



```

TTTTTTTTT1  TTTTTTTTTT  YY      YY  DDDDDDDD  EEEEEEEEE  FFFFFFFF
TTTTTTTTTT  TTTTTTTTTT  YY      YY  DDDDDDDD  EEEEEEEEE  FFFFFFFF
TT          TT          YY      YY  DD          DD  EE          FF
TT          TT          YY      YY  DD          DD  EE          FF
TT          TT          YY      YY  DD          DD  EE          FF
TT          TT          YY      YY  DD          DD  EE          FF
TT          TT          YY      YY  DD          DD  EEEEEEEEE  FFFFFFFF
TT          TT          YY      YY  DD          DD  EEEEEEEEE  FFFFFFFF
TT          TT          YY      YY  DD          DD  EE          FF
TT          TT          YY      YY  DD          DD  EE          FF
TT          TT          YY      YY  DD          DD  EE          FF
TT          TT          YY      YY  DD          DD  EE          FF
TT          TT          YY      YY  DDDDDDDD  EEEEEEEEE  FF
TT          TT          YY      YY  DDDDDDDD  EEEEEEEEE  FF

```

```

SSSSSSSS  DDDDDDDD  LL
SSSSSSSS  DDDDDDDD  LL
SS          DD          DD  LL
SS          DD          DD  LL
SS          DD          DD  LL
SS          DD          DD  LL
SSSSSS    DD          DD  LL
SSSSSS    DD          DD  LL
SS          DD          DD  LL
SS          DD          DD  LL
SS          DD          DD  LL
SS          DD          DD  LL
SS          DD          DD  LL
SSSSSSSS  DDDDDDDD  LLLLLLLLLL
SSSSSSSS  DDDDDDDD  LLLLLLLLLL

```

TTT  
/o4  
/o  
/o  
/o  
/o

end

{  
{ Version: 'V04-000'

```

{*****
{*
{* COPYRIGHT (c) 1978, 1980, 1982, 1984 BY
{* DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.
{* ALL RIGHTS RESERVED.
{*
{* THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED
{* ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE
{* INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER
{* COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
{* OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY
{* TRANSFERRED.
{*
{* THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE
{* AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT
{* CORPORATION.
{*
{* DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS
{* SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
{*
{*****

```

```

{++
{ Revision history:
{
{ V03-002 MIRO370 Michael I. Rosenblum 20-Mar-1984
{ Define a read state word (TTY$W_RB_RDSTATE) and the
{ Wrap state bit.
{
{ Create TTYDEF.SDL from TTYDEF.MAR. Move macro definitions
{ to TTYMACS.MAR.
{
{ See TTYMACS.MAR for additional history.
{--

