

APPENDIX A

MACRO 11 ASSEMBLY LANGUAGE GRAPHICS PACKAGE

```

TITLE TUBEON
  GLOBL TUBEON
  MCALL .MTATCH, .PRINT, .EXIT
; THIS SUBROUTINE WILL ATTACH TEKTRONIX 4006-1
; AT CHANNEL 0 (CONSOLE) TO PLOTTING JOB
;
TUBEON:
  .MTATCH #EMT0, #AST0, #0 ; TRY TO ATTACH CRT
  BCS    CANT              ; IF CANT SEND ERROR MESSAGE
  RTS    PC                ; RETURN
CANT:    .PRINT #TUBMES
  .EXIT
TUBMES:  .ASCII /CAN'T ATTACH CRT ON CHA 0 IN SUB TUBEON/
  .EVEN
EMT0:    .BLKW 4
AST0:    .BLKW 1
  .TITLE TEST
  .GLOBL TEST, MYBUF, MYCNT, PLOT
;
TEST:    MOV     #ZONE2, MYBUF
  MOV     #31, MYCNT
  JSR     PC, PLOT
  RTS     PC
ZONE:    .BYTE 3
  .BYTE 37
ADDR:    .WORD 0
  .BYTE 0
  .BYTE 27
ZONE2:   .BYTE 31
  .BYTE 65
  .BYTE 66
  .BYTE 67
  .BYTE 68
  .BYTE 69
  .BYTE 70
  .BYTE 71
  .BYTE 72
  .BYTE 73
  .BYTE 74
  .BYTE 75
  .BYTE 76
  .BYTE 77
  .BYTE 78
  .BYTE 79
  .BYTE 80
  .BYTE 81
  .BYTE 82
  .BYTE 83
  .BYTE 84
  .BYTE 85

```

```

        BYTE 86.
        .BYTE 87.
        .BYTE 88.
        .BYTE 89.
        .BYTE 90.
        .BYTE 91.
        .BYTE 13.
        .BYTE 10.
        .BYTE 7.
        .EVEN
;
;
;
        .TITLE SENDIT
        .GLOBL SENDIT, BUFADR, TERTYP
        .MCALL .MTOUT, .PRINT, .EXIT
; THIS SUBROUTINE SENDS A CHARACTER BUFFER TO
; THE 4006-1, OR THE 4662
;
;
SENDIT:
        MOV     R0, REG0
        MOV     R1, REG1
        MOV     R2, REG2
        MOV     R3, REG3
        MOV     R4, REG4
        MOV     R5, REG5
        MOV     SP, REG6
        MOV     #SEND, R0
        .MTOUT
        MOV     REG0, R0
        MOV     REG1, R1
        MOV     REG2, R1
        MOV     REG3, R3
        MOV     REG4, R4
        MOV     REG5, R5
        MOV     REG6, SP
        BCS     SORRY
        RTS     PC
SORRY:   .PRINT  #CNTSEN
        .EXIT
CNTSEN:  .ASCII  /ERROR IN SUB SENDIT IN ATTEMPTING TO SEND BUFFER/
SEND:    .BYTE   3
        .BYTE   37
        .WORD   0
        .BYTE   0
        .BYTE   3
BUFADR:  .WORD   SEND+2
BYTCNT:  .WORD   SEND+5
TERTYP:  .WORD   SEND+4

```

```

      .BLKW
REG1:  .BLKW
REG2:  .BLKW
REG3:  .BLKW
REG4:  .BLKW
REG5:  .BLKW
REG6:  .BLKW
;
;
;
      .TITLE PLOT
      .GLOBL MYBUF,MYCNT,TUBSTA,PLTSTA,BUFADR,BYTCNT,TERTYP
; THIS SUBROUTINE OUTPUTS A CHARACTER BUFFER WHO'S ADDRESS
; WAS PREVIOUSLY WRITTEN IN MYBUF AND WHO'S BYTE COUNT WAS
; WRITTEN IN MYCNT...TUBSTA IS A FLAG INDICATING IF CRT IS 0=ON
; OR -1=OFF...PLTSTA SIMILARLY DESCRIBES STATUS OF PLOTTER
;
;
PLOT:
      TST     TRYIT
      BPL     PLOT2
      JSR     PC,TRY
      RTS     PC

PLOT2:
      TST     TUBSTA
      BMI     PLT
      MOV     MYBUF,@BUFADR
      MOVB    MYCNT,@BYTCNT
      MOVB    #0,@TERTYP
      JSR     PC,SENDIT

PLT:   TST     PLTSTA
      BMI     OUT
      MOV     MYBUF,@BUFADR
      MOVB    MYCNT,@BYTCNT
      MOVB    #5,@TERTYP
      JSR     PC,SENDIT
      MOV     MYBUF,MYBUS
      MOV     MYCNT,MYCNS
      MOV     #NULBUF,MYBUF
      MOV     #4,MYCNT
      MOV     MYBUF,@BUFADR
      MOVB    MYCNT,@BYTCNT
      JSR     PC,SENDIT
      MOV     MYBUS,MYBUF
      MOV     MYCNS,MYCNT

OUT:   RTS     PC
MYCNS: .BLKW
MYBUS: .BLKW
NULBUF: .BLKW 10.
MYBUF: .BLKW

```

```

; .BLKW
TUBSTA: .WORD 0
PLTSTA: .WORD 0
;
;
;
; .TITLE TRY
; .GLOBL TRY,PLOT,MYCNT,MYBUF,TRYIT
;
; THIS SUBROUTINE WHEN ACTIVATED BY SETTING TRYIT NEGATIVE
; WILL CAUSE THE OCTAL EQUIVALENT OF EACH ASCII
; CHARACTER TO BE OUTPUT TO THE TERMINALS INSTEAD OF THE CHARACTERS
; THEMSELVES.
;
TRY:
    CLR     TUBSTA           ;TURN ON CRT
    NEG     TRYIT           ;TURN OFF FLAG
    MOV     MYBUF,MYBUF2    ;PROPAGATE BYTE BUFFER POINTER
    MOV     #1,MYCNT2       ;FIRST BYTE TO SEND
    MOV     MYCNT,MYCNTS    ;PREPARE FINISH COUNT
    INC     MYCNTS          ;ONE PAST END OF BUFFER
LOOP:  CMP     MYCNTS,MYCNT2 ;ANY LEFT?
    BEQ     OVER           ;NOPE...JOB IS OVER
    MOVB    @MYBUF2,BYT1    ;MAKE THREE COPIES
    MOVB    @MYBUF2,BYT2
    MOVB    @MYBUF2,BYT3
    MOV     #BYT1,@#4100
    BIC     #177077,BYT1
    ASR     BYT1
    ASR     BYT1
    ASR     BYT1
    ASR     BYT1
    ASR     BYT1
    ASR     BYT1
    ADD     OFFSET,BYT1     ;ADD ASCII OFFSET
    BIC     #177707,BYT2
    ASR     BYT2
    ASR     BYT2
    ASR     BYT2
    ADD     OFFSET,BYT2     ;ADD ASCII OFFSET
    BIC     #177770,BYT3
    ADD     OFFSET,BYT3     ;ADD ASCII OFFSET
    MOV     #BYT1,MYBUF     ;PREPARE FOR SUBROUTINE PLOT
    MOV     #7,MYCNT        ;SEVEN BYTES FOR OUTPUT
    CLR     TUBSTA
    JSR     PC,PLOT
    INC     MYBUF2          ;POINT TO NEXT CHARACTER
    INC     MYCNT2          ;NUMBER SENT
    BR      LOOP           ;GO AROUND FOR NEXT CHARACTER

