

Graph2D Library --- CP/M ---

Generated by Doxygen 1.8.19

1 Graph2D - CP/M Port	1
2 File Index	3
2.1 File List	3
3 File Documentation	5
3.1 'tktrnx.fd' File Reference	5
3.1.1 Detailed Description	5
3.2 'tktrnx.fd'	5
3.3 Alpha.mac File Reference	5
3.3.1 Detailed Description	6
3.4 Alpha.mac	6
3.5 Gethdc.asm File Reference	7
3.5.1 Detailed Description	7
3.6 Gethdc.asm	7
3.7 Graph.asm File Reference	8
3.7.1 Detailed Description	8
3.8 Graph.asm	9
3.9 Graph.bib File Reference	9
3.9.1 Detailed Description	9
3.10 Graph.bib	10
3.11 Graphjob.sou File Reference	11
3.11.1 Detailed Description	11
3.12 Graphjob.sou	11
3.13 Graphsta.sou File Reference	12
3.13.1 Detailed Description	12
3.14 Graphsta.sou	12
3.15 Hdcout.for File Reference	12
3.15.1 Detailed Description	13
3.15.2 Function/Subroutine Documentation	13
3.15.2.1 hdcout()	13
3.16 Hdcout.for	13
3.17 Hdcout1.asm File Reference	15
3.17.1 Detailed Description	15
3.18 Hdcout1.asm	15
3.19 Mainpage.dox File Reference	15
3.20 Print.asm File Reference	15
3.20.1 Detailed Description	16
3.21 Print.asm	16
3.22 TCS.for File Reference	16
3.22.1 Detailed Description	17
3.22.2 Function/Subroutine Documentation	18
3.22.2.1 ancho()	18

3.22.2.2 anstr()	18
3.22.2.3 baksp()	18
3.22.2.4 cartn()	18
3.22.2.5 dasha()	18
3.22.2.6 dashr()	19
3.22.2.7 drawa()	19
3.22.2.8 drawr()	19
3.22.2.9 dwindo()	19
3.22.2.10 genflg()	19
3.22.2.11 home()	20
3.22.2.12 linef()	20
3.22.2.13 linhgt()	20
3.22.2.14 lintrn()	20
3.22.2.15 linwdt()	20
3.22.2.16 logtrn()	20
3.22.2.17 movea()	21
3.22.2.18 mover()	21
3.22.2.19 newlin()	21
3.22.2.20 newpag()	21
3.22.2.21 pointa()	21
3.22.2.22 pointr()	22
3.22.2.23 rel2ab()	22
3.22.2.24 rescal()	22
3.22.2.25 revcot()	22
3.22.2.26 rrotat()	22
3.22.2.27 rscale()	23
3.22.2.28 seetrm()	23
3.22.2.29 seetrn()	23
3.22.2.30 setmrg()	23
3.22.2.31 swindo()	23
3.22.2.32 twindo()	24
3.22.2.33 vcursr()	24
3.22.2.34 vwindo()	24
3.22.2.35 wincot()	24
3.23 TCS.for	25
3.24 Tcsbasi1.asm File Reference	31
3.24.1 Detailed Description	31
3.25 Tcsbasi1.asm	31
3.26 Tcsbasi2.asm File Reference	39
3.26.1 Detailed Description	39
3.27 Tcsbasi2.asm	40
3.28 TCSdrCPM.for File Reference	46

3.28.1 Detailed Description	46
3.28.2 Function/Subroutine Documentation	46
3.28.2.1 csize()	46
3.28.2.2 dcursr()	47
3.28.2.3 initt()	47
3.28.2.4 restat()	47
3.28.2.5 seeloc()	47
3.28.2.6 svstat()	47
3.28.2.7 tcslev()	48
3.29 TCSdrCPM.for	48
3.30 Tcsdrivr.asm File Reference	49
3.30.1 Detailed Description	49
3.31 Tcsdrivr.asm	50
3.32 Tcsgin.asm File Reference	53
3.32.1 Detailed Description	53
3.33 Tcsgin.asm	54
3.34 Tcshdc.asm File Reference	55
3.34.1 Detailed Description	55
3.35 Tcshdc.asm	56
3.36 Tcsital.asm File Reference	57
3.36.1 Detailed Description	57
3.37 Tcsital.asm	57
Index	59

Chapter 1

Graph2D - CP/M Port

Tektronix Graphic Driver for Commodore 128 running CP/M

The library was developed with the Microsoft Fortran IV compiler and the CP/M 8080 assembler RMAC together with the Z80 and 6502 Macros. A detailed description of the system can be found in </doc/ProgDoc/tekcpm.pdf> (in German, sorry for that...).

How to build the library:

In order to compile the library, the Commodore 128 has to be equipped with a 512 kB RAM-disc. Copy the sources from `/build` to the RAM-disc and execute `Tcsbuild.sub`. Due to limited disc space, the sources will be erased after compilation.

Using the library:

The basic drivers have to be copied to Bank 0 (e.g. during boot) before calling a graphic application the first time.

This is done by

[Graph.asm](#), [Tcsbasi1.asm](#), [Tcsbasi2.asm](#).

To switch back to alphanumerical mode use [Alpha.mac](#).

[TCS.for](#), [TCSdrCPM.for](#), [Tcsdrivr.asm](#), [Tcsgin.asm](#), [Tcshdc.asm](#), [Tcsital.asm](#)

build the library for the application programs

Hardcopies could be printed with the help of the spooler

[Hdcout.for](#), [Gethdc.asm](#), [Hdcout1.asm](#), [Print.asm](#)

The program is written for a 7 dot matrix printer and has to be adapted to the hardware used.

The file [Graph.bib](#) implements the missing graphic support of the Turbo-Pascal 3.0 Compiler.

Chapter 2

File Index

2.1 File List

Here is a list of all files with brief descriptions:

'tktrnx.fd'	TCS: Common Block TKTRNX in TCS.FOR	5
Alpha.mac	CP/M: shell command - switch to alphanumeric display mode	5
Gethdc.asm	CP/M Port: Read hardcopy file	7
Graph.asm	CP/M: Installer for the Routines in Bank 0	8
Graph.bib	CP/M Pascal: Implementation of Turbo Pascal 3.0 Graphic routines	9
Graphjob.sou	CP/M Pascal: Source of the inlinecode in Graph.bib	11
Graphsta.sou	CP/M Pascal: Source of the inlinecode in Graph.bib	12
Hdcout.for	CP/M: Printer spooler for 7-dot matrix printer Seikosha GP100VC	12
Hdcout1.asm	CP/M Port: Read VDC-RAM	15
Print.asm	CP/M Port: Direct output to printer	15
TCS.for	TCS: Tektronix Plot 10 Emulation	16
Tcsbasi1.asm	CP/M Port: Basic Graphic Routines Bank 0	31
Tcsbasi2.asm	CP/M Port: Basic Graphic Routines Bank 0 - subprograms	39
TCSdrCPM.for	CP/M Port: Basic graphic subroutines for C128 under CP/M	46
Tcsdrivr.asm	CP/M Port: Direct graphic	49
Tcsgin.asm	CP/M Port: Graphic input	53
Tcshdc.asm	CP/M Port: Hardcopy	55
Tcsital.asm	CP/M Port: Italic	57

Chapter 3

File Documentation

3.1 'tktrnx.fd' File Reference

TCS: Common Block TKTRNX in TCS.FOR.

3.1.1 Detailed Description

TCS: Common Block TKTRNX in TCS.FOR.

Definition in file ['tktrnx.fd'](#).

3.2 'tktrnx.fd'

```
00001      COMMON /TKTRNX/
00002 C      kbaudr,kerror,kgrafi,khomey,kkmode,
00003 C      khorsz,kversz,kitalc,ksizef,
00004      1 klmrgn,kmrngn,
00005 C      ktblsz,khorzt(10),kvertt(10)
00006 C      kbeamx,kbeamy,kmovef,kpchar(4),kdasht,
00007      4 kminsx,kminsy,kmaxsx,kmaxsy,tminvx,tminvy,tmaxvx,tmaxvy,
00008 C      trealx,trealy,timagx,timagy,
00009      5 trcosf,trsinf,trscal
00010      U ,xfac,yfac,xlog,ylog
00011
```

3.3 Alpha.mac File Reference

CP/M: shell command - switch to alphanumeric display mode.

3.3.1 Detailed Description

CP/M: shell command - switch to alphanumeric display mode.

Version

1.0

Author

(C) 1995 Dr.-Ing. Klaus Friedewald

Copyright

GNU LESSER GENERAL PUBLIC LICENSE Version 3

Definition in file [Alpha.mac](#).

3.4 Alpha.mac

```

00001      title 'Umschaltung alphanumericische Anzeige und Versionskennung'
00002      .comment *
00003
00004 - Aufruf: alpha
00005
00006 Version 1.0: 6.1.94      Dr. K.Friedewald
00007      *
00008
00009      .z80
00010 TcsTab equ      4000h
00011 jmpfar equ      0ffffh
00012 bdos equ      05h
00013 PrStr equ      09h      ; Bdos   9: Print String
00014 PrChr equ      2       ; Bdos   2: Console Output
00015
00016 ;
00017 ; Umschaltung in den alphahnumerischen Modus
00018 ;
00019      ld      hl,TcsTab+3
00020      call    JmpFar
00021 ;
00022 ; Holen Version des Graphiktreibers
00023 ;
00024      ld      hl,TcsTab+3fh
00025      call    JmpFar
00026
00027      ld      a,d           ; Version,Release
00028      rrca
00029      rrca
00030      rrca
00031      rrca
00032      and     0fh           ; Version
00033      add     a,'0'         ; ASCII Ziffer
00034      ld      (VerSt),a
00035      ld      a,d
00036      and     0fh           ; Release
00037      add     a,'0'
00038      ld      (RelSt),a
00039
00040      ld      hl,OptSt
00041      ld      a,e           ; Options
00042      ld      c,'+'         ; ASCII gesetzt
00043      ld      b,8           ; 8 Bit
00044 loop:  rlca
00045      jr      NC,No
00046      ld      (hl),c        ; Bit gesetzt
00047 No:     inc     hl
00048      djnz    loop
00049
00050      ld      de,Strng      ; Drucken
00051      ld      c,PrStr

```

```

00052      call    bdos
00053
00054      jp      0
00055
00056 Strng: db      'TCS Version '
00057 VerSt: db      '0.'
00058 RelSt: db      '0 - Options ('
00059 OptSt: db      '.....)$'
00060
00061      end

```

3.5 Gethdc.asm File Reference

CP/M Port: Read hardcopy file.

3.5.1 Detailed Description

CP/M Port: Read hardcopy file.

Version

1.0

Author

(C) 1995 Dr.-Ing. Klaus Friedewald

Copyright

GNU LESSER GENERAL PUBLIC LICENSE Version 3

Definition in file [Gethdc.asm](#).

3.6 Gethdc.asm

```

00001      title    'TCS: Wiedereinlesen von Hardcopyfiles'
00002 ;   Einlesen eines *.HDC-Files. Aufruf aus Fortran:
00003 ;
00004 ;   LOGICAL GETHDC
00005 ;   ...
00006 ;   IF (GETHDC (Filename)) goto error
00007 ;
00008 ;   26.5.88           Dr.-Ing. K. Friedewald
00009 ;
00010 $*Macro
00011
00012      maclib    z80
00013      public    gethdc
00014      name      'GETHDC'
00015
00016 tcstab equ     4000h
00017 jmpfar equ     0FFF1h
00018
00019 openf  equ     15
00020 setdma equ     26
00021 multio equ     44
00022 reads  equ     20
00023 closef equ     16
00024
00025 filsiz equ     640/8*200/128    ; Anzahl Records
00026
00027 bdos    macro    function, parameter

```

```

00028      mvi      c,function
00029      lxi      d,parameter
00030      call     5
00031      endm
00032
00033 gethdc  lxi      d,fcbnam      ; Kopieren Filename
00034      lxi      b,8              ; 8 Buchstaben
00035      ldir
00036      mvi      a,0              ; Loeschen FCB-Felder
00037      sta      fcbcr
00038      sta      fcbex
00039
00040      bdos     openf,fcb
00041      cpi      255              ; File vorhanden?
00042      rz                      ; nein => false
00043      bdos     setdma,iobuf
00044      bdos     multio,1
00045
00046      lxi      d,0-1            ; Bildschirmadresse-1
00047      mvi      b,Filsiz        ; Anzahl Records
00048
00049 Getlp   push     b
00050      push     d
00051      bdos     reads,fcb
00052      pop      d
00053      mvi      b,128            ; Laenge log. Record
00054      lxi      h,iobuf
00055      shld     bufptr
00056 getlp1  inx      d            ; naechste VDC-Adresse
00057      push     d
00058      push     b                ; restliche Blocklaenge
00059      lhld     bufptr
00060      mov      c,m
00061      inx      h
00062      shld     bufptr
00063      lxi      h,tcstab+2Ah     ; Poke VDC
00064      call     jmpfar
00065      pop      b
00066      pop      d
00067      djnz     getlp1
00068      pop      b
00069      djnz     getlp
00070
00071      bdos     closef,fcb
00072      ret
00073
00074      dseg
00075 fcb      db      0              ; Default Drive
00076 fcbnam   db      '???????'    ; Filename
00077      db      'HDC'            ; extension
00078 fcbex    db      0
00079      ds      19
00080 fcbcr    db      0
00081
00082 iobuf     ds      128
00083 bufptr    dw      0
00084      end
00085

```

3.7 Graph.asm File Reference

CP/M: Installer for the Routines in Bank 0.

3.7.1 Detailed Description

CP/M: Installer for the Routines in Bank 0.

Version

1.0

Author

(C) 1995 Dr.-Ing. Klaus Friedewald

Copyright

GNU LESSER GENERAL PUBLIC LICENSE Version 3

Definition in file [Graph.asm](#).

3.8 Graph.asm

```
00001          title   'Initialisierung C128 Grundgraphik'
00002 ;
00003 ; Installationsprogramm fuer die Graphikroutinen in Bank 0
00004 ;
00005          Aufruf:    GRAPH
00006 ;                  Kopiert die Routinen nach Bank0. Bei Angabe eines
00007 ;                  beliebigen Buchstabens als Parameter wird zusaetzlich
00008 ;                  der Textmodus eingeschaltet.
00009 ;
00010 ;          19.4.88      Dr.-Ing. K. Friedewald
00011 ;
00012 $*Macro
00013
00014          extrn      TCSinit
00015          maclib     z80
00016 TcsTab    equ       4000h
00017 Jmpfar    equ       0FFF1h
00018 CmdTail   equ       080h
00019
00020          call       TCSinit
00021
00022          lda        CmdTail
00023          cmp        0
00024          jrz        NoText
00025          lxi        h,TcsTab+3      ; TextMode
00026          call       jmpfar
00027
00028 NoText    jmp       0
00029          end
00030
```

3.9 Graph.bib File Reference

CP/M Pascal: Implementation of Turbo Pascal 3.0 Graphic routines.

3.9.1 Detailed Description

CP/M Pascal: Implementation of Turbo Pascal 3.0 Graphic routines.

Version

1.0

Author

(C) 1995 Dr.-Ing. Klaus Friedewald

Copyright

GNU LESSER GENERAL PUBLIC LICENSE Version 3

Definition in file [Graph.bib](#).