```

```
module $TTYVECDEF;
```

```
/*  
/*  
/*  
/*
```

```
aggregate CLASS_DEF structure PREFIX "CLASS";
```

```
  GETNXT      longword unsigned tag "" /*  
  PUTNXT      longword unsigned tag "" /*  
  SETUP_UCB   longword unsigned tag "" /*  
  DS_TRAN     longword unsigned tag "" /*  
  DDT         longword unsigned tag "" /*  
  READERROR   longword unsigned tag "" /*  
  DISCONNECT  longword unsigned tag "" /*  
  FORK        longword unsigned tag "" /*  
  POWERFAIL   longword unsigned tag "" /*  
  TABLES     longword unsigned tag "" /*
```

```
end CLASS_DEF;
```

```
aggregate PORT_DEF structure PREFIX PORT;
```

```
  STARTIO     longword unsigned tag "" /*  
  DISCONNECT  longword unsigned tag "" /*  
  SET_LINE    longword unsigned tag "" /*  
  DS_SET      longword unsigned tag "" /*  
  XON         longword unsigned tag "" /*  
  XOFF        longword unsigned tag "" /*  
  STOP        longword unsigned tag "" /*  
  STOP2       longword unsigned tag "" /*  
  ABORT       longword unsigned tag "" /*  
  RESUME      longword unsigned tag "" /*  
  SET_MODEM   longword unsigned tag "" /*  
  DMA         longword unsigned tag "" /*  
  MAINT       longword unsigned tag "" /*  
  FORKRET     longword unsigned tag "" /*
```

```
  constant 'LENGTH' equals . tag "" /* must be at end.
```

```
end PORT_DEF;
```

```
end_module $TTYVECDEF;
```

```
module $TTYSYMDEF;
```

```
/*++
```

```
/*
```

```
/* Miscellaneous symbols used by the terminal driver.
```

```
/*
```

```
/*--
```

```
/*
```

```
/* FORK DISPATCHER BIT DEFINITIONS
```

```
/*
```

```
aggregate FORK structure prefix TTYS;
```

```
    FD_UN SOL      bitfield mask; /* SEND UNSOLICITED INPUT MESSAGE
    FD_GETAHD      bitfield mask; /* CREATE A TYPEAHEAD BUFFER
    FD_DISCONNECT  bitfield mask; /* DISCONNECT AND DELIVER HANGUPAST
    FD_PORTFORK    bitfield mask; /* FORK DISPATCH FOR THE PORT DRIVER
    FD_UNLINK      bitfield mask; /* UNLINK PUCB & LUCB (DETACH)
    FD_LINK        bitfield mask; /* LINK PUCB & LUCB (CONNECT)
    FD_BUSY        bitfield mask; /***** MUST REMAIN AT THE END *****)
```

```
end FORK;
```

```
constant (
```

```
    CTRLA,        /* 1
    CTRLB,        /* 2
    CTRLC,        /* 3
    CTRLD,        /* 4
    CTRL E,       /* 5
    CTRL F,       /* 6
    BELL,         /* 7
    BS,           /* 8
    TAB,         /* 9
    LF,          /* 10
    VT,          /* 11
    FF,          /* 12
    CR,          /* 13
    CTRLN,       /* 14
    CTRL O,      /* 15
    CTRL P,      /* 16
    CTRL Q,      /* 17 (XON)
    CTRL R,      /* 18
    CTRL S,      /* 19 (XOFF)
    CTRL T,      /* 20
    CTRL U,      /* 21
    CTRL V,      /* 22
    CTRL W,      /* 23
    CTRL X,      /* 24
    CTRL Y,      /* 25
    CTRL Z,      /* 26
    ESCAPE       /* 27
```

```
) equals 1 increment 1 prefix TTYS tag C;
```

```
constant XON equals 17 prefix TTYS tag C;
```

```
constant XOFF equals 19 prefix TTYS tag C;
```

```
constant BLANK equals 32 prefix TTYS tag C;
```

```
constant DOLLAR equals 36 prefix TTYS tag C;
```

```
constant PLUS equals 43 prefix TTYS tag C;
```

```
constant ZERO equals 48 prefix TTYS tag C;
```

```

constant ONE      equals 49 prefix TTY$ tag C;
constant SCRIPT   equals 96 prefix TTY$ tag C;
constant LOWA     equals 97 prefix TTY$ tag C;
constant LOWZ     equals 123 prefix TTY$ tag C;
constant LOWESC1  equals 125 prefix TTY$ tag C;
constant LOWESC2  equals 126 prefix TTY$ tag C;
constant DELETE   equals 127 prefix TTY$ tag C;
constant NL       equals 128 prefix TTY$ tag C;
constant CSI      equals 155 prefix TTY$ tag C;

```

```

/*
/* Miscellaneous values
/*

```

```

constant MAXPAGLEN equals 255 prefix TTY$ tag C;
constant MAXPAGWID equals 511 prefix TTY$ tag C;
constant HIGHPL    equals 22 prefix TTY$ tag C;

