January 1983

INT 111/83

IMPROVEMENTS TO THE LGRAPHICS PACKAGE ON THE TCA TOKAMAK PDP 11-60

J.B. Lister and M.-H. Poget

Centre de Recherches en Physique des Plasmas
Association Euratom - Confédération Suisse
Ecole Polytechnique Fédérale de Lausanne

IMPROVEMENTS TO THE LGRAPHICS PACKAGE ON THE TCA TOKAMAK PDP 11-60

1. Introduction

The LGRAPHICS package (INT 76/79) was written as a high-level set of routines which used either, or both of, the PLOT-10 and Versaplot software packages to control the visual Display Units or Versatek printer/plotter respectively. This total package was fairly large, as is the data retrieval package. The small addressable memory in the PDP 11-60 meant that we had to overlay programs which performed any detailed calculations on whole traces. We have counteracted this problem by rewriting separately the VDU control package and the Versaplot package, with minor restrictions. This led to an increased speed of execution in the case of the old-fashioned coding of PLOT-10, and a reduction in the task-image size. In the case of the Versatek, a considerable reduction in user task-image size was obtained and speed was maintained. The Versatek dump program, on the other hand, was increased in size. We now discuss the two modifications and the restrictions they impose.

2. V D U

The PLOT-10 package is an extremely universal and cumbersome software (20 routines) which spends a great deal of time doing very little. We wrote a very simple routine, HPDRAW, which uses the HP2648A escape sequence commands and the fairly efficient redundant byte rejection algorithm of the HP2648A. The latter avoids sending bytes which repeat themselves, using bits in the "LOW-X" byte to define the rejection. We have no subroutine calls within HPDRAW: which saves time. We have lost the possibility of declaring blank windows and other such subtleties which were never used on TCA. The full screen is used and the aspect-ratio of a VDU drawing is not the same as that of the Versatek drawing, which works in absolute units of cm. The transmission to the terminal is by use of an asynchronous write via a QIO request called in the FORTRAN program. HPDRAW is reproduced in Appendix I.

3. VERSATEK

We have rejected all the Versaplot routines, with the exception of the MTX routine which sends the QIO to the Versatek and is written in MACRO, and the symbol-drawing routines. We have made the major restriction of "one page equals one drawing" which should not in fact be restrictive since this was the only way in which LGRAPHICS was used on TCA. Instead of a MAP file, we therefore have an explicit intermediate file which simply contains the full-page Vector data, in point-line units, and which is extremely simple. The LOW-X and LOW-Y bytes are always sent with the 7 least significant bits of vector data. The eigth bit of each determines whether the most significant byte is sent or implicitly repeated. The most significant bytes contain the pen-up, pen-down information. In addition, we define a "page" coordinate (0,0) and an "end" coordinate (1,0) to complete the definition of the intermediate file. VFILE•BIN is written sequentially in records of 512 Bytes by a synchronous write. In this way we only have a simple user routine, VCREER (see Appendix II), which enormously reduces the task-image file, by in excess of 6 K-Bytes. In addition, the Graphics Library, VTLIB, became so small that we have simply made two object code modules LGRAPH•OBJ and LGRLOG•OBJ. The former contains all the graphics package with the exception of LOGAXE and LCONV which are held in the latter. Task-Build time is thereby also reduced.

The Vector-to-Raster phase is performed by the program RAST (not appended). This program decodes the VFILE•BIN file, page by page, and creates a scratch sliced-vector file, all file operations being asynchronous. This second file contains the start-stop vector information of each slice containing NLINES (74) lines, separately. Each slice is then a separate "drawing" containing Vector information, packed as in VFILE, but with the "local" Y-coordinate. During this preparation phase (in VPREP) the Versatek is not attached. When a page is completed an end-of-page code is written and the next page is prepared until the end-of-data code is met. The final Vector-to-Raster operation is then performed on each page. The date is printed by the print-software before the first slice is dumped. The black-bit calculation is extremely tedious in FORTRAN and could be optimised in MACRO, if ever... . When a slice is completed it is sent asynchronously to the Versatek and the alternate buffer is filled with the next slice.