3.10 Graph.bib

```

00001 (* Turbo-Pascal Treiber fuer C128-Graphik
00002      17.5.88      Dr.-Ing. K. Friedewald      *)
00003
00004 const
00005   OrgConOut = 825; (* Turbo 3.0 und CP/M 80 *)
00006   OrgError = 8414;
00007
00008 var
00009   WhereX, WhereY : integer;
00010
00011 procedure OrgGotoXY (x,y:integer); begin GotoXY(x,y) end;
00012
00013 procedure graphjob (JumpAdr, par1, par2, par3, par4: integer);
00014   begin INLINE ($2A/JMPADR/$ED/$5B/PA1/$ED/$4B/PA2/$DD/$2A/PA3
00015     /$FD/$2A/PA4/$CD/$F1/$FF ) end;
00016
00017 procedure GraphStatus (JumpAdr: integer; var par1,par2,par3,par4: integer);
00018   begin INLINE ($2A/PA1/$5E/$23/$56/$2A/PA2/$4E/$23/$46/$2A/JMPADR/$CD/$F1/$FF
00019     /$2A/PA1/$73/$23/$72/$2A/PA2/$71/$23/$70/$2A/PA3/$DD/$E5/$D1/$73/$23
00020     /$72/$FD/$E5/$D1/$2A/PA4/$73/$23/$72);   end;
00021
00022 procedure GraphConout (c:char);
00023   var x1,x2,y1,y2: integer;
00024   begin
00025     GraphStatus ($4030,x1,x2,y1,y2); (* Getwindow *)
00026     GraphJob ($401e,0,0,639,199); (* SetWindow *)
00027     GraphJob ($4012,(WhereX-1)*8,WhereY*8-1,0,0);
00028     GraphJob ($401b,ord(c),0,0,0);
00029     GraphJob ($401e,x1,y1,x2,y2); (* Altes Window *)
00030     GraphStatus ($402d,WhereX,WhereY,x1,x1);
00031     WhereX:= succ( succ(WhereX) div 8);   WhereY:= succ(WhereY) div 8;
00032   end;
00033
00034
00035 procedure TextMode;
00036   begin
00037     ConOutPtr:= OrgConOut;
00038     ErrorPtr:= OrgError;
00039     GraphJob ($4003,0,0,0,0);
00040   end;
00041
00042 procedure GraphError (nr, adr:integer);
00043   begin
00044     TextMode; writeln ('Runtime-Error im HiRes-Modus aufgetreten!'); halt
00045   end;
00046
00047 procedure HiRes;
00048   begin
00049     if ConOutPtr=OrgConOut then begin
00050       ConOutPtr:= addr(GraphConOut);
00051       ErrorPtr:= addr(GraphError);
00052     end;
00053     WhereX:= 1; WhereY:= 1;
00054     GraphJob ($4000,0,0,0,0)
00055   end;
00056
00057 procedure HiResColor (Col:integer); begin GraphJob ($4006,col,16,0,0) end;
00058
00059 procedure GraphBackground (Col:integer); begin GraphJob ($4006,16,col,0,0) end;
00060
00061 procedure GraphWindow (x1,y1,x2,y2:integer); begin GraphJob ($401e,x1,y1,x2,y2); end;
00062
00063 procedure Plot (x,y, Colour:integer);
00064   begin
00065     if Colour=0 then GraphJob ($400C, x,y,0,0)
00066     else
00067       GraphJob ($4009, x,y,0,0)
00068     end;
00069
00069 procedure Draw (x1,y1,x2,y2, Colour:integer);
00070   begin
00071     GraphJob ($4012,x1,y1,0,0);
00072     if Colour=0 then GraphJob ($400f, x2,y2,0,0)
00073     else
00074       GraphJob ($400f, x2,y2,255,0)
00075     end;
00076
00076 procedure TextColor (Colour: integer);
00077   begin
00078     if colour>15 then begin
00079       colour:= colour-16;
00080       if ConOutPtr=OrgConOut then write (#27,'G2') (* Blinken ein *) end
00081     else begin
00082       if ConOutPtr=OrgConOut then write (#27,'G0')
00083     end;
00084     HiResColor (Colour)
00085   end;

```



```

00086
00087 procedure TextBackGround (Colour: integer);
00088 begin
00089   GraphBackGround (Colour)
00090 end;
00091
00092 procedure GotoXY (x,y: integer);
00093 begin
00094   if ConOutPtr=OrgConOut then
00095     OrgGotoXY (x,y)
00096   else begin
00097     WhereX:= x;      WhereY:= y
00098   end
00099 end;
00100
00101 function GetDotColor (x,y: integer): integer;
00102 var
00103   dum : integer;
00104 begin
00105   GraphStatus ($4024,x,y,dum,dum);      (* CheckPoint *)
00106   GetDotColor:= x
00107 end;
00108

```

3.11 Graphjob.sou File Reference

CP/M Pascal: Source of the inlinecode in [Graph.bib](#).

3.11.1 Detailed Description

CP/M Pascal: Source of the inlinecode in [Graph.bib](#).

Author

(C) 1995 Dr.-Ing. Klaus Friedewald

Copyright

GNU LESSER GENERAL PUBLIC LICENSE Version 3

Definition in file [Graphjob.sou](#).

3.12 Graphjob.sou

```

00001 ; Pascal Procedure Graphjob fuer GRAPH.BIB
00002
00003 jmpfar equ      0ffff1h
00004 jmpadr ext
00005 par1  ext
00006 par2  ext
00007 par3  ext
00008 par4  ext
00009
00010      ld      hl, (JmpAdr)
00011      ld      de, (Par1)
00012      ld      bc, (Par2)
00013      ld      ix, (Par3)
00014      ld      iy, (Par4)
00015      call   jmpfar
00016      end

```

3.13 Graphsta.sou File Reference

CP/M Pascal: Source of the inlinecode in [Graph.bib](#).

3.13.1 Detailed Description

CP/M Pascal: Source of the inlinecode in [Graph.bib](#).

Author

(C) 1995 Dr.-Ing. Klaus Friedewald

Copyright

GNU LESSER GENERAL PUBLIC LICENSE Version 3

Definition in file [Graphsta.sou](#).

3.14 Graphsta.sou

```
00001 ; Procedure GraphStatus fuer Graph.bib
00002
00003 jmpfar equ      0ffff1h
00004 Jmpadr ext
00005 par1  ext
00006 par2  ext
00007 par3  ext
00008 par4  ext
00009
00010      ld      hl, (Par1)
00011      ld      e, (hl)
00012      inc     hl
00013      ld      d, (hl)
00014      ld      hl, (par2)
00015      ld      c, (hl)
00016      inc     hl
00017      ld      b, (hl)
00018      ld      hl, (JmpAdr)
00019
00020      call    jmpfar
00021
00022      ld      hl, (Par1)
00023      ld      (hl), e
00024      inc     hl
00025      ld      (hl), d
00026      ld      hl, (Par2)
00027      ld      (hl), c
00028      inc     hl
00029      ld      (hl), b
00030      ld      hl, (Par3)
00031      push    ix
00032      pop     de
00033      ld      (hl), e
00034      inc     hl
00035      ld      (hl), d
00036      push    iy
00037      pop     de
00038      ld      hl, (par4)
00039      ld      (hl), e
00040      inc     hl
00041      ld      (hl), d
00042      end
```

3.15 Hdcout.for File Reference

CP/M: Printer spooler for 7-dot matrix printer Seikosha GP100VC.

Functions/Subroutines

- program [hdcout](#)

3.15.1 Detailed Description

CP/M: Printer spooler for 7-dot matrix printer Seikosha GP100VC.

Version

1.0

Author

(C) 1995 Dr.-Ing. Klaus Friedewald

Copyright

GNU LESSER GENERAL PUBLIC LICENSE Version 3

Definition in file [Hdcout.for](#).

3.15.2 Function/Subroutine Documentation

3.15.2.1 hdcout()

program hdcout

Definition at line 1 of file [Hdcout.for](#).

3.16 Hdcout.for

```
00001      program hdcout
00002 C Gibt saemtliche Files des Typs BILD?.HDC auf einem GP100VC aus
00003 C      Dr.-Ing. K. Friedewald
00004 C
00005      logical peekv, gethdc
00006      logical ishort, inilen, inistr(5), finlen, finstr(2)
00007      logical filnam(8)
00008      integer count
00009      integer*4 ibit, ishift
00010      logical ibitl(4), oldbyt
00011      integer ibit2(2)
00012      equivalence( ibit, ibitl(1), ibit2(1) )
00013      data inilen, inistr/5, 8, 27, 16, 0, 40/
00014      data finlen, finstr/2, 13, 15/
00015      data filnam/'B','I','L','D','A',' ',' ',' '/
00016
00017      call tcsini
00018      call initt
00019 C
00020 C Schleife ueber die Hardcopyfiles
00021 C
00022
```

```

00023 5      if (gethdc(filnam)) goto 99
00024
00025 c
00026 c Initialisierung des Druckers
00027 c
00028      write (2,200)
00029      do 10 ishort=1,inilen
00030 10      call print (4,inistr(ishort))
00031 c
00032 c Ausgabe
00033 c
00034      idone= 0
00035      ispalt= 0
00036 20      continue
00037      ishift= 2.** (9.-idone)+.5
00038      izeile= 199
00039      oldbyt= 0
00040      count= 0
00041 25      if (izeile.lt.0) goto 30
00042      ibit= 0
00043      ibit1(2)= peekv(izeile*80+ispalt)
00044      if (ispalt.lt.79) ibit1(1)= peekv(izeile*80+ispalt+1)
00045      ibit= ibit/ishift
00046 c
00047 c GP-100: Bit0 = oberste Nadel!
00048 c
00049      if (ibit1(1).lt.0) ibit2(1)= ibit2(1)-128
00050      ibit1(3)= 0
00051      ibit1(4)= 64
00052      do 27 ishort=1,7
00053      ibit1(3)= ibit1(3)+ mod(ibit1(1),2)*ibit1(4)
00054      ibit1(1)= ibit1(1)/2
00055      ibit1(4)= ibit1(4)/2
00056 27      continue
00057      ibit2(2)= ibit2(2)+128
00058 c
00059 c Druckwegoptimierung
00060 c
00061      if ((ibit1(3).eq.oldbyt).and.(count.le.252)) goto 33
00062      if (count.eq.0) goto 35
00063      call print (4,26)
00064      call print (4,count)
00065      call print (4,oldbyt)
00066      count= 0
00067 35      continue
00068      oldbyt= ibit1(3)
00069 33      continue
00070      count= count+2
00071      izeile= izeile-1
00072      goto 25
00073 30      continue
00074      idone= idone+7
00075      if (idone.lt.8) goto 40
00076      idone= idone-8
00077      ispalt= ispalt+1
00078 40      continue
00079      if (count.eq.0) goto 50
00080      call print (4,26)
00081      call print (4,count)
00082      call print (4,oldbyt)
00083 50      continue
00084      call print (4,13)
00085      if (ispalt.lt.80) goto 20
00086 c
00087 c Drucker auf Seitenanfang
00088 c
00089      do 60 i=1,finlen
00090 60      call print (4,finstr(i))
00091      write (2,200)
00092 c
00093 c Naechstes File
00094 c
00095      filnam(5)= filnam(5)+1
00096      goto 5
00097 99      call alpha
00098      write (1,110)
00099
00100 110      format (1x,'HDCout done.')
00101 200      format (1h0,///)
00102      end
00103

```

3.17 Hdcout1.asm File Reference

CP/M Port: Read VDC-RAM.

3.17.1 Detailed Description

CP/M Port: Read VDC-RAM.

Version

1.0

Author

(C) 1995 Dr.-Ing. Klaus Friedewald

Copyright

GNU LESSER GENERAL PUBLIC LICENSE Version 3

Definition in file [Hdcout1.asm](#).

3.18 Hdcout1.asm

```
00001      title    'TCS: Hilfsroutinen zur Druckerausgabe'
00002 ; Lesen des VDC-RAMS:
00003 ; LOGICAL P,PEEKV
00004 ; p= peekv(adr)
00005 ;
00006 ;      26.5.88      Dr.-Ing. K. Friedewald
00007 ;
00008 $*Macro
00009
00010      maclib    z80
00011      public    peekv
00012      name      'HDCou1'
00013
00014 tcstab equ     4000h
00015 jmpfar equ     0FFF1h
00016
00017 peekv mov      e,m
00018      inx      h
00019      mov      d,m
00020      lxi      h,4027h
00021      call     jmpfar
00022      mov      a,e
00023      ret
00024
00025      end
00026
```

3.19 Mainpage.dox File Reference

3.20 Print.asm File Reference

CP/M Port: Direct output to printer.

3.20.1 Detailed Description

CP/M Port: Direct output to printer.

Version

1.0

Author

(C) 1995 Dr.-Ing. Klaus Friedewald

Copyright

GNU LESSER GENERAL PUBLIC LICENSE Version 3

Definition in file [Print.asm](#).

3.21 Print.asm

```

00001 ; Subroutine PRINT (Adr+256*SecAdr, Char)
00002 ;   Ausgabe eines Buchstabens ohne Codewandlung auf dem Geraet
00003 ;   26.5.88           Dr.-Ing. K. Friedewald
00004     title   'Print# fuer CP/M'
00005     name    'Print'
00006     public print
00007
00008 bdos equ 5
00009 bios equ 50
00010
00011 parblk equ 0fd00h
00012 bufptr equ 0fe00h
00013 vicdrv equ parblk+2
00014 victrk equ parblk+3
00015 viccnt equ parblk+5
00016 vicdat equ parblk+6
00017
00018 cseg
00019 print mov a,m           ; Geraeteadresse
00020     sta vicdrv
00021     inx h
00022     mov a,m           ; Sek. Adr.
00023     sta victrk
00024     ldax d           ; auszugebender Buchstabe
00025     sta vicdat
00026
00027     mvi a,1           ; Anzahl Buchstaben
00028     sta viccnt
00029
00030     lxi h,vicdat       ; Bufferadresse, in Bank 0
00031     shld BufPtr
00032
00033     lxi d,BiosP
00034     mvi c,Bios
00035     jmp bdos           ; und Rueckkehr
00036
00037 dseg
00038 BiosP db 30           ; Bios: Userfunction
00039     db 4               ; 6502 Job
00040     dw 0               ; Dummy-Parameter
00041     dw 0
00042     dw 7               ; Jobnr: Print
00043     end
00044
```

3.22 TCS.for File Reference

TCS: Tektronix Plot 10 Emulation.

Functions/Subroutines

- subroutine [vcursr](#) (IC, X, Y)
- subroutine [drawr](#) (X, Y)
- subroutine [mover](#) (X, Y)
- subroutine [pointr](#) (X, Y)
- subroutine [dashr](#) (X, Y, iL)
- subroutine [rel2ab](#) (Xrel, Yrel, Xabs, Yabs)
- subroutine [drawa](#) (X, Y)
- subroutine [movea](#) (X, Y)
- subroutine [pointa](#) (X, Y)
- subroutine [dasha](#) (X, Y, iL)
- subroutine [wincot](#) (X, Y, IX, IY)
- subroutine [revcot](#) (IX, IY, X, Y)
- subroutine [anstr](#) (NChar, IStrin)
- subroutine [ancho](#) (ichar)
- subroutine [newlin](#)
- subroutine [cartn](#)
- subroutine [linef](#)
- subroutine [baksp](#)
- subroutine [newpag](#)
- function [linhgt](#) (Numlin)
- function [linwdt](#) (NumChr)
- subroutine [lintrn](#)
- subroutine [logtrn](#) (IMODE)
- subroutine [twindo](#) (IX1, IX2, IY1, IY2)
- subroutine [swindo](#) (IX, LX, IY, LY)
- subroutine [dwindo](#) (X1, X2, Y1, Y2)
- subroutine [vwindo](#) (X, XL, Y, YL)
- subroutine [rescal](#)
- subroutine [rrotat](#) (Grad)
- subroutine [rscale](#) (Faktor)
- subroutine [home](#)
- subroutine [setmrg](#) (Mlinks, Mrecht)
- subroutine [seetrm](#) (IBaud, Iterm, ICSIZE, MaxScr)
- subroutine [seetrn](#) (xf, yf, key)
- logical function [genflg](#) (ITEM)

3.22.1 Detailed Description

TCS: Tektronix Plot 10 Emulation.