```

```

/*
/* EDIT READ STATES (STORED IN THE MODE FIELD OF THE READ BUFFER)
/*

```

```

constant (
NORMAL,          /* NORMAL CONTROL-R OR CONTROL-U
CLRECHO,        /* ECHO WITH TABS EXPANDED TO SPACES
ECHLINE,        /* ECHO FROM GIVEN STRING
UPDCURSOR,      /* UPDATE THE CURSOR POSITION THEN EXIT
EXITING,        /* EXIT NOW
MOVECURSOR,     /* MOVE THE CURSOR TO ITS FINAL PLACE
CLRREST,        /* CLEAR THE REST OF THE LINE
PRMECHO,        /* ECHO OUT A PROMPT
PRMECHO1,       /* RETURN STATE FOR PROMPT ECHOING
AESECHO,        /* ECHO AES STRING ALONE
RVECHO,         /* ECHO READ VERIFY STRING
SIMCEOL         /* SIMULATE CLEAR TO END OF LINE
) equals 0 increment 1 prefix TTY$ tag K_ER;

```

```

/*
/* EDITING TOKENS
/*

```

```

constant (
CTRLU,          /* CONTROL-U
CTRLR,          /* CONTROL-R
DELEFT,         /* DELETE CHARACTER LEFT
ESCAPE,         /* ESCAPE PREFIX CHARACTER
                /****** END OF THE NORMAL EDITING CHARACTERRSn
BACK_CHAR,      /* BACKUP 1 CHARACTER
FORWARD_CHAR,  /* MOVE FORWARD 1 CHARACTER
MOVE_EOC,       /* MOVE TO THE END OF LINE
MOVE_BOL,       /* MOVE TO THE BEGINNING OF THE LINE
DELETE_WORD,    /* DELETE WORD TO THE LEFT
QUOTING,        /* AND THE QUOTE CHARACTER
RECALL,         /* RECALL THE LAST Command
TOGGEL,         /* TOGGEL BETWEEN INSERT AND OVERSTRIKE MODES
UNUSED,         /* *** MUST REMAIN AT THE END ***
TERMINATE       /* INDICATES CHARACTERS NOT ALLOWED IN EDITING INPUT LINE
) equals 1 increment 1 prefix TTY$ tag K_ET;

```

```

constant EDITNORMAL equals TTY$K_ET_ESCAPE prefix TTY$ tag K;

```

```

/*

```

/\* INTERNAL FUNCTION CODES

/\*  
constant (  
    READ,          /\* READ FUNCTION  
    WRITE,         /\* WRITE FUNCTION  
    SETM,          /\* SET MODE  
    SETC,          /\* SET CHAR  
    HANGUP,        /\* HANGUP  
    MAINT,         /\* MAINTENCE FUNCTION  
    CTRL,          /\* CONTROL ENABLE  
    CONNECT,      /\* CONNECT TO DETACHED TERMINAL  
    DISCON        /\* DISCONNECT FROM ATTACHED TERMINAL  
  ) equals 0 increment 1 prefix TTYS tag C\_FC;/\*  
/\* CHARACTER CHARACTERISTICS/\*  
aggregate CHAR CHAR structure prefix TTYS;  
    CH\_FILE        bitfield length 3;  
    CH\_LOWER       bitfield mask;  
    CH\_SPEC        bitfield mask;  
    CH\_CTRL        bitfield mask;  
    CH\_CTRL3       bitfield mask;  
    CH\_CTRL2       bitfield mask;  
end CHAR\_CHAR;