It is not very elegant but it has a subtle advantage in that whole pages are always printed and the users do not waste days writing Overlay Descriptions. In addition a logical unit (1) has been freed for general use.

We note finally that the slice-size can be tailored to suit. Having small but many slices reduces the RAST task-image size, but slows the performance. A final advantage of the new system is that two drawings can very simply be merged, should the need ever manifest itself.

```
C
      SUBROUTINE HPDRAW(X,Z,IC)
C
C     DIRECT DRAWING ONTO HP SCREEN
C
C     IC=2 PENDOWN
C     IC=3 PEN-UP
C
      IMPLICIT BYTE Y
      REAL YØ
      LOGICAL LFLAG
      INCLUDE 'COMTEK.FTN'
      INCLUDE 'COMVTK.FTN'
      DATA LGRAP/.FALSE./,IUP/42/,NBUF/Ø/
      DATA YFF,YGS,YUS,YESC/12,29,31,27/
C
      LGSX=.TRUE.
C IF PENDOWN KEEP ADDING BYTES
      IF(LGRAP.AND.IC.EQ.2) GOTO 1
C
C     START NEW GRAPH SEQUENCE
C
      YBUF(NBUF+1)=YGS
      NBUF=NBUF+1
      LGSX=.FALSE.
   1  CONTINUE
C
      XLAST=X+XØ
      ZLAST=Z+YØ
      IX=XLAST*719/26.5
      IZ=ZLAST*359/19.5
      IF(IX.GT.719) IX=719
      IF(IX.LT.Ø) IX = Ø
      IF(IZ.GT.359) IZ=359
      IF(IZ.LT.Ø) IZ=Ø
      YXL=MOD(IX,32)+64
      YYL=MOD(IZ,32)+96
      YXH=IX/32+32
      YYH=IZ/32+32
C
C     PEN DOWN ?
C
      IF(LGRAP.OR.IC.EQ.3)GOTO 12
C
C     SEND OLD DATA
C
      YBUF(NBUF+1)=YOYH
      YBUF(NBUF+2)=YOYL
      YBUF(NBUF+3)=YOXH
      YBUF(NBUF+4)=YOXL
      NBUF=NBUF+4
      LGSX=.TRUE.
  12  CONTINUE
C
C     CHECK CHANGED OR JUST AFTER GS
C
      IF(LGSX.AND.YOYH.EQ.YYH)GOTO 21
      YBUF(NBUF+1)=YYH
      YOYH=YYH
      NBUF=NBUF+1
  21  CONTINUE
      IF(LGSX.AND.YOYL.EQ.YYL.AND.YOXH.EQ.YXH)GOTO 22
      YBUF(NBUF+1)=YYL
      YOYL=YYL
      NBUF=NBUF+1
  22  CONTINUE
      IF(LGSX.AND.YOXH.EQ.YXH)GOTO 23
      YBUF(NBUF+1)=YXH
      YOXH=YXH
      NBUF=NBUF+1
  23  CONTINUE
      YBUF(NBUF+1)=YXL
      YOXL=YXL
      NBUF=NBUF+1
      LGRAP=.TRUE.
      IF(NBUF.LT.74)RETURN
C
      ENTRY HPSEND
C
C     SEND CURRENT BUFFER , ENDING WITH US
C
      IF(NBUF.EQ.Ø)RETURN
      YBUF(NBUF+1)=YUS
      NBUF=NBUF+1
      GOTO 9
C
      ENTRY HPCLER
C
C     INITIALISE TERMINAL
C     SEND ERASE
C
      CALL GETADR(IPARAM(1),YSEND(1))
      CALL SETEF(24)