Version

4.0

Author

(C) 2022 Dr.-Ing. Klaus Friedewald

Copyright

GNU LESSER GENERAL PUBLIC LICENSE Version 3

System independent subroutines

Definition in file [TCS.for](#).

3.22.2 Function/Subroutine Documentation

3.22.2.1 ancho()

```
subroutine ancho (
    ichar )
```

Definition at line [315](#) of file [TCS.for](#).

3.22.2.2 anstr()

```
subroutine anstr (
    NChar,
    dimension(1) IStrin )
```

Definition at line [305](#) of file [TCS.for](#).

3.22.2.3 baksp()

```
subroutine baksp
```

Definition at line [360](#) of file [TCS.for](#).

3.22.2.4 cartn()

```
subroutine cartn
```

Definition at line [341](#) of file [TCS.for](#).

3.22.2.5 dasha()

```
subroutine dasha (
    X,
    Y,
    iL )
```

Definition at line [266](#) of file [TCS.for](#).

3.22.2.6 dashr()

```
subroutine dashr (  
    X,  
    Y,  
    iL )
```

Definition at line 212 of file [TCS.for](#).

3.22.2.7 drawa()

```
subroutine drawa (  
    X,  
    Y )
```

Definition at line 233 of file [TCS.for](#).

3.22.2.8 drawr()

```
subroutine drawr (  
    X,  
    Y )
```

Definition at line 188 of file [TCS.for](#).

3.22.2.9 dwindo()

```
subroutine dwindo (  
    X1,  
    X2,  
    Y1,  
    Y2 )
```

Definition at line 438 of file [TCS.for](#).

3.22.2.10 genflg()

```
logical function genflg (  
    ITEM )
```

Definition at line 534 of file [TCS.for](#).

3.22.2.11 home()

```
subroutine home
```

Definition at line [494](#) of file [TCS.for](#).

3.22.2.12 linef()

```
subroutine linef
```

Definition at line [350](#) of file [TCS.for](#).

3.22.2.13 linhgt()

```
function linhgt (
    Numlin )
```

Definition at line [376](#) of file [TCS.for](#).

3.22.2.14 lintrn()

```
subroutine lintrn
```

Definition at line [394](#) of file [TCS.for](#).

3.22.2.15 linwdt()

```
function linwdt (
    NumChr )
```

Definition at line [384](#) of file [TCS.for](#).

3.22.2.16 logtrn()

```
subroutine logtrn (
    IMODE )
```

Definition at line [404](#) of file [TCS.for](#).

3.22.2.17 movea()

```
subroutine movea (  
    X,  
    Y )
```

Definition at line [244](#) of file [TCS.for](#).

3.22.2.18 mover()

```
subroutine mover (  
    X,  
    Y )
```

Definition at line [196](#) of file [TCS.for](#).

3.22.2.19 newlin()

```
subroutine newlin
```

Definition at line [333](#) of file [TCS.for](#).

3.22.2.20 newpag()

```
subroutine newpag
```

Definition at line [368](#) of file [TCS.for](#).

3.22.2.21 pointa()

```
subroutine pointa (  
    X,  
    Y )
```

Definition at line [255](#) of file [TCS.for](#).

3.22.2.22 pointr()

```
subroutine pointr (
    X,
    Y )
```

Definition at line [204](#) of file [TCS.for](#).

3.22.2.23 rel2ab()

```
subroutine rel2ab (
    Xrel,
    Yrel,
    Xabs,
    Yabs )
```

Definition at line [220](#) of file [TCS.for](#).

3.22.2.24 rescal()

```
subroutine rescal
```

Definition at line [457](#) of file [TCS.for](#).

3.22.2.25 revcot()

```
subroutine revcot (
    IX,
    IY,
    X,
    Y )
```

Definition at line [290](#) of file [TCS.for](#).

3.22.2.26 rrotat()

```
subroutine rrotat (
    Grad )
```

Definition at line [477](#) of file [TCS.for](#).

3.22.2.27 rscale()

```
subroutine rscale (
    Faktor )
```

Definition at line 486 of file [TCS.for](#).

3.22.2.28 seetrm()

```
subroutine seetrm (
    IBaud,
    Iterm,
    ICSize,
    MaxScr )
```

Definition at line 512 of file [TCS.for](#).

3.22.2.29 seetrn()

```
subroutine seetrn (
    xf,
    yf,
    key )
```

Definition at line 523 of file [TCS.for](#).

3.22.2.30 setmrg()

```
subroutine setmrg (
    Mlinks,
    Mrecht )
```

Definition at line 503 of file [TCS.for](#).

3.22.2.31 swindo()

```
subroutine swindo (
    IX,
    LX,
    IY,
    LY )
```

Definition at line 426 of file [TCS.for](#).

3.22.2.32 twindo()

```
subroutine twindo (  
    IX1,  
    IX2,  
    IY1,  
    IY2 )
```

Definition at line 419 of file [TCS.for](#).

3.22.2.33 vcursr()

```
subroutine vcursr (  
    IC,  
    X,  
    Y )
```

Definition at line 178 of file [TCS.for](#).

3.22.2.34 vwindo()

```
subroutine vwindo (  
    X,  
    XL,  
    Y,  
    YL )
```

Definition at line 445 of file [TCS.for](#).

3.22.2.35 wincot()

```
subroutine wincot (  
    X,  
    Y,  
    IX,  
    IY )
```

Definition at line 277 of file [TCS.for](#).

3.23 TCS.for

```
00001 C> \file TCS.for
00002 C> \brief TCS: Tektronix Plot 10 Emulation
00003 C> \version 4.0
00004 C> \author (C) 2022 Dr.-Ing. Klaus Friedewald
00005 C> \copyright GNU LESSER GENERAL PUBLIC LICENSE Version 3
00006 C> \~german
00007 C> Systemübergreifende TCS-Routinen
00008 C> \~english
00009 C> System independent subroutines
00010 C> \~
00011 C
00012 C CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC Changelog CCCCCCCCCCCCCCCCCCCCCCCCCCCC
00013 C
00014 C 27.11.20 Version 4.0:
00015 C Einheitliche Version CPM/DOS/Windows/SDL2
00016 C
00017 C 17.08.20 Version 3.2
00018 C Harmonisierung der Verwendung des Commonblocks TKTRNX
00019 C Variable KHOMEY wird jetzt (analog alter DOS-Version) verwendet.
00020 C Da KHOMEY nicht in der CP/M Version vorhanden ist, muss ab dieser
00021 C Version fuer eine Compilation unter CP/M die entsprechende Zeile
00022 C in der SUBROUTINE HOME geändert werden.
00023 C
00024 C 13.11.17 Version 3.1
00025 C Anpassung an OpenWatcom 2.0
00026 C Bugfix: Unterscheidung Aufrufe ueber windowsx.h (winl6) und GDI (win32)
00027 C - SelectPen -> SelectObject
00028 C - DeletePen -> DeleteObject
00029 C - DeleteBrush -> DeleteObject
00030 C - GetStockBrush -> GetStockObject
00031 C - DeleteRgn -> DeleteObject
00032 C - SelectFont -> SelectObject
00033 C - DeleteFont -> DeleteObject
00034 C
00035 C 27.03.13 Version 3.0
00036 C Anpassung an Windows 7 und OpenWatcom 1.9
00037 C Anpassung an gfortran anstelle von g77 der GCC
00038 C
00039 C 22.12.05 Version 2.19
00040 C Elimination berechnetes GOTO in LOGTRN
00041 C
00042 C 18.10.05 Version 2.18
00043 C Anpassung der Windowsversionen zur gemeinsamen Verwendung SDL2:
00044 C TCSDrWIN.for
00045 C TCSDWInc.h
00046 C - Überfuehrung der Deklaration aus TCSDwIN.c nach *.h:
00047 C GraphicError und CreateMainWindow_IfNecessary
00048 C - Definition der Fehlernummern als Konstante statt enum
00049 C Abhaengigkeit Watcom-Defaultwindowssystem eliminiert
00050 C - TCSDWInc.c: Kein Abbruch bei OpenWatcom > 1.3 und
00051 C definiertem Symbol trace_calls
00052 C
00053 C 26.10.04 Version 2.17
00054 C Bugfix Windows-System: Größe und Defaultposition des Status-
00055 C fensters wird bei der Erzeugung berechnet -> 1. RESTORE nach
00056 C Verkleinern des Graphikfensters entspricht dem vorherigen
00057 C Bild. 2. Angleichung des Verhaltens von 16- und 32bit Windows
00058 C Bei Definition des Symbols STAT_WINDOW_PRIVATE erhält das
00059 C Statusfenster einen privaten Devicekontext.
00060 C Zusammenfuehrung Initialisierung der Windows-Library und
00061 C Windows-DLL -> zusätzliche Sourcefiles
00062 C TCSinitt.for, CreateMainWindow.c, GetMainInstance.c
00063 C
00064 C 23.06.04 Version 2.16:
00065 C Anpassungen an GNU-Compiler fuer Win32. Zusätzliches Sourcefile
00066 C fuer die GNU-Version: WinMain.c
00067 C CSIZE in Windows-Version: Korrektur Rundungsfehler
00068 C
00069 C 08.06.04 Version 2.15:
00070 C Umbenennung lib$movc3 in lib_movc3 (entsprechend ANSI-Fortran)
00071 C Modul STRINGS.FOR: Version 1.24
00072 C
00073 C 27.06.03 Version 2.14:
00074 C Verarbeitung Steuerzeichen in ANCHO
00075 C
00076 C 21.10.02 Version 2.13:
00077 C Einheitliche Version CPM/DOS/Windows
00078 C
00079 C CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
00080 C
00081 C Grundversion fuer C128 / Version 1.0:
00082 C
00083 C Zugehoerige Module:
00084 C TKTRNX.FOR Common-Block TKTRNX
00085 C TCSBASIC.ASM Low-Level Routinen in Bank 0, C128 spezifisch
```

```

00086 C          TCSDRIVR.ASM  Treiber fuer TCSBASIC
00087 C          TCSGIN.ASM    Treiber des Gin-Cursors
00088 C
00089 C      20.4.88          Dr.-Ing. K. Friedewald
00090 C                      4000 Duesseldorf 1
00091 C                      Gerresheimerstr. 84
00092 C
00093 C      21.10.02 Version 2.13:
00094 C          Vereinheitlichung CPM/DOS/Windowsversion
00095 C          Zusätzliches Modul: TCSdrCPM.FOR: früher Teil von TCS.FOR
00096 C          Ausschließliche Verwendung von durch grosses "C" eingeleiteten
00097 C          Kommentaren zur Kompatibilität mit FORTRAN 4
00098 C          Umbenennung des Includefiles in Tktrnx.fd. So kann unter CP/M
00099 C          das als Teil des Filenamens interpretierte "" der INCLUDE-
00100 C          Anweisung entsprechend der 8.3 Filenamen umgesetzt werden.
00101 C          Implementierung Unterprogramm TCSLEV
00102 C          Bugfix: Kommentar in Tktrnx.fd wurde falsch gekennzeichnet
00103 C                  (c statt C) -> SVSTAT und RESTAT fehlerhaft, da nicht
00104 C                  erkannte Kommentare zusaetzliche Variablen erzeugten.
00105 C
00106 C          TBD: Implementierung vertikale Auflösung von 400 Pixeln
00107 C
00108 C          CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
00109 C
00110 C      Anpassung an DOS:
00111 C
00112 C          Änderungen gegenüber CP/M-Version:
00113 C          SEELOC, DCURSR, SVSTAT, RESTAT, CSIZE in TCSdrDOS.FOR
00114 C      Bugfix: DASHA, DASHR - Korrektur Parameterliste
00115 C          SEETRM - ibaud statt ibaudr
00116 C
00117 C      Zugehörige Module:
00118 C          TKTRNX.FOR      Common-Block TKTRNX
00119 C          TCSdrDOS.FOR    Bildschirmtreiber
00120 C          TCSdDOSa.ASM    Betriebssystemspezifische Low-Level Routinen
00121 C          HDCOPY.FOR      Hardcopyroutine
00122 C          STRINGS.FOR     Hilfsroutinen zur Stringverarbeitung
00123 C          OUTTEXT.FOR     nur für WATCOM-Compiler
00124 C
00125 C      25.10.01 Version 2.00: Dr.-Ing. K. Friedewald
00126 C
00127 C      07.02.02 Version 2.10:
00128 C          Implementierung multilinguale Fehlermeldungen
00129 C
00130 C      11.10.02 Version 2.12:
00131 C          Vereinheitlichung DOS/Windowsversion
00132 C
00133 C          CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
00134 C
00135 C      Anpassungen an Microsoft-Windows:
00136 C
00137 C          Änderungen gegenüber DOS-Version:
00138 C          INITT befinden sich jetzt in TCSdrWIN.FOR bzw. TCSinitt.FOR
00139 C
00140 C      Zugehörige Module:
00141 C          TKTRNX.FOR      Common-Block TKTRNX
00142 C          TKTRNX.h        Common-Block TKTRNX für Zugriff durch C
00143 C          TCSdrWIN.FOR    Bildschirmtreiber
00144 C          TCSdWINc.c      Windowspezifische API-Routinen
00145 C          TCSdWINc.h      Compiler- und systemspezifische Deklarationen
00146 C          STRINGS.FOR     Hilfsroutinen zur Stringverarbeitung
00147 C
00148 C      27.10.01 Version 2.11: Dr.-Ing. K. Friedewald
00149 C
00150 C      11.10.02 Version 2.12:
00151 C          Vereinheitlichung DOS/Windowsversion
00152 C
00153 C
00154 C          CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
00155 C
00156 C      Anpassungen an SDL2:
00157 C
00158 C          Änderungen gegenüber Windows-Version:
00159 C          Fehlerausgabe in den Windows-Debug-Channel (bzw. *ix Fehlerkanal)
00160 C          Statusfenster analog DOS nur einzeilig ohne Scrollmöglichkeit
00161 C
00162 C      Zugehörige Module:
00163 C          TKTRNX.FOR      identisch mit Windows-Version
00164 C          TKTRNX.h        identisch mit Windows-Version
00165 C          TCSdrSDL.FOR    SDL2-spezifische API-Routinen
00166 C          TCSdSDLc.c      SDL2-spezifische API-Routinen
00167 C          TCSdSDLc.h      Compiler- und systemspezifische Deklarationen
00168 C          STRINGS.FOR     identisch mit Windows-Version
00169 C
00170 C      27.11.20 Version 4.00: Dr.-Ing. K. Friedewald
00171 C
00172 C