end\_module \$TTYSYMDDEF;

```
module $TTYRBDEF;
```

```
/*++
```

```
/* Read buffer definitions
```

```
/*
```

```
/* This buffer is allocated everytime a read is issued. The
/* buffer contains all the information necessary to perform this read.
```

```
/*
```

```
/*--
```

```
aggregate TTYRBDEF structure prefix TTY$;
```

```

RB_TXT          longword unsigned;    /* Address of the first character of
                                        /* the read data.
RB_UVA          longword unsigned;    /* READ BUFFER - USER VIRTUAL ADDR
RB_SIZE         word unsigned;        /* READ BUFFER - BLOCK SIZE
RB_SPARE1       byte unsigned;
RB_ECHLEN       byte unsigned;        /* NUMBER OF CHARACTERS TO ECHO
                                        /* WHEN OUTPUTTING FROM ECHSTR
RB_ECHOAREA     quadword unsigned;    /* WORDS TO ECHO CHARACTERS FROM
RB_ECHSTR       longword unsigned;    /* ADDRESS OF THE FIRST CHARACTER
                                        /* TO OUTPUT DURING EDITECHOING.
RB_PIC          longword unsigned;    /* ADDRESS OF THE PICTURE STRING
                                        /* FOR READ VERIFY
RB_TERM         longword unsigned;    /* THE ADDRESS OF THE TERMINATOR BITMASK
RB_MOD          longword unsigned;    /* MODIFIER LONGWORD
RB_AES          longword unsigned;    /* ADDRESS OF THE AES STRING
RB_AESLEN       word unsigned;        /* THE LENGTH OF THE AESSTRING

RB_RDSTATE_OVERLAY union fill;
  RB_RDSTATE     word unsigned;        /* Read state information word
  RB_RDSTATE_BITS structure fill prefix TTY$;
  RS_WRAP        bitfield mask;      /* THE READ HAS WRAPPED EITHER IN THE PROMPT OR INITIAL STRING
end RB_RDSTATE_BITS;
end RB_RDSTATE_OVERLAY;

RB_LIN          longword unsigned;    /* ADDRESS OF THE FIRST CHARACTER ON
                                        /* THIS LINE.
RB_LINOFF       word unsigned;        /* OFFSET FROM THE BEGINNING OF THE
                                        /* LINE TO THE CURSOR POSITION.
RB_LINREST      word unsigned;        /* NUMBER OF CHARACTERS TO THE RIGHT
                                        /* OF THE CURSOR POSITION, USED BY
                                        /* INPUT EDITING
RB_PRMLEN       word unsigned;        /* LENGTH IN BYTES OF THE PROMPT STRING
RB_TIMOS        word unsigned;        /* READ BUFFER - TIMEOUT SECONDS
RB_CPZCUR       word unsigned;        /* CURRENT CURSOR POSITION
RB_CPZORG       word unsigned;        /* READ BUFFER - ORIGINAL HORIZON
RB_TXTOFF       word unsigned;        /* OFFSET FROM THE BEGINNING OF THE
                                        /* DATA TO THE LOCATION OF THE NEXT CHARACTER
RB_PICLEN       word unsigned;        /* the length of the picture string
RB_TXTSIZ       word unsigned;        /* THE LENGTH OF THE READ.
RB_TXTECH       word unsigned;        /* AMOUNT OF INITIAL STRING TO ECHO
RB_MODE         word unsigned;        /* VALUE INDICATING READ EDIT MODE
RB_RVFLR        byte unsigned;        /* CLEAR CHARACTER FOR READ VERIFY
RB_RVFFIL       byte unsigned;        /* READ VERIFY FILL CHARACTER
RB_ESCTKN       word unsigned;        /* ESCAPE TOKEN CHARACTER
RB_PRM_OVERLAY union fill;

```



```
      RB_PRM      address;      /* ADDRESS OF BEGINNING OF THE PROMPT
      RB_DATA     longword unsigned; /* READ BUFFER - DATA
end RB_PRM_OVERLAY;
end TTYRBDEF;
end_module $TTYRBDEF;
```