```

OV:Y

```

      NEG      TRYIT      ;TURN ON TEST FLAG
      RTS      PC      ;RETURN
OFFSET: .WORD 60      ;ASCII OFFSET
MYCNTS: .BLKW
MYBUF2: .BLKW
MYCNT2: .BLKW
BYT1: .BLKW
BYT2: .BLKW
BYT3: .BLKW
      .WORD 40      ;SPACE
;
;
;
;
      .TITLE AOE
      .GLOBL AOE,TRYIT
;
;THIS SUBROUTINE SETS FLAG TO INDICATE ALL GRAPHICS
;OUTPUT IS TO BE IN ASCII OCTAL EQUIVALENT INSTEAD OF
;ASCII CHARACTERS
;
AOE:
      MOV      ←,TRYIT
      RTS      PC
TRYIT: .WORD 1
;
;
;
      .TITLE PLOTON
      .GLOBL PLOTON
      .MCALL .MTATCH,.PRINT,.EXIT
;
; THIS SUBROUTINE WILL ATTACH THE TEKTRONIX 4662
; INTERACTIVE DIGITAL PLOTTER ON CHANNEL FIVE
;
PLOTON:
      .MTATCH #EMTS,#ASTS,#5 ;TRY TO ATTACH PLOTTER
      BCS      NOGOOD      ;IF CANT SEND ERROR MESSAGE
      RTS      PC      ;RETURN
NOGOOD: .PRINT #TROUBLE ;PRINT IT
      .EXIT
TROUBLE: .ASCII /CAN'T ATTACH PLOTTER ON CHA 5 IN SUB TUBEON/
      .EVEN
EMTS: .BLKW 3      ;EMULATOR TRAP AREA FOR CHA 5
ASTS: .BLKW 1      ;ASYN STAT WORD FOR CHA 5
;
;
;
      .TITLE INITT
      .GLOBL INITT

```

THIS SUBROUTINE INITIALIZES WORK ENVIRONMENT

```
;
;
;
;
;
        .TITLE   CLEAR
        .GLOBL   CLEAR,PLOT,MYBUF,MYCNT
;
; THIS SUBROUTINE CLEARS THE SCREEN ON THE 4006-1, RETURNS
; TERMINAL TO ALPHA MODE AND PLACES CURSOR IN HOME POSITION
;
CLEAR:
        MOV       PLTSTA,PLTSTO      ;STORE PLOTTER STATUS
        MOV       #-1,PLTSTA         ;SHUT OFF TRANSMISSION TO PLOTTER
        MOV       #CLRBUF,MYBUF
        MOV       #2.,MYCNT
        JSR       PC,PLOT
        MOV       PLTSTO,PLTSTA      ;RESTORE PLOTTER STATUS
        MOV       #-8000.,TTIME
TMI:    ADD       #1,TTIME            ;ONE SECOND DELAY
        MOV       R0,R0
        MOV       R0,R0
        MOV       R0,R0
        MOV       R0,R0
        CMP       TTIME,#77777
        BNE      TMI
        RTS       PC                 ;RETURN
CLRBUF: .BYTE     27.
        .BYTE     12.
TTIME:  .BLKW
PLTSTO: .BLKW
;
        .TITLE   PLON
        .GLOBL   PLON,PLOT,MYBUF,MYCNT,PLTSTA,TUBSTO
;
; THIS SUBROUTINE TURNS PLOTTER LOGICALLY ON AND
; READ TO PLOT
;
PLON:   CLR       PLTSTA
        MOV       TUBSTA,TUBSTO      ;STORE CRT STATUS
        MOV       #-1,TUBSTA         ;TURN OFF CRT
        MOV       #PLONB,MYBUF
        MOV       #3.MYCNT
```

```

        JSR      PC,PLOT
        MOV      TUBSTO,TUBSTA
        RTS      PC
TUBSTO: .BLKW
PLOB:   .BYTE   27.
        .BYTE   65.
        .BYTE   69.
        .EVEN
;
;
;
        .TITLE  PLOFF
        .GLOBL  PLOFF,PLOT,MYBUF,MYCNT,PLTSTA
;
; THIS SUBROUTINE TURNS PLOTTER LOGICALLY OFF
;
PLOFF:  MOV      #1,PLTSTA
        MOV      TUBSTA,TUBSTO
        MOV      #1,TUBSTA
        MOV      #PLOBFB,MYBUF
        MOV      #3,MYCNT
        JSR      PC,PLOT
        MOV      TUBSTO,TUBSTA
        NEG      PLTSTA
        RTS      PC
PLOBFB: .BYTE   27.
        .BYTE   65.
        .BYTE   70.
        .EVEN
;
;
;
        .TITLE  PRMTON
        .GLOBL  PRMTON,PLOT,MYBUF,MYCNT
;
; THIS SUBROUTINE TURNS ON THE PROMPT LIGHT ON THE 4662
;
PRMTON: MOV      #PRMTOB,MYBUF
        MOV      TUBSTA,TUBSTO
        MOV      #1,TUBSTA
        MOV      #3,MYCNT
        JSR      PC,PLOT
        MOV      TUBSTO,TUBSTA
        RTS      PC
PRMTOB: .BYTE   27.
        .BYTE   65.
        .BYTE   75.
        .EVEN
;
;
;