```



```

00173
00174 C
00175 C Graphic Input
00176 C
00177
00178     subroutine vcursr (IC,X,Y)
00179     call dcursr (ic,ix,iy)
00180     call revcot (ix,iy,x,y)
00181     return
00182 end
00183
00184 C
00185 C Virtuelle Graphik, relativ
00186 C
00187
00188     subroutine drawr (X,Y)
00189     call rel2ab (x,y,xabs,yabs)
00190     call drawa (xabs,yabs)
00191     return
00192 end
00193
00194
00195
00196     subroutine mover (X,Y)
00197     call rel2ab (x,y,xabs,yabs)
00198     call movea (xabs,yabs)
00199     return
00200 end
00201
00202
00203
00204     subroutine pointr (X,Y)
00205     call rel2ab (x,y,xabs,yabs)
00206     call pointa (xabs,yabs)
00207     return
00208 end
00209
00210
00211
00212     subroutine dashr (X,Y, iL)
00213     call rel2ab (x,y,xabs,yabs)
00214     call dasha (xabs,yabs, il)
00215     return
00216 end
00217
00218
00219
00220     subroutine rel2ab (Xrel, Yrel, Xabs, Yabs)
00221     include 'Tktrnx.fd'
00222     call seeloc (ix,iy)
00223     call revcot (ix,iy,xabs,yabs)
00224     xabs= (( xrel*trcosf - yrel*trsinf)*trscal)+xabs
00225     yabs= (( xrel*trsinf + yrel*trcosf)*trscal)+yabs
00226     return
00227 end
00228
00229 C
00230 C Virtuelles Zeichnen, absolut
00231 C
00232
00233     subroutine drawa (X,Y)
00234     include 'Tktrnx.fd'
00235     call wincot (x,y,ix,iy)
00236     call swindl (kminsx,kminsy,kmaxsx,kmaxsy)
00237     call drwabs (ix,iy)
00238     call swindl (0,0,1023,780)
00239     return
00240 end
00241
00242
00243
00244     subroutine movea (X,Y)
00245     include 'Tktrnx.fd'
00246     call wincot (x,y,ix,iy)
00247     call swindl (kminsx,kminsy,kmaxsx,kmaxsy)
00248     call movabs (ix,iy)
00249     call swindl (0,0,1023,780)
00250     return
00251 end
00252
00253
00254
00255     subroutine pointa (X,Y)
00256     include 'Tktrnx.fd'
00257     call wincot (x,y,ix,iy)
00258     call swindl (kminsx,kminsy,kmaxsx,kmaxsy)
00259     call pntabs (ix,iy)

```

```

00260      call swindl (0,0,1023,780)
00261      return
00262      end
00263
00264
00265
00266      subroutine dasha (X,Y, iL)
00267      include 'Tktrnx.fd'
00268      call wincot (x,y,ix,iy)
00269      call swindl (kminsx,kminsy,kmaxsx,kmaxsy)
00270      call dshabs (ix,iy, iL)
00271      call swindl (0,0,1023,780)
00272      return
00273      end
00274
00275
00276
00277      subroutine wincot (X,Y,IX,IY)
00278      include 'Tktrnx.fd'
00279      dx= x-tminvx
00280      dy= y-tminvy
00281      if ((xlog.lt.255.).and.(x.gt.0.)) dx= alog(x)-xlog
00282      if ((ylog.lt.255.).and.(y.gt.0.)) dy= alog(y)-ylog
00283      ix= ifix(dx*xfac+.5)+kminsx
00284      iy= ifix(dy*yfac+.5)+kminsy
00285      return
00286      end
00287
00288
00289
00290      subroutine revcot (IX,IY,X,Y)
00291      include 'Tktrnx.fd'
00292      dx= float(ix-kminsx) / xfac
00293      dy= float(iy-kminsy) / yfac
00294      x= dx + tminvx
00295      y= dy + tminvy
00296      if (xlog.lt.255.) x= 2.718282**(dx+xlog)
00297      if (ylog.lt.255.) y= 2.718282**(dy+ylog)
00298      return
00299      end
00300
00301 C
00302 C Alphanumerische Ausgabe
00303 C
00304
00305      subroutine anstr (NChar, IStrin)
00306      dimension istrin(1)
00307      do 10 i=1,nchar
00308          call ancho (istrin(i))
00309 10      continue
00310      return
00311      end
00312
00313
00314
00315      subroutine ancho (ichar)
00316      include 'Tktrnx.fd'
00317
00318      if (ichar.gt.31) goto 10
00319      if (ichar.eq.7) call bell
00320      if (ichar.eq.10) call linef
00321      if (ichar.eq.13) call cartn
00322      return
00323
00324 10      call seeloc (ix,k)
00325      call csize (ixlen,k)
00326      if (ix.gt.krmrgn-ixlen) call newlin
00327      call toutpt (ichar)
00328      return
00329      end
00330
00331
00332
00333      subroutine newlin
00334      call cartn
00335      call linef
00336      return
00337      end
00338
00339
00340
00341      subroutine cartn
00342      include 'Tktrnx.fd'
00343      call seeloc (ix,iy)
00344      call movabs (klmrgn,iy)
00345      return
00346      end

```

```

00347
00348
00349
00350     subroutine linef
00351     call seeloc (j,iy)
00352     call csize (j,iylen)
00353     if (iy.lt.iylen) call home
00354     call movrel (0,-iylen)
00355     return
00356     end
00357
00358
00359
00360     subroutine baksp
00361     call csize (ix,iy)
00362     call movrel (-ix,0)
00363     return
00364     end
00365
00366
00367
00368     subroutine newpag
00369     call erase
00370     call home
00371     return
00372     end
00373
00374
00375
00376     function linhgt (Numlin)
00377     call csize (ix,iy)
00378     linhgt= numlin*iy
00379     return
00380     end
00381
00382
00383
00384     function linwdt (NumChr)
00385     call csize (ix,iy)
00386     linwdt= numchr*ix
00387     return
00388     end
00389
00390 C
00391 C   Initialisierungsroutinen
00392 C
00393
00394     subroutine lintrn
00395     include 'Tktrnx.fd'
00396     xlog= 255.
00397     ylog= 255.
00398     call rescal
00399     return
00400     end
00401
00402
00403
00404     subroutine logtrn (IMODE)
00405     include 'Tktrnx.fd'
00406     call lintrn
00407     if ((imode .eq. 1) .or. (imode .eq. 3)) then
00408         xlog= 0.
00409     end if
00410     if ((imode .eq. 2) .or. (imode .eq. 3)) then
00411         ylog= 0.
00412     end if
00413     call rescal
00414     return
00415     end
00416
00417
00418
00419     subroutine twindo (IX1,IX2,IY1,IY2)
00420     call swindo (ix1,ix2-ix1,iy1,iy2-iy1)
00421     return
00422     end
00423
00424
00425
00426     subroutine swindo (IX,LX,IY,LY)
00427     include 'Tktrnx.fd'
00428     kminsx= ix
00429     kmaxsx= ix+lx
00430     kminsy= iy
00431     kmaxsy= iy+ly
00432     call rescal
00433     return

```

```

00434     end
00435
00436
00437
00438     subroutine dwindo (X1,X2,Y1,Y2)
00439     call vwindo (x1,x2-x1,y1,y2-y1)
00440     return
00441     end
00442
00443
00444
00445     subroutine vwindo (X,XL,Y,YL)
00446     include 'Tktrnx.fd'
00447     tminvx= x
00448     tmaxvx= x+xl
00449     tminvy= y
00450     tmaxvy= y+yl
00451     call rescal
00452     return
00453     end
00454
00455
00456
00457     subroutine rescal
00458     include 'Tktrnx.fd'
00459     xfac= 0.
00460     yfac= 0.
00461     if ((tmaxvx.eq.tminvx) .or. (tmaxvy.eq.tminvy)) return
00462     dx= tmaxvx-tminvx
00463     dy= tmaxvy-tminvy
00464     if ((xlog.eq.255.) .or. (amin1(tminvx,tmaxvx).le.0.)) goto 10
00465     xlog= alog(tminvx)
00466     dx= alog(tmaxvx)-xlog
00467 10    if ((ylog.eq.255.) .or. (amin1(tminvy,tmaxvy).le.0.)) goto 20
00468     ylog= alog(tminvy)
00469     dy= alog(tmaxvy)-ylog
00470 20    xfac= float(kmaxsx-kminsx) / dx
00471     yfac= float(kmaxsy-kminsy) / dy
00472     return
00473     end
00474
00475
00476
00477     subroutine rrotat (Grad)
00478     include 'Tktrnx.fd'
00479     trsinf= sin(grad/57.29578)
00480     trcosf= cos(grad/57.29578)
00481     return
00482     end
00483
00484
00485
00486     subroutine rscale (Faktor)
00487     include 'Tktrnx.fd'
00488     trscal= faktor
00489     return
00490     end
00491
00492
00493
00494     subroutine home
00495     include 'Tktrnx.fd'
00496 C    call movabs(klmg,750) Fuer CP/M (kein khomey verfuegbar, -> !=750)
00497     call movabs(klmg,khomey)
00498     return
00499     end
00500
00501
00502
00503     subroutine setmrg (Mlinks, Mrecht)
00504     include 'Tktrnx.fd'
00505     klmg= mlinks
00506     krmrg= mrecht
00507     return
00508     end
00509
00510
00511
00512     subroutine seetrm (IBaud, Iterm, ICSIZE, MaxScr)
00513     include 'Tktrnx.fd'
00514     ibaud= 0
00515     iterm= 1
00516     icsize= 1
00517     maxscr= 1023
00518     return
00519     end
00520

```

```

00521
00522
00523     subroutine seetrn (xf,yf,key)
00524     include 'Tktrnx.fd'
00525     xf= xfac
00526     yf= yfac
00527     key= 1
00528     if ((xlog.lt.255.).or.(ylog.lt.255.)) key=2
00529     return
00530     end
00531
00532
00533
00534     logical function genflg (ITEM)
00535     genflg= item.eq.0
00536     return
00537     end
00538

```

3.24 Tcsbasi1.asm File Reference

CP/M Port: Basic Graphic Routines Bank 0.

3.24.1 Detailed Description

CP/M Port: Basic Graphic Routines Bank 0.

Version

2.2

Author

(C) 1995 Dr.-Ing. Klaus Friedewald

Copyright

GNU LESSER GENERAL PUBLIC LICENSE Version 3

Definition in file [Tcsbasi1.asm](#).

3.25 Tcsbasi1.asm

```

00001     title    'C128 Grundgraphik'
00002 ;
00003 ;     Graphikgrundpaket fuer die Ansteuerung des C128 VDC-Chips
00004 ;
00005 ;     Das Paket wird durch das Unterprogramm TCSinit in die Bank 0 kopiert.
00006 ;     Anschliessend werden die einzelnen Funktionen ueber
00007 ;         call jmpfar
00008 ;     ausgefuehrt. Die gewuenschte Funktion und die Parameter werden in den
00009 ;     Prozessorregistern uebergeben. Hierbei enthaelt HL die Sprungadresse,
00010 ;     die uebrigen Register DE, BC, IX, IY die max. 4 Parameter.
00011 ;
00012 ;     In den Treiberroutinen muessen die Konstanten
00013 ;         TcsTab = 4000h
00014 ;         und   JmpFar = FFF1h
00015 ;     verwendet werden.
00016 ;
00017 ;     12.5.88         Dr.-Ing. K. Friedewald
00018 ;
00019 ;     Version 2.0     4.1.94

```

```

00020 ;      - Implementation des i2c-Bustreibers
00021 ;      - Versionsidentifikation
00022 ;      Version 2.1      7.3.95
00023 ;      - Beruecksichtigung 64 k VDC-RAM
00024 ;      - VDCMove, VDCFill, VDCRegR, VDCRegW, VDCinit
00025 ;
00026 ;      Dr. K. Friedewald
00027 ;
00028 ;      Version 2.2      8.12.20
00029 ;      - Entfernung Code i2c-Bustreiber
00030 ;
00031 $*Macro
00032
00033      public  TCSinit
00034      name    'TCSinit'
00035
00036      maclib  z80
00037      maclib  x6502
00038
00039      vers    equ    2h      ; Versionsnummer
00040      release equ    1h
00041
00042      VDC64k equ    1      ; 0=16k, 1=64k RAM
00043
00044      i2cflg equ    0      ; i2c-Bustreiber wird installiert (bis V2.1)
00045
00046      ZModChr equ    05bdh  ; Modifiziere Zeichensatz (ROM-Routine)
00047      ZMovVDC equ    08d8h  ; VDC-Move (ROM-Routine)
00048      ZFillVDC equ    086ch  ; VDC-Fill (ROM-Routine)
00049      ; Lage ROM-Routine unbekannt: equ 0
00050
00051      BiosVector equ 0f000h  ; BDOS (50,25) setzt XMOVE zurueck!
00052      ; equ 0: BiosVector wird RUNTIME berechnet
00053
00054      xmove    equ    BiosVector+3*29
00055      move     equ    BiosVector+3*25
00056
00057      TcsTab   equ    4000h  ; Lage in Bank 0
00058
00059      MMU      equ    0d500h
00060      confreg  equ    0ff00h  ; Konfigurationsregister
00061      ConfB0   equ    3fh    ; Bank 0
00062      ConfB1   equ    7fh    ; Bank 1
00063      ConfBI   equ    3eh    ; Bank 0 + IO
00064      ConfKI   equ    0eh    ; Kernal + IO
00065      ConfKC   equ    0fh    ; Kernal + CharROM
00066
00067      CharRom  equ    0d800h  ; Character-Rom
00068      go65     equ    0ffe0h
00069      cnt65    equ    0ffddh  ; Startvektor 6502
00070      go80     equ    0ffd0h
00071      cnt80    equ    0ffeeh  ; Startbefehl Z80
00072      JmpFar   equ    cnt80+3
00073
00074      richtung equ 0000h      ; Prozessorport 6502
00075      port     equ 0001h      ; fuer I2C
00076      vic      equ 0d000h
00077      btim      equ 10        ; Zeitkonstanten
00078      ptim      equ 20        ; fuer I2C
00079
00080      dlchr     equ    0ff62h  ; Kopieren des Charrom (8502)
00081
00082      SelReg    equ    0189h  ; Anwahl VDC-Register
00083      VDCRam    equ    0186h  ; Update VDC-Ram
00084      FilVdc   equ    07Ah    ; Fuellen VDC-Ram
00085      VicCode   equ    70h    ; Codewandlung Ascii-Cbm
00086      print     equ    68h    ; print text
00087
00088      ChrH      equ    8      ; Hoehe der Buchstaben
00089      GinW      equ    16     ; Breite Gin Cursor
00090      GinH      equ    8      ; Hoehe GinCrs.
00091
00092      rcall     macro  FuncNr  ; Aufruf Z80-ROM
00093      mvi      1,FuncNr
00094      rst      5
00095      endm
00096
00097      call165   macro  adr     ; Unterprogrammaufruf fuer 8502
00098      lxi      d,adr
00099      call     enabl65
00100      endm
00101
00102      LoadVek  macro  adr     ; Laden 2-Byte Vektor
00103      lded     adr
00104      lbcd     adr+2
00105      endm
00106