TT

L1

```
module $TTYISDEF;
```

```
/*++
```

```
/* ITEM LIST STACK STRUCTURE
```

```
/*
```

```
/* DESCRIPTION:
```

```
/* THIS STRUCTURE IS ALLOCATED OFF THE STACK WHEN AN ITEM LIST QIO IS
```

```
/* DETECTED.
```

```
/*--
```

```
aggregate TTYISDEF structure prefix TTY$;
```

```
IS_ACMODE      longword unsigned; /* ACCESS MODE MAXIMIZED WITH THE MODE OF THE CALLER
IS_EDITMODE    longword unsigned; /* PLACE TO KEEP THE MODE
IS_BUF         longword unsigned; /* THE USERS ADDRESS OF HIS BUFFER
IS_BUFLLEN     longword unsigned; /* THE LENGTH OF THE USERS BUFFER
IS_INI         longword unsigned; /* USERS INITIAL STRING ADDRESS
IS_INILEN      longword unsigned; /* LENGTH OF THE INITIAL STRING
IS_INIBUF      longword unsigned; /* length of initial buffer for fallback use
IS_ITMLST      longword unsigned; /* THE USERS ADDRESS OF THE ITEM LIST
/* USED AS A POINTER TO THE NEXT ENTRY
IS_LASTITM     longword unsigned; /* USERS ADDRESS OF THE LAST ITEM
/* CALCULATED FROM BEGINNING ADDRESS AND LENGTH
IS_MODIFY      longword unsigned; /* THE USERS MODIFIER BITS
IS_PIC         longword unsigned; /* USERS ADDRESS OF THE PICTURE STRING
IS_PICLEN      longword unsigned; /* THE LENGTH OF THE PICTURE STRING
IS_PRM         longword unsigned; /* USERS ADDRESS OF THE PROMPT STRING
IS_PRMLEN      longword unsigned; /* THE LENGTH OF THE PROMPT STRING
IS_PRMBUF      longword unsigned; /* length of prompt for fallback use
IS_SPECIFIED   longword unsigned; /* BITMASK OF SPECIFIED ITEM LIST ENTRIES
IS_TERM        longword unsigned; /* THE ADDRESS OF THE USERS TERMINATOR MASK
IS_TERMLEN     longword unsigned; /* THE LENGTH OF THE USERS TERMINATOR MASK
IS_AES         longword unsigned; /* THE ADDRESS OF THE ALTERNATE ECHO STRING
IS_AESLEN      longword unsigned; /* THE LENGTH OF THE ALTERNATE ECHO STRING
IS_TIMEOUT     longword unsigned; /* TIMEOUT VALUE
IS_FILLCHR     word unsigned; /* TWO BYTES SPECIFYING FILL AND CLEAR CHARACTER
IS_INIOFF      word unsigned; /* OFFSET INTO INITIAL STRING FOR ECHOING

constant IS_LENGTH equals . tag K; /* LENGTH
```

```
end TTYISDEF;
```

```
end_module $TTYISDEF;
```

module STTYILDEF;

/\*++

/\* Itemlist Descriptor

/\*

/\* Description:

/\* This set of definitions defines the locations of all the fields  
/\* in the terminal QIO item list.

/\*--

aggregate TTYILDEF structure prefix TTYS;

IL_LEN	word unsigned;	/* THE LENGTH OF THE BUFFER POINTED TO BY ADR
IL_TYPE	word unsigned;	/* THE TYPE CODE OF THIS ITEM
IL_ADR	longword unsigned;	/* THE USER SPECIFIED ADDRESS
IL_RETADR	longword unsigned;	/* VALUE RETURNED ADDRESS

constant IL\_LENGTH equals . tag K; /\* LENGTH

end TTYILDEF;

end\_module STTYILDEF;

TT

.....

.I

.I

.E

.I

.I

.I

.E

.I

.I

L2

```

module $TTYTADEF;
/*++
/* TYPEAHEAD BUFFER
/*
/* DESCRIPTION:
/* THIS BUFFER IS USED TO STORE CHARACTERS BEFORE PROCESSING AND
/* MOVING THEM INTO THE USERS READ BUFFER.
/* ALLOCATED ON UNSOLICITED DATA OR THE FIRST READ POSTED ON A TERMINAL
/* LINE.
/*--

  aggregate TTYTADEF structure prefix TTYS;

    #rcllen = 256;                /* THE LENGTH OF THE RECALL BUFFER

    TA_PUT      longword unsigned; /* PUT POINTER
    TA_GET      longword unsigned; /* GET POINTER
    TA_SIZE     word unsigned;     /* BLOCK SIZE
    TA_TYPE     byte unsigned;     /*
    TA_SPARE1   byte unsigned;
    TA_INAHD    word unsigned;     /* COUNT OF CHARS IN BUFFER
    TA_RCLOFF   word unsigned;     /* NUMBER OF CHARACTERS IN THE RECALL
                                           /* BUFFER USED.
    TA_END      longword unsigned; /* END ADDRESS
    TA_RCLSIZ   word unsigned;     /* THE SIZE OF THE RECALL STRING
    TA_SPARE2   word unsigned;
    TA_RCL      character length #rcllen tag 'A'; /* TYPEAHEAD BUFFER - RECALL BUFFER
    TA_DATA     longword unsigned; /* TYPEAHEAD BUFFER - DATA START

    constant TA_RCLLEN equals #rcllen tag K; /* LENGTH OF RECALL

  end TTYTADEF;
end_module $TTYTADEF;