```



```

        .TITLE PRMTOF
        .GLOBL PRMTOF,PLOT,MYBUF,MYCNT
;
; THIS SUBROUTINE TURNS OFF THE PROMPT LIGHT ON THE 4662
;
PRMTOF: MOV     #PRMTFB,MYBUF
        MOV     TUBSTA,TUBSTO
        MOV     #-1,TUBSTA
        MOV     #3,MYCNT
        JSR     PC,PLOT
        MOV     TUBSTO,TUBSTA
        RTS     PC
PRMTFB: .BYTE   27.
        .BYTE   65.
        .BYTE   78.
        .EVEN
;
;
;
        .TITLE RESET
        .GLOBL RESET,PLOT,MYBUF,MYCNT,PLTSTA
;
; THIS SUBROUTINE RESETS ALL PROGRAMMABLE PARAMETERS ON
; THE 4662 TO THEIR DEFAULT (POWER-UP) VALUES
; THE PARAMETERS INCLUDE ALPHA SCALE, ALPHA ROTATE, ALPHA
; FONT, PROMPT LIGHT, TURNAROUND DELAY, BLOCK SIZE, PROMPT
; CHARACTER, BYPASS CANCEL CHARACTER, AND SIGNATURE CHARACTER
; ALSO GRAPH MEMORY IS CLEARED, PLOTTER IS SET TO ALPHA MODE
; THE OUTPUT BUFFER IS CLEARED AND, THE CARRIAGE RETURN REFERENCE
; POINT IS SET TO HOME POSITION
;
RESET:  MOV     TUBSTA,TUBSTO
        MOV     #-1,TUBSTA
        MOV     #RESETB,MYBUF
        MOV     #3,MYCNT
        JSR     PC,PLOT
        MOV     TUBSTO,TUBSTA
        RTS     PC
RESETB: .BYTE   27.
        .BYTE   65.
        .BYTE   78.
        .EVEN
;
;
;
        .TITLE ALPRST
        .GLOBL ALPRST,PLOT,MYBUF,MYCNT
;
;

```

```

; IS SUBROUTINE RESETS ALL ALPHANUMERIC PRINTING
; PARAMETERS TO THEIR DEFAULT (POWER-UP) VALUES
; THE PARAMETERS INCLUDE ALPHA SCALE, ALPHA ROTATE, ALPHA
; FONT, AND THE CARRIAGE RETURN REFERENCE POINT
;

```

```

ALPRST:

```

```

    MOV     TUBSTA,TUBSTO
    MOV     #1,TUBSTA
    MOV     #ALPRSB,MYBUF
    MOV     #3,MYCNT
    JSR     PC,PLOT
    MOV     TUBSTO,TUBSTA
    RTS     PC

```

```

ALPRSB: .BYTE 27.
        .BYTE 65.
        .BYTE 88.
        .EVEN
;
;
;

```

```

        .TITLE  GIN
        .GLOBL  GIN,PLOT,MYBUF,MYCNT,PLTSTA
;

```

```

; THIS SUBROUTINE PLACES PLOTTER IN GRAPHIC INPUT MODE
; CAUSING PLOTTER TO ENCODE CURRENT PEN COORDINATES AND
; PEN UP/DOWN STATUS FOR TRANSMISSION TO THE HOST SYSTEM
; AND PLACE THAT DATA IN THE OUTPUT QUEUE
;

```

```

GIN:    MOV     #GINB,MYBUF
    MOV     TUBSTA,TUBSTO
    MOV     #1,TUBSTA
    MOV     #3,MYCNT
    JSR     PC,PLOT
    MOV     TUBSTO,TUBSTA
    RTS     PC

```

```

GINB:   .BYTE 27.
        .BYTE 65.
        .BYTE 77.
        .EVEN
;
;
;

```

```

        .TITLE  BLKSTR
        .GLOBL  BLKSTR,PLOT,MYBUF,MYCNT
;

```

```

; THIS SUBROUTINE IS USED FOR BLOCK MODE TRANSMISSION
; IT SIGNALS THE PLOTTER A NEW BLOCK IS BEGINNING AND
; INITIALIZES THE CHECKSUM ACCUMULATOR WITH THE CHARACTER
; '('. THE CHECKSUM ACCUMULATOR GETS BUMPED FOR ALL
; ASCII CHARACTERS UP TO AND INCLUDING ')' EXCEPT NULL(0)
;

```

D SYNC(22)

;

BLKSTR:

```

      MOV     TUBSTA,TUBSTO
      MOV     #1,TUBSTA
      MOV     #BLKSTB,MYBUF
      MOV     #3,MYCNT
      JSR     PC,PLOT
      MOV     TUBSTO,TUBSTA
      RTS     PC

```

```

BLKSTB: .BYTE 27.
        .BYTE 65.
        .BYTE 48.
        .EVEN

```

;

;

;

```

        .TITLE  SIZE
        .GLOBL  SIZE,PLOT,MYBUF,MYCNT

```

; THIS SUBROUTINE CAUSES THE PLOTTER TO SEND THE
 ; X AND Y DIMENSIONS OF THE PHYSICAL PLOTTING
 ; SURFACE TO THE LSI11-23 (IN INCHES) ALONG WITH
 ; OPTIONS INFORMATION

```

SIZE:   MOV     #SIZEB,MYBUF
        MOV     TUBSTA,TUBSTO
        MOV     #1,TUBSTA
        MOV     #3,MYCNT
        JSR     PC,PLOT
        MOV     TUBSTO,TUBSTA
        RTS     PC

```

```

SIZEB:  .BYTE 27.
        .BYTE 65.
        .BYTE 81.
        .EVEN

```

;

;

;

```

        .TITLE  SCALE
        .GLOBL  SCALE,PLOT,MYBUF,MYCNT,PLTSTA,MYNUM,MYOUT

```

; THIS SUBROUTINE IS USED IN ALPHANUMERIC MODE TO
 ; SET THE ALPHANUMERIC SCALE SIZES FOR THE CHARACTER
 ; SPACING (X) AND THE LINE SPACING (Y). THE ACTUAL
 ; CHARACTER HEIGHT IS 11/18 THE LISE;;NE SPACE AND THE
 ; CHARACTER WIDTH IS 6/9 THE CHARACTER SPACE. VALUES ARE
 ; GIVEN IN SCREEN COORDINATES. THE DEFAULT VALUES ARE XVAL=56
 ; AND YVAL=88

