T1           ;PUT IN LEFT HALF
        HRRI    T1,@1(L)        ;ADDRESS OF ADE CHARACTERS
LOOP:   MOVE    T0,(T1)         ;GET A CHARACTER
        IONEOU  T0,             ;OUTPUT IT
        AOBJN   T1,LOOP         ;ARE WE FINISHED
        POPJ    P,
        END
```

Future considerations:

Since under the 603a monitor buffered PIM I/O appears to be functioning correctly,[8] future system development of the TCS software library should consider replacing this subroutine with a subroutine which uses buffered PIM output.

6. TCS LOCAL SOFTWARE

This section presents two auxiliary subprograms and one program, which complete the TCS system software on the FED DECsystem-10. Program SETSPD is a privileged program which will set the transmission speed codes of teletypes connected to the DC10 communications interface, integer function ITRMBD will return the transmission baud rate for any job number currently active in the system, and subroutine SLP will put the controlling job to sleep for a specified number of seconds.

6.1 SUBROUTINE ITRMBD

Subroutine ITRMBD may be used to return the transmission baud rate for a particular system job number. The baud rate is the number of characters per second being transmitted and received by the terminal. It is approximately the transmission speed in bits per second divided by ten.

Calling sequence:

```
IBAUD=ITRMBD(JOBNM)
```

Argument list:

JOBNM is the system job number for which the baud rate is to be obtained. If JOBMN is equal to -1, then function ITRMBD returns the baud rate for the controlling terminal.

Required subroutines: None

Language: MACRO-10

Subprogram listing:

```
        TITLE ITRMBD

        SEARCH UUOSYM
T0=0
T1=1
L=16
P=17
        ENTRY ITRMBD
SIXBIT  /ITRMBD/
ITRMBD: SETZ    T0,                 ;CLEAR RETURN VALUE
        MOVE    T1,@0(L)            ;GET JOB NUMBER
        TRMNO.  T1,                 ;GET JOBS UDX
          JRST  BYEBYE              ;BUMMER
        MOVEM   T1,UDX              ;STORE IN TRMOP BLOCK
        MOVE    T1,[XWD 2,CODE]     ;FORM TRMOP WORD
        TRMOP.  T1,                 ;DO IT
          JRST  BYEBYE              ;ANOTHER POSSIBLE BUMMER
        MOVE    T0,SPEED(T1)        ;GOOD RETURN LOOKUP CORRECT SPEED
        IDIVI   T0,^D10             ;CONVERT TO BAUD
BYEBYE: POPJ    P,                  ;WE ARE DONE
CODE:   1031                        ;FUNCTION CODE FOR TRANSMIT SPEED
UDX:    Z
SPEED:  0
        ^D50
        ^D75
        ^D110
        ^D134
        ^D150
        ^D200
        ^D300
        ^D600
        ^D1200
        ^D1800
        ^D2400
        ^D4800
        ^D9600
        0
        0
        END
```

6.2 PROGRAM SETSPD

Program SETSPD may be used to set the transmission speed code[9] (TSC) for any terminal connected to the FED computer system. This program is intended to set the TSC of all hardwired terminals connected to the DC10 communications interface, since the DC10 cannot autobaud detect. The table named TABLE located under SUBTTL TTY-TABLE contains the TSC's and TTY line numbers which are set by this program.

Execution:

.R SETSPD

Restrictions:

In order for this program to perform correctly, it must be executed from a privileged project, programmer account.

Language: MACRO-10

Program listing:

```
        TITLE   SETSPD
        SEARCH  UUOSYM


.JBVER=137

VERNO==01       ;MAJOR VERSION NUMBER
VEDIT==2                ;MAJOR EDIT NUMBER
VMINOR==0       ;MINOR EDIT NUMBER
VWHO==06                ;WHG LAST EDITOR

VSTSPD==BYTE (3)VWHO(9)VERNO(6)VMINOR(18)VEDIT

        LOC     .JBVER
        VSTSPD
        RELOC

T1=1
T2=2
T3=3
T4=4
T5=5
T6=6
T7=7
T10=10
T11=11
L=16
```

```
P=17


        SUBTTL   THE-BIG-CODE

DEBUG:
SETSPD: JFCL
        RESET
        MOVE     P,[XWD PDLEN,PDLST]
        MOVEI    T1,NTBL            ;GET NUMBER OF TTY'S TO SET
        JUMPE    T1,BYEBYE          ;ARE THERE ANY?
        MOVE     T3,[XWD 3,SXCODE] ;YES, SET UP TRMOP FUNCTION
        MOVN     T1,T1              ;NEGATE NUMBER OF TTY'S
        HRL      T1,T1              ;IN LEFT HALF
        HRRI     T1,TABLE           ;GET ADDRESS OF TTY TABLE
        OUTSTR   [ASCIZ/
HELLO, MY NAME IS SETSPD AND I AM GOING TO SET THE XMTSPEEDS
OF THE FOLLOWING TERMINALS WHICH ARE CONNECTED TO THE DC10
/]