```

```

00107 ImmVek macro valx, valy ; Setzen 2-Byte Vektor
00108 lxi d, valx
00109 lxi b, valy
00110 endm
00111
00112 StorVek macro adr ; Abspeichern 2-Byte Vektor
00113 sded adr
00114 sbcd adr+2
00115 endm
00116
00117 dsub macro DubReg
00118 ana a ; clear Carry
00119 dsbc DubReg
00120 endm
00121
00122 FlVDC macro Start, Len, Val
00123 lxi h, Start
00124 lxi b, Len
00125 mvi d, Val
00126 push h
00127 rcall FlVDC
00128 endm
00129
00130 page
00131 ;
00132 ; Start Initialisierung
00133 ;
00134 TCSinit IF BiosVector eq 0
00135 lhld l
00136 lxi d, Move-3
00137 dad d
00138 shld MoveVc+1
00139 lxi d, XMove-Move
00140 dad d
00141 shld XMoveVc+1
00142 ENDIF
00143 lxi b, JmpFar2-JmpFar1 ; Installation JMPFAR
00144 lxi d, JmpFar ; Destination
00145 lxi h, JmpFar1 ; Source
00146 ldir
00147
00148 lxi d, StartBnk0
00149 lxi h, TcsTab
00150 CopLp push d
00151 push h
00152 lxi h, EndBnk0
00153 dsub d
00154 jz done
00155 push h ; noch zu kopierende Bytes
00156 lxi d, 128 ; Maximal 128 Bytes / MOVE
00157 dsub d
00158 jm Rest ; NByte<128
00159 pop h
00160 push d ; Kopiere 128 Bytes
00161
00162 Rest mvi b, 0 ; Kopieren Code in Bank 0
00163 mvi c, 1 ; aus Bank 1
00164 XMoveVc call xmove
00165 pop b
00166 pop h
00167 pop d
00168 MoveVc call move
00169
00170 jmp CopLp
00171
00172 done pop h ; Restore Stack
00173 pop h
00174
00175 ret
00176 ;
00177 ; Kopie JMPFAR
00178 ;
00179 jmpfar1 mvi a, ConfB0 ; Bank 0
00180 ana a ; Clear Carry => JMP
00181 RetFar equ $-jmpfar1+jmpfar
00182 sta confreg
00183 rc
00184 jmp dojob
00185 jmpfar2:
00186 ;
00187 ; Arbeitsspeicher Bank 1
00188 ;
00189 dseg
00190 SouAdd dw StartBnk0
00191 DstAdd dw TcsTab
00192 NByte dw EndBnk0-StartBnk0
00193 cseg

```

```

00194         page
00195 ;
00196 ; Code fuer Bank 0
00197 ;
00198 StartBnk0:
00199 ;
00200 ; Sprungtabelle
00201 ;
00202         jmp     Graphic
00203         jmp     Alpha      ; TcsTab + $03
00204         jmp     Colour    ; + $06
00205         jmp     SetPnt    ; + $09
00206         jmp     ClrPnt    ; + $0c
00207         jmp     Line      ; + $0f
00208         jmp     movabs    ; + $12
00209         jmp     MovRel    ; + $15
00210         jmp     DrwRel    ; + $18
00211         jmp     CharOut   ; + $1b
00212         jmp     Window   ; + $1e
00213         jmp     SetChar   ; + $21
00214         jmp     ChkPnt   ; + $24
00215         jmp     PeekVDC   ; + $27
00216         jmp     PokeVDC   ; + $2a
00217         jmp     Where     ; + $2d
00218         jmp     GetWin    ; + $30
00219         jmp     erase     ; + $33
00220         jmp     GinCrs    ; + $36
00221         jmp     GinMrl    ; + $39
00222         jmp     i2cBus    ; + $3c
00223         jmp     version   ; + $3f
00224         jmp     VDCmove   ; + $42
00225         jmp     VDCfill   ; + $45
00226         jmp     VDCRegR   ; + $48
00227         jmp     VDCRegW   ; + $4b
00228         jmp     VDCinit   ; + $4e
00229 ;
00230 ; Start aller Jobs
00231 ;
00232 dojob    equ     $-StartBnk0+TcsTab
00233         sspd     OldStack      ; Neuer Stack in Bank 0!
00234         lxi      sp,NewStack
00235         pchl     ; Sprung in Tabelle
00236 ;
00237 ; Einschalten Graphic-Mode
00238 ;
00239 Graphic  equ     $-StartBnk0+TcsTab
00240         mvi      a,25      ; VDC-Register
00241         call     SelReg
00242         inp      d
00243         setb     7,d        ; Graphic ein
00244         res      6,d        ; Attribut aus
00245         call     selreg
00246         outp     d
00247
00248         call     init
00249         call     eras
00250
00251         mvi      e,0fh      ; weisser Vordergrund
00252         mvi      c,0        ; schwarzer Hintergrund
00253         jmp      Colour    ; und zurueck Bank 0
00254 ;
00255 ; Abschalten Graphic-Mode, Kopieren Zeichensatz
00256 ;
00257 alpha    equ     $-StartBnk0+TcsTab
00258         mvi      a,25      ; VDC-Register
00259         call     SelReg
00260         inp      d
00261         res      7,d        ; Text-Mode
00262         setb     6,d        ; Attribute
00263         call     selreg
00264         outp     d
00265
00266         call165  dlchr1
00267         call     ModChar    ; Anpassung PET-ASCII
00268
00269         rcall    print
00270         db       0ffh,0    ; CRT loeschen
00271
00272         mvi      e,0fh      ; weisser Vordergrund
00273         mvi      c,0h       ; Schwarzer Hintergrund
00274         jmp      Colour    ; und zurueck Bank 0
00275
00276 dlchr1    equ     $-StartBnk0+TcsTab
00277         @lda     ConfBI,#    ; IO einblenden
00278         @sta     ConfReg
00279
00280         @lda     MMU+08h     ; Verlegen Zero-Page

```



```

00281      @sta    dlchr2+1      ; Merken fuer Rueckkehr
00282      @and    11110000b,#   ; Bank 0
00283      @ldx    ZerPg65,#
00284      @sta    MMU+08h
00285      @stx    MMU+07h
00286      @inx                      ; sowie Stack
00287      @lda    MMU+0ah
00288      @sta    dlchr3+1
00289      @and    11110000b,#
00290      @sta    MMU+0ah
00291      @stx    MMU+09h
00292
00293      @lda    ConfKI,#        ; Kopieren Zeichensatz
00294      @sta    ConfReg
00295      @jsr    dlchr
00296      @lda    ConfBI,#
00297      @sta    ConfReg
00298
00299  dlchr2  equ    $-StartBnk0+TcsTab
00300      @lda    0,#           ; wird ueberschrieben
00301      @sta    MMU+08h
00302      @ldx    0,#           ; alte Zeropage und Stack
00303      @stx    MMU+07h
00304  dlchr3  equ    $-StartBnk0+TcsTab
00305      @lda    0,#
00306      @sta    MMU+0ah
00307      @inx
00308      @stx    MMU+09h
00309
00310      @lda    ConfB0,#        ; RAM
00311      @sta    ConfReg
00312      @jmp    go80
00313 ;
00314 ; Setzen der Zeichen- und Hintergrundfarbe
00315 ;
00316  Colour  equ    $-StartBnk0+TcsTab
00317      push    b              ; Hintergrund
00318
00319      mvi     a,26            ; Lesen alter Wert
00320      call    selreg
00321      inp     a
00322
00323      sta     tmp             ; alte Farben
00324      call    MakeCol
00325      pop     d
00326      call    MakeCol ; dito fuer Background
00327      lda     tmp+1
00328      mov     d,a
00329      mvi     a,26            ; Schreiben neuer Wert
00330      call    selreg
00331      outp    d
00332      jmp     return
00333
00334  MakeCol  equ    $-StartBnk0+TcsTab
00335      lxi     h,tmp
00336      rld                      ; A: alterWert
00337      bit     4,e             ; Neuer Wert=16?
00338      jrnz    OldCol
00339      mov     a,e
00340  OldCol  inx     h          ; Einschieben in tmp+1
00341      rld
00342      ret
00343 ;
00344 ; Setzen/Loeschen eines Punktes
00345 ;
00346  SetPnt  equ    $-StartBnk0+TcsTab
00347      call    spnt
00348      jmp     return
00349
00350  clrpnt  equ    $-StartBnk0+TcsTab
00351      call    cpnt
00352      jmp     return
00353 ;
00354 ; Ziehen einer Linie
00355 ;
00356  Line    equ    $-StartBnk0+TcsTab
00357      sixd    maske           ; Maske, ueberschreibt xend!
00358      call    Lin
00359      jmp     return
00360 ;
00361 ; Move Turtle
00362 ;
00363  movabs  equ    $-StartBnk0+TcsTab
00364      StorVek xpos
00365      jmp     return
00366 ;
00367 ; Move Turtle relativ

```

```

00368 ;
00369 MovRel equ    $-StartBnk0+TcsTab
00370      call    MovRl
00371      jmp     return
00372 ;
00373 ; Draw relativ
00374 ;
00375 DrwRel equ    $-StartBnk0+TcsTab
00376      sixd    maske    ; Maske, ueberschreibt xend!
00377      call    DrwRl
00378      jmp     return
00379 ;
00380 ; Ausgabe eines Buchstabens
00381 ;
00382 CharOut equ    $-StartBnk0+TcsTab
00383      mov     a,e
00384      cpi     20h      ; Blank
00385      jp      NoCntrl
00386      cpi     0dh      ; CR
00387      jnz     return
00388      ldcd    xmin
00389      lbcd    ypos
00390      lxi     h,ChrH
00391      dad     b
00392      mov     b,h
00393      mov     c,l
00394      jmp     movabs
00395
00396 NoCntrl equ    $-StartBnk0+TcsTab
00397      mov     b,a
00398      rcall   VicCode ; Berechnung Adresse Bitmuster
00399      mvi     h,0
00400      mov     l,a
00401      dad     h      ; *2
00402      dad     h      ; *4
00403      dad     h      ; *8
00404      lxi     d,CharRom
00405      dad     d      ; Adresse Bitmuster ok
00406      shld    ChrAdr
00407      call65   GetChar
00408
00409      LoadVek CVek1
00410      call    MovRl
00411      lxi     h,chrmsk
00412      shld    chrptr
00413 chrloop lhld    chrptr ; relatives label
00414      mov     a,m
00415      sta     maske
00416      LoadVek CVek2
00417      call    DrwRl
00418      lhld    chrptr
00419      inx     h
00420      shld    chrptr
00421      lxi     d,ChrMsk+8
00422      ana     a      ; Clear Carry
00423      dsbc    d
00424      jrzc    cdone
00425      LoadVek CVek3
00426      call    movrl
00427      jr      chrloop
00428 Cdone   jmp     return
00429
00430 GetChar equ    $-StartBnk0+TcsTab
00431      @lda    ConfKC,#
00432      @sta    confreg
00433      @ldx    7,#
00434 ChrLp   @lda    CharRom,x      ; Adresse wird veraendert!
00435 ChrAdr  equ    ChrLp+1-StartBnk0+TcsTab
00436      @sta    chrmsk,x
00437      @dex
00438      @bpl    chrLp
00439      @lda    ConfB0,#
00440      @sta    confreg
00441      @jmp    go80
00442 ;
00443 ; Setzen des Bildschirmfensters
00444 ;
00445 Window equ    $-StartBnk0+TcsTab
00446      StorVek xmin
00447      sixd    xmax
00448      siyd    ymax
00449      jmp     return
00450 ;
00451 ; Setzen der Schreibrichtung
00452 ;
00453 SetChar equ    $-StartBnk0+TcsTab
00454      StorVek CVek2

```

```

00455      sixd      CVek3
00456      siyd      CVek3+2
00457      lbcd      cvek3      ; Berechnung cvek1=-7(cvek2+cvek3)
00458      call      Add7
00459      sded      CVek1      ; X-Koordinate
00460      lded      CVek2+2
00461      lbcd      CVek3+2
00462      call      Add7
00463      sded      CVek1+2 ; Y-Koordinate
00464      jmp       return
00465
00466 Add7      equ      $-StartBnk0+TcsTab
00467      mov        a,e      ; nur 4-Bit Rechnung!
00468      add        c
00469      mov        e,a
00470      neg
00471      slar       a
00472      slar       a
00473      slar       a      ; *(-8)
00474      add        e      ; *(-7)
00475      mov        e,a
00476      mvi        d,0
00477      rp
00478      mvi        d,0ffh ; Erweiterung 16-bit negativ
00479      ret
00480 ;
00481 ; Ueberpruefung, ob Punkt gesetzt
00482 ;
00483 ChkPnt      equ      $-StartBnk0+TcsTab
00484      mvi        a,1      ; Setzen Error-Flag
00485      sta        errflg
00486      mvi        a,030h ; Maske fuer BIT A
00487      lhld       ypos      ; benoetigt fuer Rechnung
00488      push       h
00489      sbcd       ypos
00490      call      pnt1
00491      lxi        d,01h
00492      jrnz       pntset
00493      lxi        d,0
00494 PntSet      lda        errflg ; Bereichsueberpruefung erfolgreich?
00495      cpi        0
00496      jrz        Inside
00497      lxi        d,-1
00498 Inside      pop        h      ; Wiederherstellung YPos
00499      shld       ypos
00500      jmp       return
00501 ;
00502 ; Lesen bzw. Beschreiben des Bildschirmspeichers
00503 ;
00504 PeekVdc      equ      $-StartBnk0+TcsTab
00505      xchg
00506      call      VdcRam
00507      inp        e
00508      mvi        d,0      ; Ergaenzung auf 16-Bit
00509      jmp       return
00510
00511 PokeVdc      equ      $-StartBnk0+TcsTab
00512      mov        a,c
00513      push       psw      ; Neuer Wert
00514      xchg
00515      call      VdcRam
00516      pop        psw
00517      outp       a
00518      jmp       return
00519 ;
00520 ; Ausgabe Turtle
00521 ;
00522 where      equ      $-StartBnk0+TcsTab
00523      loadVek    xpos
00524      jmp       return
00525 ;
00526 ; Ausgabe Bildschirmfenster
00527 ;
00528 GetWin      equ      $-StartBnk0+TcsTab
00529      loadVek    xmin
00530      lixd       xmax
00531      liyd       ymax
00532      jmp       return
00533 ;
00534 ; Bild loeschen
00535 ;
00536 Erase      equ      $-StartBnk0+TcsTab
00537      call      eras
00538      jmp       return
00539 ;
00540 ; Verschieben (relativ) des Gin-Cursors
00541 ;

```

```

00542 GinMrl equ    $-StartBnk0+tcsTab
00543         lhld   GinX
00544         dad     d
00545         shld    GinX
00546         lhld   GinY
00547         dad     b
00548         shld    GinY          ; jetzt: Zeichnen
00549 ;
00550 ; Zeichnen des GIN-Cursors
00551 ;
00552 GinCrs equ    $-StartBnk0+TcsTab
00553         LoadVek xpos
00554         StorVek tmp
00555         LoadVek GinX
00556         rept    (GinW-1)/2      ; Zeichnen horizontale Linie
00557             dcx    d
00558         endm
00559         StorVek xpos
00560         mvi     b,GinW-1          ; Zaehler
00561 GinCx   push    b
00562         call    xorpnt
00563         lhld   xpos
00564         inx     h
00565         shld   xpos
00566         pop     b
00567         djnz   GinCx
00568         Loadvek GinX          ; Zeichnen vertikale Linie
00569         rept    (GinH-1)/2
00570             dcx    b
00571         endm
00572         StorVek xpos
00573         mvi     b,GinH-1
00574 GinCy   push    b
00575         call    xorpnt
00576         lhld   ypos
00577         inx     h
00578         shld   ypos
00579         pop     b
00580         djnz   GinCy
00581         Loadvek tmp          ; Wiederherstellung Turtle
00582         Storvek xpos
00583         LoadVek GinX
00584         jmp     return
00585 ;
00586 ; i2c Bus Handling
00587 ;
00588 i2cBus equ    $-StartBnk0+TcsTab
00589         if i2cflg eq 1
00590
00591         ; *****
00592         ; * Bereich fuer i2c-Bustreiber *
00593         ; *****
00594         mvi     d,03          ; Dummyroutine
00595         jmp     return
00596
00597         else
00598             mvi     d,03          ; i2cflg=0
00599             jmp     return          ; Dummyroutine
00600         endif
00601 ;
00602 ; Identifikation der Programmversion
00603 ;
00604 version equ    $-StartBnk0+TcsTab
00605         mvi     d,vers*16+release
00606
00607         if (Biosvector or ZModChr or ZMovVDC or ZFilVDC) eq 0
00608         option set    1b
00609         else
00610         option set    0b
00611         endif
00612         if i2cflg eq 1
00613         option set    option or 10b
00614         endif
00615         if VDC64k eq 1
00616         option set    option or 100b
00617         endif
00618         mvi     e,option
00619
00620         jmp     return
00621 ;
00622 ; Blockverschiebung VDC-RAM
00623 ;
00624 VDCmove equ    $-StartBnk0+TcsTab
00625         pushix
00626         pop     h
00627         mov     a,c
00628         call    SMovVDC

```

```

00629      jmp      return
00630 ;
00631 ; Fuellem VDC-RAM
00632 ;
00633 VDCfill equ      $-StartBnk0+TcsTab
00634      pushix
00635      pop       h
00636      mov       a,c
00637      call      SFilVDC
00638      jmp      return
00639 ;
00640 ; Lesen VDC-Register
00641 ;
00642 VDCRegR equ      $-StartBnk0+TcsTab
00643      mov       a,d
00644      call      SelReg
00645      inp       e
00646      mvi       d,0h
00647      jmp      return
00648 ;
00649 ; Schreiben VDC-Register
00650 ;
00651 VDCRegW equ      $-StartBnk0+TcsTab
00652      mov       a,d
00653      call      SelReg
00654      outp      e
00655      jmp      return
00656 ;
00657 ; Initialisierung VDC
00658 ;
00659 VDCinit equ      $-StartBnk0+TcsTab
00660
00661      if        VDC64k eq 1
00662      mvi       a,28d
00663      call      SelReg
00664      inp       e
00665      setb      4,e
00666      mvi       a,28d
00667      call      SelReg
00668      outp      e
00669      call65    dlchr1
00670      call      ModChar
00671      endif
00672
00673      rcall     print
00674      db        0ffh,0          ; CRT loeschen
00675      jmp      return
00676

```

3.26 Tcsbasi2.asm File Reference

CP/M Port: Basic Graphic Routines Bank 0 - subprograms.