```

```

module STTYMDMDEF;      /* aka STTYMODEM via hack in TTYMACS.MAR
/*
/*   Modem control state table definitions
/*
/*
/*   state entry definitions
/*
/*   aggregate MODEM_STATE structure prefix MODEMS;
    ST_ONMASK   byte unsigned;      /*output signals to activate
    ST_OFFMASK  byte unsigned;      /*output signals to disable
    ST_TIMER    word unsigned;      /*timer amount to init
    ST_ROUTINE  word unsigned;      /*action routine

    constant ST_LENGTH equals . tag C; /* LENGTH
end MODEM_STATE;
/*
/*   transition definitions
/*
/*   aggregate MODEM_TRANS structure prefix MODEMS;
    TRAN_TYPE      byte unsigned; /* element type
    TRAN_TYPE2     byte unsigned; /* unused element type
    TRAN_NSTATE    word unsigned; /* next state offset from root
    TRAN_OFFMASK   byte unsigned; /* input signals test on
    TRAN_ONMASK    byte unsigned; /* input signals test off

    constant TRAN_LENGTH equals . tag C; /* LENGTH
end MODEM_TRANS;
/*
/*   transition type codes
/*
/*   constant (
    TRAN_DATASET,      /* dataset
    TRAN_TIME,        /* timer
    TRAN_END,         /* end of transition list
    TRAN_DIALTYPE,    /* test for sysgen parameter
    TRAN_DZ11,        /* controller = DZ11
    TRAN_NOMODEM      /* line not enabled for modem
    ) equals 0 increment 1 prefix MODEM tag $C;
/*
/*   argument type codes
/*
/*   constant (
    INIT,             /* init line
    SHUTDOWN,        /* hangup command
    NULL,            /* null, for detecting preset conditions
    DATASET,         /* dataset interrupt
    TIMER,           /* timer expiration
    ) equals 0 increment 1 prefix MODEM tag $C;
/*
/*   aggregate MODEM_BITS structure prefix TIMCTRLS;
    CANCEL          bitfield mask; /*CANCEL TIMER REQUEST

```

TTYDEF.SDL;1

16-SEP-1984 16:46:04.74<sup>10</sup> Page 12

```
ACTIVE bitfield mask; /*TIMER CURRENTLY ACTIVE
end MODEM_BITS;
constant ENABLE equals %X8000 prefix MODEMS tag 'M'; /*mask enable
end_module $TTYMDMDEF;
```

```
module $TTYDEF;
```

```
/*++
```

```
/*
```

```
/* Terminal driver write packet (TWP)
```

```
/*
```

```
/*--
```

```
aggregate TTYWBDEF structure prefix TTY$;
```

```
WB_FLINK      longword unsigned; /*
```

```
WB_BLINK      longword unsigned; /*
```

```
WB_SIZE       word unsigned; /*
```

```
WB_TYPE       byte unsigned; /*
```

```
WB_FIPL       byte unsigned; /*
```

```
WB_FPC        longword unsigned; /*
```

```
WB_FR3        longword unsigned; /*
```

```
WB_FR4        longword unsigned; /*
```

```
WB_MAP        longword unsigned; /*
```

```
WB_NEXT       longword unsigned; /*
```

```
WB_END        longword unsigned; /*
```

```
WB_IRP        longword unsigned; /*
```

```
WB_STATUS     word unsigned; /*
```

```
WB_BCNT       word unsigned; /*
```

```
WB_RETADDR    longword unsigned; /*
```

```
constant WB_LENGTH equals . tag K; /* LENGTH
```

```
constant WB_LENGTH equals . tag C; /* LENGTH
```

```
WB_DATA       longword unsigned; /*
```

```
end TTYWBDEF;
```

```
end_module $TTYDEF;
```