;

*

```

$
;
SCALE:
    MOV     TUBSTA,TUBSTO
    MOV     #-1,TUBSTA
    TST     (RS)+      ;SKIP THE ARGUMENT COUNT
    MOV     @(RS)+,XVAL
    MOV     @(RS),YVAL    ;GET THE VALUES FOR X AND Y
    SUB     #4,RS        ;RESTORE REGISTER 5
    MOV     XVAL,MYNUM    ;PREPARE FOR SUBROUTINE NUMBER
    MOV     #XVAL,MYOUT   ;POINTER TO OUTPUT BUFFER
    JSR     PC,NUMBER     ;CONVERT TO ASCII
    MOV     #3,BYTE
    ADD     MYCNT,BYTE    ;COUNTER FOR NUMBER OF BYTES
    MOVB    #44,MYOUT    ;PLACE COMMA IN BUFFER
    INC     MYOUT
    INC     BYTE
    MOV     YVAL,MYNUM    ;CONVERT YVAL TO ASCII
    JSR     PC,NUMBER     ;JUMP TO CONVERSION ROUTINE
    ADD     MYCNT,BYTE    ;ADD NUMBER OF BYTES
    MOVB    #0,MYOUT     ;TERM CHARACTER
    INC     BYTE
    MOV     BYTE,MYCNT    ;PREPARE FOR PLOT ROUTINE
    MOV     #SCALEB+1,MYBUF ;POINTER TO BYTE BUFFER
    MOV     #SCALEB,MYCNT
    MOV     #MYCNT,MYOUT
    JSR     PC,PLOT
    MOV     TUBSTO,TUBSTA
    RTS     PC            ;RETURN

SCALEB: .BYTE 0
        .BYTE 27.
        .BYTE 65.
        .BYTE 73.
XVAL:   .BLKW 2
YVAL:   .BLKW 3
BYTE:   .BLKW

;
;
;
    .TITLE  NUMBER
    .GLOBL  NUMBER,MYCNT,MYOUT,MYNUM

;
; THIS SUBROUTINE ACCEPTS A BUFFER ADDRESS PREVIOUSLY
; WRITTEN IN "MYOUT" AND AN INTEGER NUMBER WRITTEN
; IN "MYNUM" AND FILLS THE BUFFER WITH THE ASCII DECIMAL
; EQUIVALENT OF THE GIVEN NUMBER... "MYCS

```

```

$
; EQUIVALENT OF THE GIVEN NUMBER... "MYCNT" IS FILLED WITH
; THE NUMBER OF BYTES USED
;
;
NUMBER:
      MOV     R0,NREG0      ;PROPAGATE R0
      MOV     R1,NREG1      ;PROPAGATE R1
      MOV     R2,NREG2      ;PROPAGATE R2
      CLR     FLA           ;ZERO DIGIT FLAG
      MOV     #1,MYCNT      ;# OF BYTES +1
      MOV     MYNUM,R1      ;PROPAGATE MYNUM
      CLR     R0           ;READY FOR DIVISION
      DIV     #1000.,R0     ;WHATS IN THOUSANDS PLACE?
      TST     R0           ;NON ZERO?
      BEQ     T100         ;IF ZERO TRY 100'S PLACE
      INC     FLA           ;COUNT FUTURE ZEROS
      ADD     #60,R0        ;ASCII OFFSET
      MOVB    R0,@MYOUT     ;PUT VALUE IN USER BUFFER
      SUB     #60,R0        ;RESTORE
      INC     MYOUT         ;POINT TO NEXT BYTE
      INC     MYCNT         ;BUMP ADDER
      CLR     R1           ;READY FOR MULTIPLICATION
      MUL     #1000.,R0     ;FOR SUBTRACTING FROM NUMBER
      NEG     R1           ;PREPARE MYNUM WITH 1000'S PLACE
      ADD     MYNUM,R1      ;ABSENT
      MOV     R1,MYNUM      ;PRESERVE PREPARATION
T100:  MOV     MYNUM,R1      ;GET FRESH COPY
      CLR     R0           ;READY FOR DIVISION
      DIV     #100.,R0     ;SEPARATE HUNDREDS PLACE
      TST     FLA           ;WAS PREVIOUS DIGIT NON ZERO?
      BNE     T99          ;IF YES NEEDNT TEST PRESENT
      TST     R0           ;BRANCH IF R0 IS ZERO
      BEQ     T10          ;BRANCH IF R0 IS ZERO
      INC     FLA           ;ASCII OFFSET
T99:   ADD     #60,R0        ;PLACE VALUE IN BUFFER
      MOVB    R0,@MYOUT     ;RESTORE
      SUB     #60,R0        ;POINT TO NEXT BYTE
      INC     MYOUT         ;BUMP COUNTER
      INC     MYCNT         ;READY FOR MULTIPLICATION
      CLR     R1           ;PLACE DIGIT IN 100'S PLACE
      MUL     #100.,R0     ;ERASE 100'S PLACE
      NEG     R1           ;FRESH COPY
      ADD     MYNUM,R1
T10:   MOV     R1,MYNUM

      CLR     R0           ;READY FOR DIVISION
      DIV     #10.,R0      ;SEPARATE TENS DIGIT
      TST     FLA           ;WAS PREVIOUS DIGIT NON ZERO?
      BNE     T9

```

```

      TST      R0                      ;IS DIGIT NON ZERO?
      BEQ      T1                      ;IF YES GO TO DO ONES PLACE
T9:    ADD      #68,R0                 ;ASCII OFFSET
      MOVB     R0,MYOUT                ;PLACE VALUE IN USERS BUFFER
      SUB      $68,R0                 ;RESTORE
      INC      MYOUT                  ;POINT TO NEXT BYTE
      INC      MYCNT                  ;BUMP COUNTER
      CLR      R1                     ;READY FOR MULTIPLICATION
      MUL      #18,R0                 ;PLACE DIGIT IN 100'S PLACE
      NEG      R1                     ;ERASE 100'S PLACE
      ADD      MYNUM,R1               ;
      ADD      #68,R1
T1:    MOVB     R1,MYOUT                ;PLACE ONE'S DIGIT IN USER BUFFER
      INC      MYOUT                  ;POINT TO NEXT BYTE
      MOV      NREG0,R0               ;RESTORE REGISTER 0
      MOV      NREG1,R1               ;RESTORE REGISTER 1
      MOV      NREG2,R2               ;RESTORE REGISTER 2
      RTS      PC                     ;RETURN

FLA:   .BLKW
MYOUT:  .BLKW
MYNUM:  .BLKW
NREG0:  .BLKW
NREG1:  .BLKW
NREG2:  .BLKW
;
;
;
;
;
*      .TITLE  ROTATE
      .GLOBL  ROTATE,PLOT,MYBUF,MYCNT,MYNUM,MYOUT
;
; THIS SUBROUTINE IS USED IN ALPHA MODE TO SET THE ANGLE
; AT WHICH ALPHANUMERIC CHARACTERS ARE PRINTED ON THE PLOTTING
; SURFACE.  THE ANGLE IS MEASURED COUNTERCLOCKWISE WITH RESPECT
; TO THE POSITIVE X AXIS...THE POWER UP DEFAULT VALUE IS 0 DEGREES
; THE RANGE IS FROM ZERO TO POSITIVE 360.  THE ANGLE ENTERED MUST
; BE IN SINGLE WORD INTEGER FORMAT AND INDICATE TENTHS OF A DEGREE
; THEREFORE A ROTATION OF 980 PRODUCES THE VERTICLE POSITIVE
; Y AXIS
; IN ADDITION, THIS SUBROUTINE SETS THE CURRENT CARRIAGE RETURN
; REFERENCE POINT IN A LINE PERPENDICULAR TO THE ROTATION ANGLE
; SET
;
ROTATE:
      TST      (R5)+                  ;SKIP THE ARGUMENT COUNT
      MOV      @(R5),ANGL              ;GET THE VALUE FOR ANGLE
      TST      -(R5)                  ;RESTORE REGISTER 5
      MOV      ANGL,MYNUM              ;PREPARE FOR SUBROUTINE NUMBER