3.26.1 Detailed Description

CP/M Port: Basic Graphic Routines Bank 0 - subprograms.

Author

(C) 1995 Dr.-Ing. Klaus Friedewald

Copyright

GNU LESSER GENERAL PUBLIC LICENSE Version 3

Definition in file [Tcsbasi2.asm](#).

3.27 Tcsbasi2.asm

```

00001          page
00002 ;
00003 ; Unterprogramme
00004 ;
00005 ; Initialisierung
00006 ;
00007 init      equ      $-StartBnk0+TcsTab
00008          ImmVek    0,0
00009          StorVek    xmin
00010          ImmVek    639,199
00011          StorVek    xmax
00012          ImmVek    0,-7
00013          StorVek    CVek1
00014          ImmVek    7,0
00015          StorVek    CVek2
00016          ImmVek    -7,1
00017          StorVek    CVek3
00018          ImmVek    319,99
00019          StorVek    GinX
00020          ret
00021 ;
00022 ; Rueckkehr Bank 1
00023 ;
00024 return     equ      $-StartBnk0+TcsTab
00025          mvi        a,ConfB1          ; Bank 1
00026          stc        ; Set Carry=> Return
00027          lspd       OldStack
00028          jmp        RetFar
00029 ;
00030 ; Aufruf der 6502-Subroutine (DE)
00031 ;
00032 enabl65    equ      $-StartBnk0+TcsTab
00033          lhld       cnt65      ; Retten Original-Wiederaufnahmen
00034          shld       tmp
00035          lda        cnt80
00036          lhld       cnt80+1
00037          sta        tmp+2
00038          shld       tmp+3
00039
00040          sded       cnt65      ; Neue Wiederaufnahmen
00041          mvi        a,0c3h      ; Code: JMP
00042          lxi        h,enabl80
00043          sta        cnt80
00044          shld       cnt80+1
00045
00046          jmp        go65
00047
00048 enabl80    equ      $-StartBnk0+TcsTab
00049          ei         ; bei Prozessorumschaltung abgestellt
00050          lhld       tmp      ; Alte Wiederaufnahmen
00051          shld       cnt65
00052          lda        tmp+2
00053          lhld       tmp+3
00054          sta        cnt80
00055          shld       cnt80+1
00056          ret
00057 ;
00058 ; Loeschen des Bildschirmes
00059 ;
00060 eras       equ      $-StartBnk0+TcsTab
00061          FlVDC      0,4000h,0    ; Bildschirm komplett loeschen
00062          lxi        h,0          ; Setzen Turtle
00063          shld       xpos
00064          shld       ypos
00065          ret
00066 ;
00067 ; Unterprogramm Loeschen/Setzen eines Punktes
00068 ;
00069 cpnt       equ      $-StartBnk0+TcsTab
00070          mvi        a,028h      ; Maske fuer RES A
00071          jr         pnt
00072
00073 spnt       equ      $-StartBnk0+TcsTab
00074          mvi        a,020h      ; Maske fuer SET A
00075
00076 pnt        StorVek    xpos      ; Aktualisierung Turtle
00077 ; Bereichsueberpruefung
00078 pnt1       equ      $-StartBnk0+TcsTab
00079          lhld       xmax      ; x>xmax?
00080          ana        a
00081          dsbc       de
00082          rm
00083
00084          lhld       xmin      ; x<xmin?
00085          dcx        h          ; x=xmin ok

```

```

00086      ana      a
00087      dsbc      de
00088      rp
00089
00090      lhld      ymax      ; y>ymax?
00091      ana      a
00092      dsbc      bc
00093      rm
00094
00095      lhld      ymin      ; y<ymin?
00096      dcx      h          ; y=ymin ok
00097      ana      a
00098      dsbc      bc
00099      rp
00100
00101      ana      a          ; Clear Carry
00102      rarr      d          ; x/2
00103      rarr      e
00104      rar      ; auch Clear Carry!
00105      rarr      d; x/4
00106      rarr      e
00107      rar
00108      rarr      d          ; x/8
00109      rarr      e
00110      rar
00111      rrc      ; Einschub Bit SET bzw. RES
00112      rrc      ; 2. Befehlsbyte SET/RES x,A
00113      neg      ; Bit 7 steht links!
00114      sta      bitst+1
00115
00116      lhld      ypos      ; Berechnung Adresse
00117      slar      l          ; y*2
00118      ralr      h
00119      slar      l          ; y*4
00120      ralr      h
00121      dad      b          ; y*5
00122      slar      l          ; y*10
00123      ralr      h
00124      slar      l          ; y*20
00125      ralr      h
00126      slar      l          ; y*40
00127      ralr      h
00128      slar      l          ; y*80
00129      ralr      h
00130      dad      d          ; HL: VDC-Adresse
00131
00132      call      VDCRam
00133      inp      a          ; Alter Wert
00134 bitst equ      $-StartBnk0+TcsTab
00135      setb      0,a      ; wird veraendert!
00136      push      psw      ; A wird von Selreg benutzt
00137      mvi      a,0
00138      sta      errflg    ; keine Bereichsueberschreitung
00139      call      VDCRam
00140      pop      psw      ; Update
00141      outp      a
00142      ret
00143 ;
00144 ; Unterprogramm Ziehen einer Linie
00145 ;
00146 Lin      equ      $-StartBnk0+TcsTab
00147      StorVek xend      ; Merken Endpunkt
00148      lhld      xpos      ; X links/rechts?
00149      dsub      d
00150      jm      rechts
00151      lxix      -1
00152      jr      links
00153 rechts equ      $-StartBnk0+TcsTab
00154      lhld      xend      ; Berechnung Betrag
00155      ldcd      xpos
00156      dsub      d
00157      lxix      1
00158 links  xchg      ; relatives Label!
00159
00160      lhld      ypos      ; Y rauf/runter?
00161      dsub      b
00162      jm      runter
00163      lxix      -1
00164      jr      rauf
00165 runter equ      $-StartBnk0+TcsTab
00166      lxix      1
00167      lhld      yend
00168      lbcd      ypos
00169      dsub      b
00170 rauf   mov      b,h      ; relatives Label!
00171      mov      c,l
00172

```

```

00173      sixd      DeltaX  ; Merken Richtungen
00174      siyd      Deltay
00175
00176      dsub      d        ; Berechnung Steigung
00177      push      psw      ; Rette Flag
00178      jm        DxGross
00179      mov       h,b
00180      mov       l,c
00181      lxi       b,-1
00182      jrz      DxGldy
00183      inx       d        ; Nur wenn dx<>dy
00184 DxGldy:
00185 LpY          equ      $-StartBnk0+TcsTab
00186      inx       b
00187      dsub      d
00188      jrnz     NoNullY
00189      inx       b        ; keine Schrittaenderung
00190      jr        NullY
00191 NoNullY     jp        LpY      ; relatives Label!
00192 NullY       lxi       d,1      ; relatives Label!
00193      shld     Nxklein ; Teilungsrest
00194      lxi       h,0
00195      shld     Nyklein
00196      jr        DyGross
00197
00198 DxGross     equ      $-StartBnk0+TcsTab
00199      xchg
00200      inx       b
00201      lxi       d,-1
00202 LpX          equ      $-StartBnk0+TcsTab
00203      inx       d
00204      dsub      b
00205      jrnz     NoNullX
00206      inx       d        ; keine Schrittaenderung
00207      jr        NullX
00208 NoNullX     jp        LpX      ; relatives Label!
00209 NullX       lxi       b,1      ; relatives Label!
00210      shld     Nyklein ; Teilungsrest
00211      lxi       h,0
00212      shld     NxKlein
00213
00214 DyGross     sbcd      ndx      ; relatives Label!
00215      sded      ndy      ; Merken Steigung
00216      sbcd      CountX  ; Initialisieren Zeichnen
00217      sded      CountY
00218
00219 LinLp       equ      $-StartBnk0+TcsTab
00220 LoadVek     xpos      ; relatives Label!
00221      lhld     xend
00222      dsub      d
00223      jrnz     NoXend
00224      pop      psw      ; Flag x/y-Lauf
00225      jm        finish
00226      push     psw
00227 NoXend      lhld     yend      ; relatives Label!
00228      dsub      b
00229      jrnz     NoYend
00230      pop      psw
00231      jp        Finish
00232      push     psw
00233 NoYend      lda      maske    ; relatives Label!
00234      rlc
00235      sta      maske
00236      jrnz     Loch
00237      call     spnt
00238      jr        loch1
00239 loch        call     cpnt      ; relatives Label!
00240 loch1       lxi       d,1      ; relatives Label!
00241      lhld     countx
00242      dsub      d
00243      jrnz     xgut
00244      lbcd     xpos
00245      lhld     DeltaX
00246      dad      b
00247      shld     xpos
00248      lhld     Nxklein
00249      ana      a
00250      dadc     d
00251      shld     Nxklein
00252      jrnz     CxOk
00253      lhld     ndx
00254      inx       h
00255      shld     ndx
00256 CxOk        lhld     ndx      ; relatives Label!
00257
00258 xgut        shld     CountX  ; relatives Label!
00259      lhld     county

```



```

00260      dsub      d
00261      jrnz      Ygut
00262      lbcd      ypos
00263      lhld      DeltaY
00264      dad       b
00265      shld      ypos
00266      lhld      Nyklein
00267      ana       a
00268      dadc      d
00269      shld      Nyklein
00270      jrnz      CyOk
00271      lhld      ndy
00272      inx       h
00273      shld      ndy
00274 CyOk      lhld      ndy      ; relatives Label!
00275 Ygut      shld      county    ; relatives Label!
00276      jmp       LinLp
00277
00278 Finish     equ      $-StartBnk0+TcsTab
00279      LoadVek  xend      ; Setzen des letzten Punktes
00280      lda       maske
00281      rlc
00282      jrnz      Loch2
00283      call      spnt
00284      jr        loch3
00285 loch2      call      cpnt      ; relatives Label!
00286 loch3      ret           ; relatives Label!
00287 ;
00288 ; Unterprogramm MoveRel
00289 ;
00290 MovR1      equ      $-StartBnk0+TcsTab
00291      lhld      xpos
00292      dad       d
00293      shld      xpos
00294      lhld      ypos
00295      dad       b
00296      shld      ypos
00297      ret
00298 ;
00299 ; Unterprogramm DrwRel
00300 ;
00301 DrwR1      equ      $-StartBnk0+TcsTab
00302      lhld      xpos      ; Berechnung xend
00303      dad       d
00304      xchg
00305      lhld      ypos      ; Berechnung yend
00306      dad       b
00307      mov       b,h
00308      mov       c,l
00309      call      lin
00310      ret
00311 ;
00312 ; Unterprogramm Invertieren Punkt
00313 ;
00314 xorpn1     equ      $-StartBnk0+TcsTab
00315      loadvek  xpos
00316      mvi      a,1
00317      sta      errflg      ; Vorbereitung Bereichsueberpruefung
00318      mvi      a,30h      ; Maske BIT A
00319      call      pnt1
00320      push     psw
00321      lda      bitst+1
00322      jrz      xorpn1      ; Punkt ist nicht gesetzt
00323      ani      1011$1111b  ; RESET-Befehl=SET ohne bit 6
00324 xorpn1    ori      1000$0000b ; SET-Befehl
00325      sta      xorpn3+1
00326      lda      errflg
00327      cpi      0
00328      jrz      xorpn2      ; Im Fenster
00329      pop      psw
00330      ret
00331 xorpn2     pop      psw      ; alter Wert
00332 xorpn3     equ      $-StartBnk0+TcsTab
00333      setb     0,a          ; wird veraendert
00334      push     psw
00335      call      vdcram      ; HL: noch VIC-Adresse
00336      pop      psw
00337      outp     a
00338      ret
00339 ;
00340 ; Ersatz fuer Z80-ROM Routine: Anpassung Zeichensatz
00341 ;
00342      IF ZModChr ne 0
00343
00344 ModChar     equ      ZModChr      ; Anspringen Z80-ROM
00345
00346      ELSE

```

```

00347
00348 ModChar equ    $-StartBnk0+TcsTab
00349 CopVDC macro  source,dest,count
00350             lxi    h,source
00351             lxi    d,dest
00352             lxi    b,count
00353             push   h
00354             rcall  7Ch
00355             endm
00356 Cp8VDC macro  Dest,Count      ; Kopieren Buchstaben
00357             lxi    h,Dest      ; DE: Source
00358             mvi    b,Count
00359             push   h
00360             rcall  7Eh
00361             endm
00362
00363             FlVDC   3800h,0400h,0
00364             CopVDC 37A0h,38A0h,0008h
00365             CopVDC 3690h,3890h,0008h
00366             CopVDC 35e0h,38e0h,0018h
00367             CopVDC 3010h,3610h,0198h
00368             FlVDC 3000h,0200h,0
00369             CopVDC 2000h,3400h,0008h
00370             CopVDC 21b0h,35b0h,0028h
00371             CopVDC 21c0h,3800h,0008h
00372             CopVDC 21e0h,3810h,0018h
00373             CopVDC 2400h,3c00h,03f8h
00374             lxi    d,0f1ah
00375             Cp8VDC 35C0h,1
00376             Cp8VDC 35e0h,3
00377             Cp8VDC 37b0h,5
00378             ret
00379
00380             ENDIF
00381 ;
00382 ; Ersatz Move VDC-RAM
00383 ;
00384             IF ZMovVDC ne 0
00385
00386 SMovVDC equ    ZMovVDC
00387
00388             ELSE
00389
00390 SMovVDC equ    $-StartBnk0+TcsTab
00391             push   PSW
00392             call   VDCRam
00393             mvi    a,24
00394             call   SelReg
00395             inp    a
00396             ori    80h
00397             outp   a
00398             mvi    a,32
00399             call   selreg
00400             outp   d
00401             mvi    a,33
00402             call   SelReg
00403             outp   e
00404             mvi    a,30
00405             call   SelReg
00406             pop    PSW
00407             outp   a
00408             ret
00409
00410             ENDIF
00411 ;
00412 ; Ersatz Fill VDC-RAM
00413 ;
00414             IF ZFilVDC ne 0
00415
00416 SFilVDC equ    ZFilVDC
00417
00418             ELSE
00419
00420 SFilVDC equ    $-StartBnk0+TcsTab
00421             push   PSW
00422             call   VDCRam
00423             outp   d
00424             pop    PSW
00425             dcr    a
00426             rz
00427             push   PSW
00428             mvi    a,24
00429             call   SelReg
00430             inp    a
00431             ani    7fh
00432             outp   a
00433             mvi    a,30