LOOP:   MOVE     T2,(T1)            ;GET TTY SPEED CODE AND #
        HLRM     T2,SPDCDE          ;STORE SPEED CODE IN TRMOP BLOCK
        HRRZ     T2,T2              ;GET TTY #
        ADDI     T2,.UXTRM          ;MAKE IT A UNIVERSAL DEVICE INDEX
        MOVEM    T2,UDX             ;PLACE IN TRMOP BLOCK
        TRMOP.   T3,                ;SET IT
          JRST   BAD
        OUTSTR   [ASCIZ/TTY/]
        SUBI     T2,.UXTRM          ;GET TTY # AGAIN
        HRL      T2,T2              ;MOVE TO LEFT HALF
        PUSHJ    P,OCTOUT           ;OUTPUT TTY #

SPDOUT:
        HLRZ     T2,(T1)            ;GET SPEED CODE AGAIN
        PUSH     P,T1               ;SAVE T1
        MOVE     T1,SPEED(T2)       ;GET SPEED IN CHARACTERS/SEC
        OUTSTR   [ASCIZ/  TO       /]
        PUSHJ    P,DECOUT           ;TYPE IT OUT
        POP      P,T1               ;RESTORE T1
        OUTSTR   [ASCIZ/
/]
          JRST   CONT
BAD:
        OUTSTR   [ASCIZ/?ERROR - YOU ARE NOT PRIVILEGED TO SET TTY/]
        SUBI     T2,.UXTRM          ;GET TTY NUMBER AGAIN
        HRL      T2,T2              ;MOVE TO LEFT HALF
        PUSHJ    P,OCTOUT           ;OUTPUT TO TTY
        JRST     SPDOUT             ;TYPE OUT SPEED

CONT:
        AOBJN    T1,LOOP

BYEBYE:
        EXIT
```

```
        SUBTTL   SUBROUTINES

OCTOUT:
        MOVE     T6,[POINT 3,T2]    ;POINTER TO T2
        SETZ     T5,                ;SET NONZERO OUT FLAG
        MOVEI    T4,6               ;ONLY 1/2 WORD WORTH
OCTLOP: ILDB     T7,T6              ;GET OCTAL CHARACTER
        SKIPN    T7                 ;NONZERO?
        JUMPE    T5,.+4             ;NO, HOW ABOUT NONZERO FLAG
        ADDI     T7,60              ;MAKE IT ASCII
        OUTCHR   T7                 ;PRINT IT
        SETO     T5,                ;NONZERO FLAG ON
        SOJG     T4,OCTLOP          ;MORE CHARACTERS
        SKIPN    T5                 ;WAS IT 0
        OUTCHR   ["0"]              ;TYPE IT OUT
        POPJ     P,                 ;RETURN

DECOUT:
        IDIVI    T1,12              ;DIVIDE BY 10
        HRLM     T2,(P)             ;SAVE REMAINDER
        SKIPE    0,T1               ;SEE IF ANYTHING LEFT
        PUSHJ    P,DECOUT           ;YES, GO BACK
        HLRZ     T1,(P)             ;GET DIGIT BACK
        ADDI     T1,"0"             ;MAKE IT ASCII
        OUTCHR   T1                 ;TYPE IT
        POPJ     P,                 ;LOOP BACK UNTIL DONE
        SUBTTL   TTY-TABLE

REPEAT  0,<

THE FOLLOWING LIST CONTAINS THE XMTSPEED CODE AND TERMINAL LINE
NUMBER IN HALF WORD FORMAT FOR ALL THE TERMINALS WHOSE XMTSPEEDS
ARE NOT SET BY HARDWARE; I.E., THE LINES CONNECTED TO THE DC10.

LEFT HALF CONTAINS THE XMTSPEED CODE
RIGHT HALF CONTAINS THE TTY #

THE TRANSMISSION CODE IS LISTED BELOW

CODE    CHARACTERS PER SEC
0       0
1       50
2       75
3       110
4       134
5       150
6       200
7       300
10      600
11      1200
12      1800
13      2400
14      4800
15      9600
16      EXTERNAL A
```

```
17      EXTERNAL B
```

FOR EXAMPLE, SUPPOSE THE DC10 IS TRANSMITTING TO TTY1 AT 4800 CPS. THEN THE CORRESPONDING ENTRY IN TABLE WOULD BE

```
        14,,1                           ;TTY1 XMTSPEED OF 4800 CPS

>
TABLE:
        11,,0                           ;TTY0 XMTSPEED OF 1200 CPS.
        15,,2                           ;TTY2 XMTSPEED OF 9600 CPS.
        11,,12                          ;TTY12 XMTSPEED OF 1200 CPS.
        11,,16                          ;TTY16 XMTSPEED OF 1200 CPS.
NTBL=.-TABLE

        SUBTTL  MEMORY
SXCODE: 2031                            ;CODE TO SET SPEED
UDX:    Z
SPDCDE: Z
SPEED:  0
        ^D50
        ^D75
        ^D110
        ^D134
        ^D150
        ^D200
        ^D300
        ^D600
        ^D1200
        ^D1800
        ^D2400
        ^D4800
        ^D9600
        0
        0
PDLST:  0,,PDERR
        BLOCK   100
PDLEN=.-PDLST
PDERR:  ASCIZ/?PDL OVERFLOW
/
        END     SETSPD
```

6.3 SUBROUTINE SLP

Subroutine SLP may be used to put a program to sleep for a specified number of seconds. This subroutine has been substituted for subroutine IOWAIT in the TCS subroutines HDCOPY, NEWPAG, and ERASE. The reason for this substitution was discussed in the system considerations section of this report.

Calling sequence:

```
CALL SLP (ISEC)
```

Argument list:

ISEC is the number of seconds to sleep.

Required subroutines: None

Language: MACRO-10

Subprogram listing:

```
        TITLE   SLP
        ENTRY   SLP

T0=0
L=16
P=17

SIXBIT  /SLP/
SLP:
        MOVE    T0,@0(L)        ;NUMBER OF SECONDS TO SLEEP
        SLEEP   T0,             ;GOOD NIGHT
        POPJ    P,              ;GOOD MORNING
        END
```

7. CONCLUSIONS

Hopefully, this report will sufficiently document the FED system dependencies of the TCS software in order that future updates to this software product will be upwards compatible, thus ensuring system continuity between user application programs.

ACKNOWLEDGMENTS

I would like to express my appreciation to C. L. Thompson for his comments on the internal operation of the TCS software, K. H. Carpenter for his suggestion to use PIM output in subroutine ADEOUT, to R. D. Burris for his patience in helping me debug MACRO subroutines, and to D. W. Swain for his empirical determination of the delay time for a Tektronix hardcopy unit.

REFERENCES

1. H. A. Kamel and M. W. McCabe, "The GIFTS System: Version 4B User's Manual," University of Arizona, Tucson, Arizona (March 1978).

2. "Tektronix PLOT 10 Terminal Control System User Manual," Tektronix, Inc., Beaverton, Oregon (February 1977).

3. "Tektronix PLOT 10 Terminal Control System System Manual," Tektronix, Inc., Beaverton, Oregon (February 1977).

4. "Tektronix PLOT 10 Terminal Control System Installation Guide," Tektronix, Inc., Beaverton, Oregon (February 1977).

5. "Tektronix PLOT 10 Terminal Control System Verification Routine," Tektronix, Inc., Beaverton, Oregon (February 1977).

6. "DECsystem-10 MACRO Assembler Reference Manual," Digital Equipment Corporation, Maynard, Massachusetts (April 1978).

7. Appendix D in "DECsystem-10 FORTRAN Programmer's Reference Manual," Digital Equipment Corporation, Maynard, Massachusetts (June 1977).

8. C. L. Thompson, private communication (April 1979).

9. "DECsystem-10 Monitor Calls Manual," Digital Equipment Corporation, Maynard, Massachusetts (January 1978).

ORNL/TM-6918
Dist. Category UC-20

INTERNAL DISTRIBUTION

1. F. W. Baity
2. J. K. Ballou
3. S. C. Bates
4. R. D. Burris
5. W. D. Cain
6. J. D. Callen
7. K. H. Carpenter
8. L. A. Charlton
9. B. A. Clark
10. D. N. Clark
11. W. K. Dagenhart
12. L. Deleanu
13. K. N. Fischer
14. R. H. Fowler
15. P. Gaffney
16. J. C. Glowienka
17-36. W. H. Gray
37. G. R. Haste
38. T. C. Jernigan
39. P. W. King
40. C. E. Knight, Jr.
41. E. A. Lazarus
42. C. J. Long
43. M. S. Lubell
44. R. W. McGaffey
45. O. B. Morgan
46. J. K. Munro, Jr.
47. A. P. Navarro
48. G. H. Neilson
49. D. R. Overbey
50. C. E. Parker
51. J. K. Penny
52. J. Sheffield
53. K. A. Stewart
54. W. C. T. Stoddart
55. D. W. Swain
56. P. L. Walstrom
57. W. R. Wing
58-59. Laboratory Records Department
60-61. Central Research Library
62. Document Reference Section
63. Laboratory Records, ORNL-RC
64. ORNL Patent Office
65. Fusion Energy Division Library
66. Fusion Energy Division Communications Center

EXTERNAL DISTRIBUTION

67. D. J. Anthony, General Electric Company, Building 23, Room 290, 1 River Road, Schenectady, NY 12345
68. R. W. Conn, Department of Nuclear Engineering, University of Wisconsin, Madison, WI 53706
69. H. K. Forsen, Exxon Nuclear Company, Inc., 777 106th Avenue, N.E., Bellevue, WA 98004
70. H. P. Furth, Princeton Plasma Physics Laboratory, Princeton University, P.O. Box 451, Princeton, NJ 08540
71. R. W. Gould, Building 116-81, California Institute of Technology, Pasadena, CA 91109
72. R. L. Hirsch, Exxon Research and Engineering Company, P.O. Box 101, Florham Park, NJ 07101
73. Office of Assistant Manager, Energy Research and Development, Department of Energy, Oak Ridge Operations Office, Oak Ridge, TN 37830

74-195. Given distribution as shown in TID-4500, Magnetic Fusion Energy (Distribution Category UC-20)