```

```

$      MOV     #ANGL,MYOUT      ; POINTER TO OUTPUT BUFFER
      JSR     PC,NUMBER        ; CONVERT TO ASCII
      MOV     MYOUT,POSIT      ; PROPAGATE BUFFER POINTER
      DEC     POSIT
      MOVB    @POSIT,@MYOUT
      MOVB    #46.,@POSIT      ; PUT PERIOD IN BUFFER
      ADD     #4,MYCNT         ; HEADER CHARACTERS
      INC     MYOUT
      CLRB    @MYOUT
      MOV     @ROT+1,MYBUF     ; PREPARE FOR PLOT SUBROUTINE
      JSR     PC,PLOT          ; OUTPUT BUFFER
      RTS     PC               ; RETURN

POSIT: .BLKW 3
ROT:   .BYTE 0
      .BYTE 27.
      .BYTE 65.
      .BYTE 74.
ANGL:  .BLKW 5

```

```

      .TITLE  DATABY
      .GLOBL  DATABY,XPLOT,YPLOT,BUFIT

```

```

; THIS SUBROUTINE WILL CODE TWO 12 BIT INTEGERS INTO
; FIVE ASCII CHARACTERS COMPATIBLE WITH THE 4806-1
; AND THE 4662. XPLOT SHOULD CONTAIN THE X SCREEN COORDINATE
; AND YPLOT THE Y SCREEN COORDINATE. ALSO BUFIT SHOULD
; CONTAIN THE ADDRESS OF THE USER BUFFER IN WHICH THE CHARACTERS
; ARE TO BE PLACED

```

```

DATABY:

```

```

      MOV     R0,GRE1
      MOV     YPLOT,R0
      BIC     #170177,R0
      ASH     #7,R0
      ADD     #40,R0
      MOV     R0,B1
      MOV     YPLOT,R0
      BIC     #177774,R0
      ASH     #2,R0
      MOV     R0,B2
      ADD     #140,B2
      MOV     XPLOT,R0
      BIC     #177774,R0
      ADD     R0,B2
      MOV     YPLOT,R0
      BIC     #177603,R0
      ASH     #2,R0
      ADD     #140,R0

```

```

$
      MOV      R0,B3
      MOV      XPLOT,R0
      BIC      #170177,R0
      ASH      #7,R0
      ADD      #40,R0
      MOV      R0,B4
      MOV      XPLOT,R0
      BIC      #177603,R0
      ASH      #2,R0
      ADD      #100,R0
      MOV      R0,B5
      MOVB     B1,@BUFIT
      INC      BUFIT
      MOVB     B2,@BUFIT
      INC      BUFIT
      MOVB     B3,@BUFIT
      INC      BUFIT
      MOVB     B4,@BUFIT
      INC      BUFIT
      MOVB     B5,@BUFIT
      INC      BUFIT
      MOV      GRE1,R0
      RTS      PC
GRE1:  .BLKW
XPLOT: .BLKW
YPLT:  .BLKW
B1:    .BLKW
B2:    .BLKW
B3:    .BLKW
B4:    .BLKW
B5:    .BLKW
BUFIT: .BLKW
;
;
;
      .TITLE  DRAWI
*      .GLOBL  DRAWI,BUFIT,XPLOT,YPLOT,PLOT,MYBUF,MYCNT
;
;
;THIS SUBROUTINE WILL CAUSE A VECTOR
;TO BE DRAWN FROM THE PRESENT PEN POSITION TO IX,IY
;COORDINATES MUST BE IN SINGLE WORD INTEGER FORMAT.
;THIS SUBROUTINE IS COMPATABLE WITH THE FORTRAN CALL
; 'BCALL DRAWI(IX,IY)
;
;
DRAWI:
      MOV      #882,BUFIT
      JSR      PC,DATABY
      TST      (R5)+          ;SKIP THE ARGUMENT COUNT

```



```

$
      MOV      @(RS)+,XPL0T      ;GET XI
      MOV      @(RS),YPL0T      ;GET IY
      JSR      PC,DATABY        ;CONVERT TO 5 ASCII BYTES
      SUB      #4,RS            ;RESTORE RS
      MOV      #BB2,MYBUF       ;PLOT BUFFER POINTER
      MOV      #10,MYCNT        ;BYTE COUNT
      JSR      PC,PLOT
      RTS      PC
BB2:   .BLKW   5

;
;
;
      .TITLE  MOVEI
      .GLOBL  MOVEI,BUFIT,XPL0T,YPL0T,PLOT,MYBUF,MYCNT

; THIS SUBROUTINE WILL CAUSE AN UNWRITTEN VECTOR TO
; BE DRAWN TO POSITION IXI,IYI
; THE COORDINATES ARE IN SINGLE WORD INTEGER FORMAT
; THIS SUBROUTINE IS COMPATIBLE WITH THE FORTRAN CALL
; CALL MOVEI(IXI,IYI). THE VALUES MUST
; BE IN SCREEN COORDINATES WHERE THE SCREEN DIAGONAL IS
; 0,0 TO 4096,2731. THE PHYSICAL SCREEN IS 10 INCHES
; HIGH AND 15 INCHES LONG
;
MOVEI:
      MOV      #BC2,MYBUF       ;PLOT BUFFER POINTER
      MOV      #1,MYCNT        ;BYTE COUNT
      JSR      PC,PLOT          ;SEND 11. BYTES
      TST      (RS)+
      MOV      @(RS)+,XPL0T
      MOV      @(RS),YPL0T
      SUB      #4,RS
      RTS      PC              ;RETURN
BC2:   .WORD   29.

;
;
;
      .TITLE  POINT
      .GLOBL  POINT
      .MCALL  .MTIN

; THIS SUBROUTINE IS USED IN GRAPHIC INPUT MODE
; TO DIGITIZE DATA ON THE PLOTTER. WHEN CALLED
; FROM A FORTRAN COMMAND CALL POINT (IX,IY), WHERE
; BOTH IX AND IY ARE IN SINGLE WORD INTEGER, THE ROUTINE
; WILL WAIT UNTIL THE USER LINES UP HIS POINT WITH THE
; CROSS HAIR AND PRESSES THE CALL BUTTON. THE TWELVE
; BIT INTEGERS IX AND IY WILL THEN BE RETURNED TO THE PROGRAM
; IN SCREEN COORDINATES OF POINT (IX,IY)
;