```

```

00434      call  SelReg
00435      pop   PSW
00436      outp  a
00437      mvi   b,0
00438      mov   c,a
00439      inx   b
00440      dad   b
00441      push  d
00442      push  h
00443      mvi   a,12h
00444      call  SelReg
00445      inp   h
00446      mvi   a,13h
00447      call  selreg
00448      inp   l
00449      pop   d
00450      pop   b
00451 Floc1  mov   a,h
00452      cmp   d
00453      jrnz  Floc2
00454      mov   a,l
00455      cmp   e
00456 Floc2  rnc
00457      push  b
00458      call  VDCRam
00459      pop   b
00460      outp  b
00461      inx   h
00462      jr    Floc1
00463
00464      ENDIF
00465
00466 EndBnk0:
00467 ;
00468 ; Datenbereich in Bank 0 (wird nicht kopiert)
00469 ;
00470 OldStack equ    $-StartBnk0+TcsTab
00471 xpos     equ    OldStack+2
00472 ypos     equ    xpos+2
00473 xmin     equ    ypos+2
00474 ymin     equ    xmin+2
00475 xmax     equ    ymin+2
00476 ymax     equ    xmax+2
00477 CVek1    equ    ymax+2
00478 CVek2    equ    CVek1+4
00479 CVek3    equ    CVek2+4
00480 GinX     equ    CVek3+4
00481 GinY     equ    GinX+2
00482 errflg   equ    GinY+2
00483 NewStack equ    errflg+101h    ; Stack fuer Z80
00484
00485 tmp      equ    NewStack+1      ; temporaerer Arbeitsspeicher
00486
00487 Maske    equ    tmp            ; temporaer fuer Line
00488 Xend     equ    tmp+1
00489 YEnd     equ    tmp+3
00490 Deltax   equ    tmp+5
00491 DeltaY   equ    tmp+7
00492 Ndx     equ    tmp+9
00493 NDy     equ    tmp+11
00494 CountX   equ    tmp+13
00495 CountY   equ    tmp+15
00496 Nxxlein equ    tmp+17
00497 Nyklein equ    tmp+19
00498
00499 ChrMsk   equ    tmp+21 ; temporaer fuer CharOut
00500 ChrPtr   equ    tmp+29
00501
00502 ZahlAus  equ    tmp+5 ; temporaer fuer i2cBus
00503 Zahlein  equ    tmp+6 ; bis tmp+4 fuer call65
00504 FehlNr   equ    tmp+7
00505 PuffAus  equ    tmp+8 ; 12 Byte Ausgabepuffer
00506 PuffEin  equ    tmp+20 ; Eingabepuffer
00507 Byte     equ    tmp+32 ; Arbeitsspeicher
00508 BusTim   equ    tmp+33
00509 PegTim   equ    tmp+34
00510
00511 EndTmp   equ    tmp+35
00512
00513 ZerPg65  equ    (high EndTmp)+1 ; temporaerer 6502-Stack
00514      end

```

3.28 TCSdrCPM.for File Reference

CP/M Port: Basic graphic subroutines for C128 under CP/M.

Functions/Subroutines

- subroutine [tcslev](#) (LEVEL)
- subroutine [svstat](#) (Array)
- subroutine [restat](#) (Array)
- subroutine [dcursr](#) (IC, IX, IY)
- subroutine [csize](#) (Ixlén, iylen)
- subroutine [initt](#)
- subroutine [seeloc](#) (IX, IY)

3.28.1 Detailed Description

CP/M Port: Basic graphic subroutines for C128 under CP/M.

Version

(2002,312,1)

Author

(C) 1995 Dr.-Ing. Klaus Friedewald

Copyright

GNU LESSER GENERAL PUBLIC LICENSE Version 3

Definition in file [TCSdrCPM.for](#).

3.28.2 Function/Subroutine Documentation

3.28.2.1 [csize\(\)](#)

```
subroutine csize (  
    Ixlén,  
    iylen )
```

Definition at line [82](#) of file [TCSdrCPM.for](#).

3.28.2.2 dcursr()

```
subroutine dcursr (
    IC,
    IX,
    IY )
```

Definition at line 71 of file [TCSdrCPM.for](#).

3.28.2.3 initt()

```
subroutine initt
```

Definition at line 92 of file [TCSdrCPM.for](#).

3.28.2.4 restat()

```
subroutine restat (
    real, dimension(1) Array )
```

Definition at line 58 of file [TCSdrCPM.for](#).

3.28.2.5 seeloc()

```
subroutine seeloc (
    IX,
    IY )
```

Definition at line 106 of file [TCSdrCPM.for](#).

3.28.2.6 svstat()

```
subroutine svstat (
    real, dimension(1) Array )
```

Definition at line 45 of file [TCSdrCPM.for](#).

3.28.2.7 tcslev()

```
subroutine tcslev (
    dimension(3) LEVEL )
```

Definition at line 27 of file [TCSdrCPM.for](#).

3.29 TCSdrCPM.for

```
00001 C
00002 C   TCS Graphik Grundfunktionen für C128 unter CP/M
00003 C
00004 C   Version (2002,312,1)
00005 C
00006 C   Vereinheitlichung DOS/Windowsversion
00007 C   Zusätzliches Modul: TCSdrCPM.FOR, früher Teil von TCS.FOR.
00008 C   Muss vor TCS.FOR in die Library aufgenommen werden, da sonst
00009 C   der 1-pass Linker von CP/M rscale,setmrg,rrotat nicht findet.
00010 C   Implementierung Unterprogramm TCSLEV
00011 C   Umbenennung TKTRNX.FOR in 'TKTRNX.FD'
00012 C
00013 C
00014 C   15.10.02      Dr.-Ing. K. Friedewald
00015 C
00016 C   Bugfix SVSTAT, RESTAT: Beginn Kommentarzeilen "C" statt "c"
00017 C   Länge Commonblock korrigiert
00018 C
00019 C   08.11.02      Dr.-Ing. K. Friedewald
00020 C
00021 C
00022 C
00023 C
00024 C
00025 C   Ausgabe der Softwareversion
00026 C
00027 C   subroutine tcslev(LEVEL)
00028 C   dimension level(3)
00029 C   Aenderungsjahr
00030 C   level(1)=2002
00031 C   Aenderungstag
00032 C   level(2)= 312
00033 C   System= CP/M
00034 C   level(3)= 1
00035 C
00036 C   return
00037 C   end
00038 C
00039 C
00040 C
00041 C
00042 C   Abspeichern Terminal Status Area
00043 C
00044 C
00045 C   subroutine svstat (Array)
00046 C   real array(1)
00047 C   common /tktrnx/ arr(14)
00048 C   do 10 i=1,14
00049 10   array(i)= arr(i)
00050 C   call seeloc (i,j)
00051 C   array(15)= float(i)
00052 C   array(16)= float(j)
00053 C   return
00054 C   end
00055 C
00056 C
00057 C
00058 C   subroutine restat (Array)
00059 C   real array(1)
00060 C   common /tktrnx/ arr(14)
00061 C   do 10 i=1,14
00062 10   arr(i)= array(i)
00063 C   call movabs (ifix(array(15)), ifix(array(16)))
00064 C   return
00065 C   end
00066 C
00067 C
00068 C   Graphic Input
00069 C
00070 C
00071 C   subroutine dcursr (IC,IX,IY)
```

```

00072      call gincrs (ic,ix,iy)
00073      ix= ix*8/5
00074      iy= (199-iy)*4
00075      return
00076      end
00077
00078 C
00079 C Alphanumerische Ausgabe
00080 C
00081
00082      subroutine csize (Ixlen,iylen)
00083      ixlen= 11
00084      iylen= 32
00085      return
00086      end
00087
00088 C
00089 C Initialisierungsroutinen
00090 C
00091
00092      subroutine initt
00093      call lintrn
00094      call swindo (0,1023,0,780)
00095      call vwindo (0.,1023.,0.,780.)
00096      call rrotat (0.)
00097      call rscale (1.)
00098      call setmrg (0,1023)
00099      call initt1
00100      call home
00101      return
00102      end
00103
00104
00105
00106      subroutine seeloc (IX,IY)
00107      call seelc1 (ix,iy)
00108      ix= ix*8/5
00109      iy= (199-iy)*4
00110      return
00111      end
00112

```

3.30 Tcsdrivr.asm File Reference

CP/M Port: Direct graphic.

3.30.1 Detailed Description

CP/M Port: Direct graphic.

Version

1.1

Author

(C) 1995 Dr.-Ing. Klaus Friedewald

Copyright

GNU LESSER GENERAL PUBLIC LICENSE Version 3

Definition in file [Tcsdrivr.asm](#).

3.31 Tcsdrivr.asm

```

00001      title 'TCS: direkte Graphik'
00002  $*Macro
00003  ; Tektronix Plot 10 Emulation fuer C128
00004  ;
00005  ;      Grundfunktionen des TCS
00006  ;          20.4.88      Dr.-Ing. K. Friedewald
00007  ;
00008  ;      Version 1.1
00009  ;      Ergaenzung Dummy-Routinen der Windowsversion
00010  ;          bckcol, lincol, txtcol, defaultcolour
00011  ;          dblsiz, nrmsiz, winlbl
00012  ;
00013  ;      Einfuehrung WIN16-Linestyles in DSHABS und DSHREL
00014  ;
00015  ;          20.10.2002      K. Friedewald
00016  ;
00017  maclib  z80
00018
00019  public  inittl, seelcl, swindl,statst,alpha
00020
00021  public  tinput, bell, finitt, erase
00022  public  movabs, pntabs, drwabs, dshabs
00023  public  movrel, pntrel, drwrel, dshrel
00024  public  toutpt, toutst
00025  public  anmode, alfmod, pclipt, iowait
00026  public  bckcol, lincol, txtcol, defaultcolour
00027  public  dblsiz, nrmsiz, winlbl
00028
00029  name    'TCSDRIVR'
00030
00031  tcstab equ 4000h
00032  jmpfar equ 0fff1h
00033  bdos equ 5
00034
00035  inittl lxi h,tcstab
00036  jmp jmpfar
00037
00038  tinput push h ; Adresse Parameter
00039  mvi c,6 ; Direct Console I/O
00040  mvi e,0FDh ; Wait for Char
00041  call bdos
00042  pop h
00043  mov m,a
00044  inx h ; Ergaenzen auf 16bit
00045  mvi a,0
00046  mov m,a
00047  DumRet ret ; auch: Ruecksprung Dummy-Routinen
00048
00049  anmode equ DumRet ; Dummy-Routinen
00050  iowait equ DumRet
00051  alfmod equ DumRet
00052  pclipt equ DumRet
00053  bckcol equ DumRet
00054  lincol equ DumRet
00055  txtcol equ DumRet
00056  defaultcolour equ DumRet
00057  dblsiz equ DumRet
00058  nrmsiz equ DumRet
00059  winlbl equ DumRet
00060
00061  bell mvi c,2 ; Console Out
00062  mvi e,7 ; Bell
00063  jmp bdos
00064
00065  statst push h ; retten Parameter
00066  push d
00067  lxi d,0
00068  lxi b,5
00069  lxi h,tcstab+12h ; MovAbs
00070  call jmpfar
00071  lxi d,0
00072  lxi b,0
00073  lxix 639
00074  lxiy 4
00075  lxi h,tcstab+1eh ; Setzen Window
00076  call jmpfar
00077  pop d
00078  pop h
00079  call toutst
00080  lxi d,0 ; altes Window
00081  lxi b,0
00082  lxix 639
00083  lxiy 199
00084  lxi h,tcstab+1eh
00085  jmp jmpfar ; und zurueck

```



```

00086
00087 finitt lxi h,fimsgl ; 2 Dummy-Parameter
00088 lxi d,fimssg
00089 call statst ; Ausgabe Meldung
00090 call bell
00091 mvi c,6 ; Warten auf beliebigen Tastendruck
00092 mvi e,0FDh
00093 call bdos
00094 alpha lxi h,tcstab+3 ; alpha
00095 jmp jmpfar
00096
00097 dseg
00098 fimssg db 'press any key to continue'
00099 fimsgl dw $-fimssg
00100 cseg
00101
00102 erase lxi h,tcstab+33h
00103 jmp jmpfar
00104
00105 toutpt mov e,m ; Parameter: ASCII
00106 lxi h,tcstab+1bh ; CharOut
00107 jmp jmpfar
00108
00109 toutst mov b,m ; N Char
00110 xchg
00111 toutlp mov e,m
00112 push h ; Adr. String
00113 push b
00114 lxi h,tcstab+1bh ; CharOut
00115 call jmpfar
00116 pop b
00117 pop h
00118 inx h ; Naechster Buchstabe
00119 djnz toutlp
00120 ret
00121
00122 pntabs call getxy
00123 lxi h,tcstab+09h ; SetPnt
00124 jmp jmpfar
00125
00126 drwabs lxix 0FFh ; Durchgezogene Linie
00127 Line call getxy
00128 lxi h,tcstab+0fh ; Line
00129 jmp jmpfar
00130
00131 dshabs sbcd dshab1+2 ; Adresse der LinienMaske
00132 inx b ; Pointer auf High-Byte
00133 ldax b ; High-Byte in [A]
00134 dcx b ; Restore Pointer
00135 cpi 00h
00136 jnz dshab1 ; HiByte>0 -> kein WIN16 Linienstil
00137 ldax b ; LowByte der Maske
00138 cpi 00h ; Durchgezogene Linie
00139 jnz dshab2 ; Pruefe weiter falls nicht
00140 lxix 0ffffh
00141 jr line
00142 dshab2 cpi 01h ; dotted?
00143 jnz dshab3 ; nein
00144 lxix 0aaaaah
00145 jr line
00146 dshab3 cpi 02h ; dash-dotted?
00147 jnz dshab4 ; nein
00148 lxix 0e4e4h
00149 jr line
00150 dshab4 cpi 03h ; dashed?
00151 jnz dshab1 ; nein-> benutze Parameter als Maske
00152 lxix 0f0fh
00153 jr line
00154 dshab1 lixd 1234 ; Parameter wird eingetragen
00155 jr line ; Rest wie DRWABS
00156
00157 movabs call getxy
00158 lxi h,tcstab+12h
00159 jmp jmpfar
00160
00161 pntrel call movrel
00162 lxi d,0 ; DrwRel mit 0,0
00163 lxi b,0
00164 lxix 0FFh
00165 lxi h,tcsTab+18h
00166 jmp jmpfar
00167
00168 drwrel lxix 0FFh
00169 LinRel call getrel
00170 lxi h,tcstab+18h
00171 jmp jmpfar
00172