      TØ=SECNDS(Ø.)-3
C
      YBUF(1)=YESC
      YBUF(2)=YFF
      YBUF(3)=YUS
      YOXH=Ø
      YOYL=Ø
      YOYH=Ø
      NBUF=3
C
C     SEND THE BUFFER
C
   9  CONTINUE
      CALL WAITFR(24)
C
C     CHANGE THE BUFFER
C
      DO 5 I=1,NBUF
   5  YSEND(I)=YBUF(I)
      IPARAM(2)=NBUF
      NBUF=Ø
C
C     SEND
C
  51  DT=SECNDS(TØ)
      IF(DT.LT.Ø.12)GOTO 51
      CALL QIO("41Ø,5,24,1,ISP,IPARAM)
      LGRAP=.FALSE.
      TØ=SECNDS(Ø.)
      RETURN
      END
      SUBROUTINE VCREER
C
      INCLUDE 'VERSTK.COM'
C
C ENSEMBLE DE ROUTINES DESTINEES A CREER UN BUFFER
C CONTENANT LES COORDONNEES DES POINTS A TRACER EN VALEURS ENTIERES
C
      DIMENSION IBUF(256),YBUF(1)
      EQUIVALENCE(IBUF,YBUF)
C
C
      ENTRY VGO
C
      OPEN(UNIT=2,NAME='VP:VFILE.BIN',TYPE='UNKNOWN',
     * ACCESS='SEQUENTIAL',ERR=9ØØ,
     * FORM='UNFORMATTED')
      GO TO 9Ø1
C TRY AGAIN , MAYBE LOCKED
 9ØØ  OPEN(UNIT=2,NAME='VP:VFILE.BIN',TYPE='NEW',
     * ACCESS='SEQUENTIAL',FORM='UNFORMATTED')
 9Ø1  CONTINUE
      IPT=1
      IY3OLD=Ø
C
C IPT EST LE POINTEUR DE IBUF
C
      RETURN
C
C
      ENTRY VFILE(X,Z,IC)
C
C ECRITURE DANS LE BUFFER CREE PAR VGO
C
      IX=X*PTSX
      IF(IX.LT.1) IX=1
      IF(IX.GT.2Ø47) IX=2Ø47
      IY=Z*PTSY
      IF(IY.LT.1) IY=1
      IF(IY.GT.IYMAX) IY=IYMAX
C
      YBUF(IPT)=IAND(IX,127)
      YBUF(IPT+1) = IAND(IY,127)
C
      IY3=IAND(IY,192Ø)/128
      IY3=IY3 + IAND(IX,192Ø)/8
C
      IF(IC.EQ.3) YBUF(IPT)=YBUF(IPT) - 128
C
      IPT=IPT+2
      IF(IY3.EQ.IY3OLD) GO TO 71
C
      IY3OLD=IY3
      IF(IY3.GT.127) IY3 = 127 - IY3
      YBUF(IPT)=IY3
      YBUF(IPT-1)=YBUF(IPT-1) - 128
      IPT=IPT+1
 71   CONTINUE
C
      IF(IPT.LT.51Ø)RETURN
C
C PREPARE LE BUFFER SUIVANT SI LE BUFFER COURANT EST PLEIN
C
 2    IPT=1
      WRITE(2)IBUF
      RETURN
C
C
      ENTRY VPAGE
C
C PASSAGE A LA PAGE SUIVANTE
C
      YBUF(IPT)=Ø
      YBUF(IPT+1)=-128
      YBUF(IPT+2)=Ø
      IPT=IPT+3
      IF(IPT.LT.51Ø)RETURN
C
C BUFFER PLEIN ENVOYE SUR DM:
C
      GO TO 2
C
C
      ENTRY VSTOP
C
C FIN DE L'OPERATION AVEC FERMETURE DU FICHIER
C
      YBUF(IPT)=Ø
      YBUF(IPT+1)=-127
      YBUF(IPT+2)=Ø
C
C Ø,1 INDIQUE LA FIN DE FICHIER
C
      WRITE(2)IBUF
C
C ECRITURE DU DERNIER BUFFER
C
      CLOSE(UNIT=2)
      RETURN
      END
```