```

```

$
POINT:
MOV    #ZSBUF, ZAR1
JSR    PC, BYPASS
MOV    #ZZ5, @#4100
MOV    R0, ZZ0
MOV    R5, ZZ5
MOV    #ZAREA, R0
.MTIN
MOV    #ZSBUF, R5
CLR    R0
MOVB   (R5)+, R0
BIC    #40, R0
ASH    #10, R0
MOV    R0, XIN
*      MOVB   (R5)+, R0
BIC    #40, R0
ASH    #10, R0
MOV    R0, YIN
MOVB   (R5)+, R0
BIC    #40, R0
ASH    #5, R0
ADD    R0, XIN
MOVB   (R5)+, R0
BIC    #40, R0
ASH    #5, R0
ADD    R0, YIN
MOVB   (R5)+, R0
BIC    #40, R0
ADD    R0, XIN
MOVB   (R5), R0
BIC    #40, R0
ADD    R0, YIN
MOV    YIN, R0
ASH    #-3, R0
MOV    R0, YIN
MOV    XIN, R0
ASH    #-3, R0
MOV    R0, XIN
MOV    ZZ5, R5
TST    (R5)+
MOV    XIN, @ (R5)+
MOV    YIN, @ (R5)
MOV    ZZ5, R5
MOV    ZZ0, R0
SS:    MOV    #ZZ5, @#4100
RTS    PC
ZZ5:   .BLKW
ZZ0:   .BLKW
ZAREA: .BYTE   2
        .BYTE  37
ZAR1:  .WORD   ZSBUF

```

```

$
    .BYTE 5
    .BYTE 9.
ZSBUF: .BLKW 4
XIN:   .BLKW
YIN:   .BLKW
;
;
;
    .TITLE BYPASS
    .GLOBL BYPASS
; THIS SUBROUTINE SETS THE BYPASS CANCEL CHARACTER
;
BYPASS:
    MOV     TUBSTA, TUBSTO
    MOV     #-1, TUBSTA
    MOV     #BYBE, MYBUF
    MOV     #4, MYCNT
    JSR     PC, PLOT
    MOV     TUBSTO, TUBSTA
    RTS     PC
BYBE:     .BYTE 27.
          .BYTE 65.
          .BYTE 85.
          .BYTE 37.
;
;
;
    .TITLE ALPHA
    .GLOBL ALPHA
; THIS SUBROUTINE SHIFTS THE PLOTTER TO ALPHA MODE
; TO ENABLE THE PRINTING OF ASCII TEXT CHARACTERS
;
ALPHA:
    MOV     TUBSTA, TUBSTO
    MOV     #-1, TUBSTA
    MOV     #ALPBUF, MYBUF
    MOV     #1, MYCNT
    JSR     PC, PLOT
    MOV     TUBSTO, TUBSTA
    RTS     PC
ALPBUF:    .BYTE 31.
          .EVEN
;
;
;
    .TITLE FONT
    .GLOBL FONT
;
; THIS SUBROUTINE IS USED TO SELECT ONE OF THE SEVEN
*
```

\$
;AVAILABLE PRINTING FONTS WHICH ARE NUMBERED ZERO
;THROUGH SIX.

;
FONT:

```

      MOV     TUBSTA,TUBS00
      MOV     #-1,TUBSTA
      TST     (RS)+
      MOV     @ (RS),FONBUT
      ADD     #48.,FONBUT
      SWAB    FONBUT
      MOVB    #84.,FONBUT
      TST     -(RS)
      MOV     #FONBUF,MYBUF
      MOV     #4,MYCNT
      JSR     PC,PLOT
      MOV     TUBS00,TUBSTA
      RTS     PC
TUBS00: .BLKW
FONBUF: .BYTE 27.
        .BYTE 65.
FONBUT: .BYTE 84.
        .BYTE 8.

```

;
;
;

```

      .TITLE  HOME
      .GLOBL  HOME

```

;

;THIS SUBROUTINE WILL CAUSE THE PEN ON THE 4862
;AND THE CURSOR ON THE 4886-1 TO MOVE TO THE
;UPPER LEFT CORNER OF THE CURRENTLY DEFINED PAGE

;

HOME:

```

      MOV     #HOMBUF,MYBUF
      MOV     #2,MYCNT
      JSR     PC,PLOT
      RTS     PC
HOMBUF: .BYTE 27.
        .BYTE 12.

```

;
;
;
;
;
;
;

```

      .TITLE  BAKSPA
      .GLOBL  BAKSPA

```

;

;THIS SUBROUTINE SENDS AN ALPHANUMERIC BACK SPACE TO THE PLOTTER

;
*

```

$
BAKSPA: MOV    #BAKSPB,MYBUF
        MOV    #1,MYCNT
        JSR    PC,PLOT
        RTS    PC
BAKSPB: .BYTE  8.
        .EVEN
;
;
;
        .TITLE  ZERO
        .GLOBL  ZERO
;
;THIS SUBROUTINE SENDS A ALPHA ZERO TO THE PLOTTER
;
ZERO:   MOV    #ZROBUF,MYBUF
        MOV    #3,MYCNT
        JSR    PC,PLOT
        RTS    PC
ZROBUF: .BYTE  31.
        .BYTE  48.
        .BYTE  29.
        .EVEN
;
;
;
        .TITLE  GRMODE
        .GLOBL  GRMODE
;THIS SUBROUTINE SHIFTS PLOTTER TO GRAPH MODE
;
GRMODE: MOV    #GRBUF,MYBUF
        MOV    #1,MYCNT
        JSR    PC,PLOT
        RTS    PC
GRBUF:  .WORD  29.
;
;
        .TITLE  RETURN
        .GLOBL  RETURN
;
;THIS SUBROUTINE ISSUES A CARRIGE RETURN TO THE PLOTTER
;
RETURN: MOV    #RETFUB,MYBUF
        MOV    #3,MYCNT
        JSR    PC,PLOT
        RTS    PC
RETFUB: .BYTE  31.
        .BYTE  13.

```

```

$
    .BYTE 29.
    .EVEN

;
;
;
;
    .TITLE SINUM
    .GLOBL SINUM

; THIS SUBROUTINE WILL OUTPUT A SINGLE ASCII NUMBER
;
SINUM:
    TST     (R5)+      ; SKIP ARGUMENT COUNT
    MOVB    @ (R5), SINGMN
    SUB     #2, R5     ; RESTORE R5
    ADD     #48., SINGMN ; ADD ASCII OFFSET
    MOV     #1, MYCNT
    MOV     #SINGMN, MYBUF
    JSR     PC, PLOT
    RTS     PC
SINGMN: .WORD 0
;
;
;
;
    .TITLE PRINUM
    .GLOBL PRINUM

; THIS SUBROUTINE WILL PRINT AN INTEGER NUMBER
; ON THE PLOTTER AT THE PRESENT PEN POSITION
; INTEGER MUST BE FOUR DECIMAL DIGITS OR LESS
;
PRINUM:
    TST     (R5)+      ; SKIP THE ARGUMENT COUNT
    MOV     @ (R5)+, MPNUM ; GET INTEGER VALUE TO BE PRINTED
    TST     -(R5)       ; RESTORE REGISTER 5
    MOV     MPNUM, MYNUM ; PREPARE FOR SUBROUTINE NUMBER
    MOV     @MPNUM, MYOUT ; POINTER TO OUTPUT BUFFER
    JSR     PC, NUMBER  ; CONVERT TO ASCII
    MOV     #29., @MYOUT ; RETURN TO GRAPH MODE
    INC     MYCNT
    INC     MYCNT
    MOV     @MYHERE+1, MYBUF
    JSR     PC, PLOT
    RTS     PC
    .EVEN
MYHERE: .BYTE 0
        .BYTE 31.