```

```

00173 dshrel sbcd      dshrel+2
00174      inx         b           ; Pointer auf High-Byte
00175      ldax         b           ; High-Byte in [A]
00176      dcx         b           ; Restore Pointer
00177      cpi          00h
00178      jnz         dshrel       ; HiByte>0 -> kein WIN16 Linienstil
00179      ldax         b           ; LowByte der Maske
00180      cpi          00h         ; Durchgezogene Linie
00181      jnz         dshre2       ; Pruefe weiter falls nicht
00182      lxix         0ffffh
00183      jr          LinRel
00184 dshre2 cpi          01h       ; dotted?
00185      jnz         dshre3       ; nein
00186      lxix         0aaaaah
00187      jr          LinRel
00188 dshre3 cpi          02h       ; dash-dotted?
00189      jnz         dshre4       ; nein
00190      lxix         0e4e4h
00191      jr          LinRel
00192 dshre4 cpi          03h       ; dashed?
00193      jnz         dshrel       ; nein-> benutze Parameter als Maske
00194      lxix         0f0fh
00195      jr          LinRel
00196 dshrel lixd        1234       ; wird veraendert!
00197      jr          LinRel       ; Rest wie DRWREL
00198
00199 movrel call      getrel
00200      lxi          h,tcstab+15h
00201      jmp         jmpfar
00202
00203 swindl push       b           ; Rette Parameterblock
00204      call        getxy
00205      pop         h
00206      push        d           ; xmin
00207      push        b           ; ymax
00208      mov         e,m
00209      inx         h
00210      mov         d,m         ; DE: x2
00211      push        d
00212      inx         h
00213      mov         e,m
00214      inx         h
00215      mov         d,m         ; DE: y2
00216      pop         h
00217      call        getxy       ; ymin ok
00218      push        d           ; xmax
00219      popix
00220      popiy       ; ymax
00221      pop         d           ; xmin
00222      lxi          h,tcstab+1Eh
00223      jmp         jmpfar
00224
00225 seelcl push        h           ; rette Parameter
00226      push        d
00227      lxi          h,tcstab+2Dh ; Where
00228      call        jmpfar
00229      pop         h           ; 2. Parameter
00230      mov         m,c         ; Yloc
00231      inx         h
00232      mov         m,b
00233      pop         h
00234      mov         m,e
00235      inx         h
00236      mov         m,d
00237      ret
00238
00239 ;
00240 ; Hole x-y Koordinaten
00241 ;
00242 getxy push        d           ; Rette 2 Parameter
00243      mov         e,m
00244      inx         h
00245      mov         d,m         ; X-Koordinate
00246      pop         h
00247      mov         c,m
00248      inx         h
00249      mov         b,m         ; Y-Koordinate
00250 ;
00251 ; Umrechnung Koordinatensystem (1024/780) auf (640/195)
00252 ;
00253 xytran inx         d           ; runden der Koordinaten
00254      inx         b
00255      inx         b
00256
00257      srar         d           ; x=x/2+x/8
00258      rarr         e           ; /2
00259      mov         h,d

```

```
00260      mov     l,e
00261      srar    d
00262      rarr    e           ; /4
00263      srar    d
00264      rarr    e           ; /8
00265      dad     d
00266      xchg    ; X ok
00267
00268      lxi     h,199       ; Y=199-y/4
00269      srar    b
00270      rarr    c           ; /2
00271      srar    b
00272      rarr    c           ; /4
00273      ana     a           ; Clear Carry
00274      dsbc    b
00275      mov     b,h
00276      mov     c,l
00277      ret
00278
00279 ;
00280 ; XY Koordinaten fuer REL-Befehle
00281 ;
00282 getrel call    getxy
00283      lxi     h,-199      ; Hier: Ohne Offset!!
00284      dad     b
00285      mov     b,h
00286      mov     c,l
00287      ret
```

3.32 Tcsgin.asm File Reference

CP/M Port: Graphic input.

3.32.1 Detailed Description

CP/M Port: Graphic input.

Version

1.0

Author

(C) 1995 Dr.-Ing. Klaus Friedewald

Copyright

GNU LESSER GENERAL PUBLIC LICENSE Version 3

Definition in file [Tcsgin.asm](#).

3.33 Tcsgin.asm

```

00001          title 'TCS: Graphic Input'
00002 $*Macro
00003 ; Tektronix Plot 10 Emulation fuer C128
00004 ;
00005 ;      Assemblerteil der Gin-Cursorroutinen
00006 ;
00007 ; Version 1.0
00008 ;          13.5.88          Dr.-Ing. K. Friedewald
00009          maclib z80
00010
00011          public  GinCrs
00012          name    'TCSGIN'
00013
00014 tcstab    equ     04000h
00015 JmpFar    equ     0FFF1h
00016 Cia1     equ     0DC00h
00017
00018 CLinks    equ     1          ; Tastaturcode der Cursortasten
00019 CRecht    equ     6
00020 CHoch     equ     5
00021 CTief     equ     24
00022
00023 bdos      equ     5
00024
00025
00026 GinCrs    push    h          ; Parameter retten
00027            push    d
00028            push    b
00029
00030            lxi     h,tcstab+36h ; Zeichnen Ausgangsposition
00031            call    jmpfar
00032            push    psw         ; Dummy
00033
00034 CrsLp      pop     psw         ; Alter Terminator
00035            lxi     h,tcstab+36h ; Loeschen alte Position
00036            call    jmpfar
00037
00038            mvi     e,0ffh      ; Input/Status
00039            mvi     c,6         ; Direct Console i/o
00040            call    bdos
00041            push    psw         ; Terminator
00042            cpi     0
00043            jr      Joy
00044
00045            lxi     d,0
00046            lxi     b,0
00047            cpi     CLinks
00048            jrnz    NCl
00049            lxi     d,-1
00050            mvi     a,0ffh      ; kein Joystick
00051            jr      MovGin
00052 nCl        cpi     CRecht
00053            jrnz    nCr
00054            lxi     d,1
00055            mvi     a,0ffh      ; kein Joystick
00056            jr      MovGin
00057 nCr        cpi     CHoch
00058            jrnz    nch
00059            lxi     b,-1
00060            mvi     a,0ffh      ; kein Joystick
00061            jr      MovGin
00062 nch        cpi     CTief
00063            jrnz    nct
00064            lxi     b,1
00065            mvi     a,0ffh      ; kein Joystick
00066            jr      MovGin
00067 nct        mvi     a,1110$1111b ; Simulation Feuerknopf
00068            jr      MovGin
00069
00070 Joy        mvi     d,1110$0000b ; Lesen Joystick 2
00071            lxi     b,Cia1+2
00072            outp    d
00073            lxi     b,Cia1
00074            inp     a
00075            mvi     d,0FFh
00076            lxi     b,cia1+2
00077            outp    d
00078
00079            lxi     d,0
00080            lxi     b,0
00081
00082            bit     0,a
00083            jrnz    NJoyO
00084            lxi     b,-2
00085 NJoyO      bit     1,a

```

```
00086      jrnz    NJoyU
00087      lxi      b,2
00088 NJoyU      bit      2,a
00089      jrnz    NJoyL
00090      lxi      d,-4
00091 NJoyL      bit      3,a
00092      jrnz    MovGin
00093      lxi      d,4
00094 MovGin      push    psw          ; Retten Joy-Zustand
00095      lxi      h,tcstab+39h      ; Neue Position GinCrs
00096      call     jmpfar
00097      pop      psw
00098      bit      4,a
00099      jnz      CrsLp
00100
00101      lxi      h,tcstab+36h      ; Loeschen GinCrs
00102      call     jmpfar
00103
00104      pop      psw          ; Terminator
00105      pop      h          ; Parameter iy
00106      mov      m,c
00107      inx      h
00108      mov      m,b
00109      pop      h          ; Parameter ix
00110      mov      m,e
00111      inx      h
00112      mov      m,d
00113      pop      h          ; Parameter Terminator
00114      mov      m,a
00115      inx      h
00116      mvi      m,0
00117      ret
00118      end
00119
```

3.34 Tcshdc.asm File Reference

CP/M Port: Hardcopy.

3.34.1 Detailed Description

CP/M Port: Hardcopy.

Version

1.0

Author

(C) 1995 Dr.-Ing. Klaus Friedewald

Copyright

GNU LESSER GENERAL PUBLIC LICENSE Version 3

Definition in file [Tcshdc.asm](#).

3.35 Tcshdc.asm

```

00001      title    'TCS: Hardcopy'
00002 ; Abspeichern des Graphikbildschirms in Files 'BILDx.HDC'
00003 ;
00004 ; Version 1.0
00005 ;      13.5.88      Dr.-Ing. K. Friedewald
00006 ;
00007 $*Macro
00008
00009      maclib    z80
00010      public   hdcopy
00011      name     'TCSHDC'
00012
00013 tcstab equ     4000h
00014 jmpfar equ     0FFF1h
00015
00016 sfirst equ     17
00017 makef equ     22
00018 setdma equ     26
00019 multio equ     44
00020 writes equ     21
00021 reads equ     20
00022 closef equ     16
00023
00024 filsiz equ     640/8*200/128      ; Anzahl Records
00025
00026 bdos macro     function, parameter
00027 mvi             c,function
00028 lxi             d,parameter
00029 call            5
00030 endm
00031
00032 hdcopy mvi      a,'A'              ; Konstruktion des Filenamens
00033 sta            fcbpar
00034 mvi            a,0
00035 sta            fcbex
00036 sta            fcbcr
00037 sealp bdos     sfirst, fcb
00038 cpi            255
00039 jrz            notfnd
00040 lda            fcbpar
00041 inr            a                  ; Schon vorhanden
00042 sta            fcbpar              ; also naechster Versuch
00043 jr             sealp
00044 notfnd bdos     makef,fcb
00045 bdos           setdma,iobuf
00046 bdos           multio,1
00047
00048 lxi            d,0-1              ; Bildschirmadresse-1
00049 mvi            b,Filsiz           ; Anzahl Records
00050
00051 Getlp push      b
00052 mvi            b,128              ; Laenge log. Record
00053 lxi            h,iobuf
00054 shld          bufptr
00055 getlp1 inx      d                  ; naechste VDC-Adresse
00056 push          d
00057 push          b                    ; restliche Blocklaenge
00058 lxi            h,tcstab+27h       ; Peek VDC
00059 call          jmpfar
00060 lhld          bufptr
00061 mov           m,e
00062 inx           h
00063 shld          bufptr
00064 pop           b
00065 pop           d
00066 djnz          getlp1
00067 push          d
00068 bdos           writes,fcb
00069 pop           d
00070 pop           b
00071 djnz          getlp
00072
00073 bdos           closef,fcb
00074 ret
00075
00076 dseg
00077 fcb db          0                  ; Default Drive
00078 db            'BILD? '            ; Filename
00079 fcbpar equ     $-4
00080 db            'HDC'                ; extension
00081 fcbex db          0
00082 ds            19
00083 fcbcr db          0
00084
00085 iobuf ds        128

```

```
00086 bufptr dw      0
00087         end
00088
```

3.36 Tcsital.asm File Reference

CP/M Port: Italic.

3.36.1 Detailed Description

CP/M Port: Italic.

Version

1.0

Author

(C) 1995 Dr.-Ing. Klaus Friedewald

Copyright

GNU LESSER GENERAL PUBLIC LICENSE Version 3

Definition in file [Tcsital.asm](#).

3.37 Tcsital.asm

```
00001         title   'TCS: Kursivschrift'
00002 ; TCS-Subroutinen ITALIC und ITALIR
00003 ;
00004 ; Version 1.0
00005 ;      14.5.88      Dr.-Ing. K. Friedewald
00006 $*Macro
00007         maclib    z80
00008         public    italic,italir
00009         name      'TCSITL'
00010 jmpfar equ      0FFF1h
00011 TcsTab equ      4000h
00012
00013 italic lxix      -8
00014 itall  lxiy      1
00015         lxi       d,7
00016         lxi       b,0
00017         lxi       h,tcstab+21h      ; SetChar
00018         jmp       jmpfar            ; und fertig
00019
00020 italir lxix      -7
00021         jmp       itall              ; Rest s.o.
00022         end
00023
00024
```


Index

- 'tktrnx.fd', [5](#)
- Alpha.mac, [5](#)
- ancho
 - TCS.for, [18](#)
- anstr
 - TCS.for, [18](#)
- baksp
 - TCS.for, [18](#)
- cartn
 - TCS.for, [18](#)
- csize
 - TCSdrCPM.for, [46](#)
- dasha
 - TCS.for, [18](#)
- dashr
 - TCS.for, [18](#)
- dcursr
 - TCSdrCPM.for, [46](#)
- drawa
 - TCS.for, [19](#)
- drawr
 - TCS.for, [19](#)
- dwindo
 - TCS.for, [19](#)
- genflg
 - TCS.for, [19](#)
- Gethdc.asm, [7](#)
- Graph.asm, [8](#)
- Graph.bib, [9](#)
- Graphjob.sou, [11](#)
- Graphsta.sou, [12](#)
- hdcout
 - Hdcout.for, [13](#)
- Hdcout.for, [12](#)
- hdcout, [13](#)
- Hdcout1.asm, [15](#)
- home
 - TCS.for, [19](#)
- initt
 - TCSdrCPM.for, [47](#)
- linef
 - TCS.for, [20](#)
- linhgt
 - TCS.for, [20](#)
- lintrn
 - TCS.for, [20](#)
- linwdt
 - TCS.for, [20](#)
- logtrn
 - TCS.for, [20](#)
- Mainpage.dox, [15](#)
- movea
 - TCS.for, [20](#)
- mover
 - TCS.for, [21](#)
- newlin
 - TCS.for, [21](#)
- newpag
 - TCS.for, [21](#)
- pointa
 - TCS.for, [21](#)
- pointr
 - TCS.for, [21](#)
- Print.asm, [15](#)
- rel2ab
 - TCS.for, [22](#)
- rescal
 - TCS.for, [22](#)
- restat
 - TCSdrCPM.for, [47](#)
- revcot
 - TCS.for, [22](#)
- rrotat
 - TCS.for, [22](#)
- rscale
 - TCS.for, [22](#)
- seeloc
 - TCSdrCPM.for, [47](#)
- seetrm
 - TCS.for, [23](#)
- seetrn
 - TCS.for, [23](#)
- setmrg
 - TCS.for, [23](#)
- svstat
 - TCSdrCPM.for, [47](#)
- swindo
 - TCS.for, [23](#)

TCS.for, [16](#)
 ancho, [18](#)
 anstr, [18](#)
 baksp, [18](#)
 cartn, [18](#)
 dasha, [18](#)
 dashr, [18](#)
 drawa, [19](#)
 drawr, [19](#)
 dwindo, [19](#)
 genflg, [19](#)
 home, [19](#)
 linef, [20](#)
 linhgt, [20](#)
 lintrn, [20](#)
 linwdt, [20](#)
 logtrn, [20](#)
 movea, [20](#)
 mover, [21](#)
 newlin, [21](#)
 newpag, [21](#)
 pointa, [21](#)
 pointr, [21](#)
 rel2ab, [22](#)
 rescal, [22](#)
 revcot, [22](#)
 rrotat, [22](#)
 rscale, [22](#)
 seetrm, [23](#)
 seetrn, [23](#)
 setmrg, [23](#)
 swindo, [23](#)
 twindo, [23](#)
 vcursr, [24](#)
 vwindo, [24](#)
 wincot, [24](#)
Tcsbasi1.asm, [31](#)
Tcsbasi2.asm, [39](#)
TCSdrCPM.for, [46](#)
 csize, [46](#)
 dcursr, [46](#)
 initt, [47](#)
 restat, [47](#)
 seeloc, [47](#)
 svstat, [47](#)
 tcslev, [47](#)
Tcsdrivr.asm, [49](#)
Tcsgin.asm, [53](#)
Tcshdc.asm, [55](#)
Tcsital.asm, [57](#)
tcslev
 TCSdrCPM.for, [47](#)
twindo
 TCS.for, [23](#)

vcursr
 TCS.for, [24](#)
vwindo
 TCS.for, [24](#)

wincot
 TCS.for, [24](#)