```

\$
MPNUM: .BLKW 5
 .END TUBEON
*

```

$      SUBROUTINE      AXISCO(MIN,XMAX,YMIN,YMAX,XDIG,YDIG)
      DIMENSION XLABL(50),YLABL(50)
      COMMON/XDIM/XAXL,XAXH,YAXL,YAXH
C      THE PURPOSE OF THIS SUBROUTINE IS TO DRAW THE
C      CARTESIAN COORDINATE AXIS ON THE PLOTTER
      XDIF=XMAX-XMIN
      XLOG=ALOG10(ABS(XDIF))
      IF(XLOG.LT.0.)XLOG=XLOG-1.
      XFACT=(10.0**(AINT(XLOG)))#.1
      XMINN=XMIN/XFACT
      XMAXN=XMAX/XFACT
      YDIF=YMAX-YMIN
      YLOG=ALOG10(ABS(YDIF))
      IF(YLOG.LT.0.)YLOG=YLOG-1.
      YFACT=(10.0**(AINT(YLOG)))#.1
      YMINN=YMIN/YFACT
      YMAXN=YMAX/YFACT
      XMINS=AINT(XMINN)
      XMAXB=AINT(XMAXN)
      IF(XMAXB.NE.XMAXN)XMAXB=XMAXB+1.
      YMINS=AINT(YMINN)
      YMAXB=AINT(YMAXN)
      IF(YMAXB.NE.YMAXN)YMAXB=YMAXB+1.
      YFLA=0.
      IF(YMAXB.GT.20.)GO TO 7
      YMINB=YMINS*10.
      YMAXB=YMAXB*10.
      YFLA=1.
7      CONTINUE
      XFLA=0.
      IF(XMAXB.GT.20.)GO TO 8
      XMINS=XMINS*10.
      XMAXB=XMAXB*10.
      XFLA=1.
8      CONTINUE
      IXLABL=1
      IYLABL=1
      CALL INITT
      CALL PLOX
      CALL TVINDO(389,425,250,1561)
      CALL DWINDO(-6.,8.,YMINB,YMAXB)
      CALL MOVEA(0.,YMINB)
      CALL DRAWA(0.,YMAXB)
      CALL DRAWA(0.,YMINB)
      IYMIN=YMINB
      IYMAX=YMAXB
      W=1.
      IF(IYMAX-IYMIN.LT.50)W=.5
      DO 100 I=IYMIN,IYMAX
      T=I

```



```

5      SCALE2=-2.
      IF((AINT(T*0.2)*5.).EQ.T)SCALE2=-4.
      IF((AINT(T*0.1)*10.).EQ.T)SCALE2=-6.
      IF(SCALE2.NE.-8.)GO TO 55
      YLABL(IYLABL)=I
      IYLABL=IYLABL+1
55     CONTINUE
      CALL DRAWA(0.,T)
      CALL DRAWA(SCALE2,T)
      CALL DRAWA(0.,T)
      IF(V.EQ.1.)GO TO 90
      IF(I.EQ.IYMAX)GO TO 90
      SCALE2=-1.
      WR=W+T
      CALL DRAWA(0.,WR)
      CALL DRAWA(SCALE2,WR)
      CALL DRAWA(0.,WR)
90     CONTINUE
100    CONTINUE
      CALL TWINDO(425,2510,222,250)
      CALL DWINDO(XMINB,XMAXB,-6.,0.)
      CALL MOVEA(XMINB,0.)
      IXMIN=XMINB
      IXMAX=XMAXB
      W=1.
      IF(IXMAX-IXMIN.LT.50)W=0.5
      DO 200 I=IXMIN,IXMAX
      T=I
      SCALE2=-2.
      IF((AINT(T*0.2)*5.).EQ.T)SCALE2=-4.
      IF((AINT(T*0.1)*10.).EQ.T)SCALE2=-6.
      IF(SCALE2.NE.-6.)GO TO 66
      XLABL(IXLABL)=I
      IXLABL=IXLABL+1
66     CONTINUE
      CALL DRAWA(T,0.)
      CALL DRAWA(T,SCALE2)
      CALL DRAWA(T,0.)
      IF(V.EQ.1.)GO TO 190
      IF(I.EQ.IXMAX)GO TO 190
      SCALE2=-1.
      WR=W+T
      CALL DRAWA(WR,0.)
      CALL DRAWA(WR,SCALE2)
      CALL DRAWA(WR,0.)
190    CONTINUE
200    CONTINUE
      IF(CYFLA.EQ.0.)GO TO 87
34     CONTINUE
      YMINB=YMINB*0.1

```

```

YMAXB=YMAXB*0.1
DO 88 IJ=1,IYLABL-1
YLABL(IJ)=YLABL(IJ)*0.1
88 CONTINUE
87 CONTINUE
IF(YLABL(IYLABL-2).GT.10.0**YDIG)GO TO 34
GO TO 350
370 CONTINUE
YMINB=YMINB*10.
YMAXB=YMAXB*10.
DO 360 IJ=1,IYLABL-1
YLABL(IJ)=YLABL(IJ)*10.
360 CONTINUE
350 CONTINUE
IF(XLABL(IYLABL-2).LT.10.0**(YDIG-1))GO TO 370
GO TO 250
270 CONTINUE
XMINB=XMINB*10.
XMAXB=XMAXB*10.
DO 260 IJ=1,IXLABL-1
XLABL(IJ)=XLABL(IJ)*10.
260 CONTINUE
250 CONTINUE
IF(XLABL(IXLABL-2).LT.10.0**(XDIG-1))GO TO 270
GO TO 85
31 CONTINUE
XMINB=XMINB*0.1
XMAXB=XMAXB*0.1
DO 86 IJ=1,IXLABL-1
XLABL(IJ)=XLABL(IJ)*0.1
86 CONTINUE
85 CONTINUE
IF(XLABL(IXLABL-2).GT.10.0**XDIG)GO TO 31
CALL TWINDO(314,389,241,1544)
CALL DWINDO(0.,1.,YMINB,YMAXB)
CALL SCALE(25,44)
DO 300 I=1,IYLABL-1
CALL RETURN
CALL MOVEA(1.,YLABL(I))
CALL DRAVA(1.,YLABL(I))
IIL=YLABL(I)
CALL PNUM(IIL)
300 CONTINUE
CALL TWINDO(453,2546,172,208)
CALL DWINDO(XMINB,XMAXB,0.,1.)
DO 400 I=1,IXLABL-1
CALL RETURN
CALL MOVEA(XLABL(I),0.)
CALL DRAVA(XLABL(I),0.)
IIL=XLABL(I)

```

```

$
400 CALL PNUM (IIL)
CONTINUE
YAXL=YMINB
YAXH=YMAXB
XAXL=XMINB
XAXH=XMAXB
RETURN
END
SUBROUTINE YAXIS(ICHAR,STRING)
CALL RETURN
CALL TVINDO(300,395,425,1544)
CALL DWINDO(0.,1.,0.,1.)
CALL MOVEA(0.,0.)
CALL DRAWA(0.,0.)
CALL ROTATE(900)
CALL SCALE(975/ICHAR,100)
CALL ALPHA
I=MTOUT(5,STRING,ICHAR)
CALL GRMODE
CALL SETUP(A,B,C,D)
RETURN
END
SUBROUTINE XAXIS(I2CHAR,STRIN2)
CALL RETURN
CALL TVINDO(875,1793,100,258)
CALL DWINDO(0.,1.,0.,1.)
CALL MOVEA(0.,0.)
CALL DRAWA(0.,0.)
CALL ROTATE(3600)
CALL SCALE(1200/I2CHAR,100)
CALL ALPHA
I=MTOUT(5,STRIN2,I2CHAR)
CALL GRMODE
CALL SETUP(A,B,C,D)
RETURN
END
SUBROUTINE PNUM(IX)
C THIS SUBROUTINE CAUSES A NUMBER (0 TO 999) TO BE PRINTED
C ON PLOTTER
CALL ALPHA
CALL BAKSPA
IF(IX.GE.10)CALL BAKSPA
IF(IX.GE.100)CALL BAKSPA
IHUN=INT(IX*0.01)
ITEN=INT((IX-IHUN*100.)*0.1)
IONE=IX-IHUN*100.-ITEN*10.
CALL ALPHA
IF(IHUN.NE.0)CALL SINUM(IHUN)
IF(IHUN.NE.0.OR.ITEN.NE.0)CALL SINUM(ITEN)
CALL SINUM(IONE)

```

```

$      CALL GRMODE
      RETURN
      END
      SUBROUTINE DWINDO(XMIN,XMAX,YMIN,YMAX)
C **** THIS SUBROUTINE ENTERS DIMENSIONS OF THE WORLD
C **** COORDINATE SYSTEM INTO THE WORLD COMMON BLOCK
C
C
      COMMON/WORLD/WXMIN,WXMAX,WYMIN,WYMAX,DWX,DWY
      DWX=XMAX-XMIN
      DWY=YMAX-YMIN
      WXMIN=XMIN
      WXMAX=XMAX
      WYMIN=YMIN
      WYMAX=YMAX
      RETURN
      END

C
C *****
C
      SUBROUTINE TWINDO(IXMIN,IXMAX,IYMIN,IYMAX)
C
C **** THIS SUBROUTINE ENTERS THE DIMENSIONS OF THE SCREEN
C **** COORDINATE SYSTEM INTO THE SCREEN COMMON BLOCK
C
      COMMON/SCREEN/ISXMIN,ISXMAX,ISYMIN,ISYMAX,ISDX,ISDY
      ISDX=IXMAX-IXMIN
      ISDY=IYMAX-IYMIN
      ISXMIN=IXMIN
      ISXMAX=IXMAX
      ISYMIN=IYMIN
      ISYMAX=IYMAX
      RETURN
      END

C
C *****
C
      SUBROUTINE MOVEA(X,Y)
C
C **** THIS SUBROUTINE CAUSES A LIGHT VECTOR TO BE DRAWN FROM
C **** THE CURRENT PEN POSITION TO LOCATION X,Y
C
      COMMON/WORLD/WXMIN,WXMAX,WYMIN,WYMAX,DWX,DWY
      COMMON/SCREEN/ISXMIN,ISXMAX,ISYMIN,ISYMAX,ISDX,ISDY
      XX=(X-WXMIN)/DWX
      YY=(Y-WYMIN)/DWY
      IX=INT(XX*ISDX)+ISXMIN
      IY=INT(YY*ISDY)+ISYMIN
      CALL MOVEI(IX,IY)
      RETURN

```

```

†
      END
C
C*****
C
      SUBROUTINE DRAWACX,Y)
C
C **** THIS SUBROUTINE CAUSES A DARK (WRITTEN) VECTOR TO BE DRAWN
C **** FROM THE PRESENT PEN POSITION TO THE WORLD COORDINATES X,Y
C
      COMMON/WORLD/WXMIN, WXMAX, WYMIN, WYMAX, DWX, DWY
      COMMON/SCREEN/ISXMIN, ISXMAX, ISYMIN, ISYMAX, ISDX, ISDY
      XX=(X-WXMIN)/DWX
      YY=(Y-WYMIN)/DWY
      IX=INT(CXX*ISDX)+ISXMIN
      IY=INT(CYY*ISDY)+ISYMIN
      CALL DRAWI(IX,IY)
      RETURN
      END
      SUBROUTINE SETUPCXL,XH,YL,YH)
      COMMON /AXDIM/XAXL,XAXH,YAXL,YAXH
      CALL TWINDO(425,2518,258,1561)
      XL=XAXL
      XH=XAXH
      YL=YAXL
      YH=YAXH
      CALL DWINDOXL,XH,YL,YH)
      RETURN
      END

```

```

      DIMENSION W(2000)
      OPEN (UNIT=1,TYPE='OLD',NAME='DY0:CAHN.DAT')
      READ (1,5)WEIGHT
5      FORMAT(6)
      READ (1,10)((JJ,W(I),T),I=1,2000)
      S=W(1)
      WEIGHT=WEIGHT+W(1)
      DO 200 I=1,2000
      W(I)=(W(I)-S)/WEIGHT*100.
200    CONTINUE
      DO 300 IN=1,2000
      IF(W(IN).GT.0.05)GO TO 350
300    CONTINUE
350    CONTINUE
      CALL AXIS(0.,15.,0.,W(2000-IN),2.,1.)
      CALL YAXIS(20,'WT X NONANE ADSORBED')
      CALL XAXIS(14,'SECONDS ** 1/2')
      CALL MOVEA(0.,8.)
      X=0.
      DO 100 I=IN,2000-IN
      X=X+0.1
      T=SQRT(X)
      CALL DRAWACT,W(I))
10      FORMAT(I,F12.1,F6.1)
100    CONTINUE
      END

```