

```

;
; PCPUT This CP/M program will send a CP/M file to a PC sent via a serial
; port. The program on the PC should send the file in a XModem format/protocol.
; (Use Absolute Telnet).
;
; The program seems to work up to at least 19000 Baud fine. Have not tested
; it yet with faster chips.
; Note this is just the gutted Ward Christenson Modem program,
;
TRUE EQU 0FFH
FALSE EQU NOT TRUE

BASE$PORT EQU 010H ;>>> SETUP FOR SD SYSTEMS I/O8 Board <<<
MODEM$CTL$PORT EQU BASE$PORT ;010H
MODEM$SEND$MASK EQU 4
SEND$READY EQU 4 ;VALUE WHEN READY
MODEM$RECV$MASK EQU 1
RECV$READY EQU 1 ;BIT ON WHEN READY
MODEM$DATA$PORT EQU BASE$PORT+2 ;012H
MODEM$SSC$SELECT EQU 14H ;Port to select 1 of 4 SSC's on the board

KEY$CTL$PORT EQU 0 ;KEYBOARD STATUS
KEY$READY$MASK EQU 2
KEY$READY EQU 2 ;VALUE WHEN KEYBOARD READY
KEY$DATA$PORT EQU 1
ERROR$LIMIT EQU 5 ;MAX ALLOWABLE ERRORS
EXIT$CHAR EQU 'C'-40H ;CHAR TO EXIT FROM T OR C
;
ORG 100H

CALL START ;GO PRINT ID

DB 'Send a File to a PC using a SD '
DB 'Systems IO-8 Serial Board',13,10,'$'
;
;DEFINE ASCII CHARACTERS USED
SOH EQU 1
EOT EQU 4
ACK EQU 6
NAK EQU 15H
LF EQU 10
CR EQU 13
;
START POP D ;GET ID MESSAGE
MVI C,PRINT
CALL BDOS ;PRINT ID MESSAGE
;INIT PRIVATE STACK
LXI H,0 ;HL=0
DAD SP ;HL=STACK FROM CP/M
SHLD STACK ;..SAVE IT
LXI SP,STACK ;SP=MY STACK
;
CALL INIT$ACIA ;MASTER RESET THE ACIA
;GOBBLE UP GARBAGE CHARS FROM THE LINE
MVI B,1 ;TIMEOUT DELAY
CALL RECV
MVI B,1

```

```

    CALL RECV
;
    JMP SEND$FILE ;<<<<<<<<<<<<<<<<<<< Force sending file
;
                                ;INITIALIZE THE SERIAL PORT
INIT$ACIA:
    LXI D,MSG$INIT ;Say Initilizing ACIA
    CALL PRINT$MESSAGE
;
    MVI A,0 ;Select SD Systems Serial Board SSC #1
    OUT MODEM$SSC$SELECT
    MVI A,0CH ;Set to 19200 Baud
    OUT MODEM$CTL$PORT ;Sel Reg 0CH
    MVI A,2H ;<--- NOTE WILL LEAVE CSS AT THIS BAUD RATE WHEN DONE
    OUT MODEM$CTL$PORT
    RET
;
MSG$INIT DB 'SCC #1, PORT 10H & 12H selected. 19200 Baud, with RTS Active.',CR,LF,'$'

;MOVE FCB (SECOND OPERAND ON COMMAND) TO NORMAL FCB LOCATION

MOVE$FCB:
    LXI H,FCB
    LXI D,FCB+16
    MVI B,16
MOVE$LOOP:
    LDAX D
    MOV M,A
    INX D
    INX H
    DCR B
    JNZ MOVE$LOOP
    XRA A ;GET 0
    STA FCB+32 ;ZERO RECORD #
    RET
;
;*****SEND A FILE*****
;
SEND$FILE:
    CALL OPEN$FILE ;OPEN THE FILE
    LXI D,OPENM
    CALL PRINT$MESSAGE
SENDB XRA A ;GET A ZERO
    STA ERRCT ;ZERO ERROR COUNT
                                ;READ SECTOR, SEND IT
    CALL READ$SECTOR
    LDA SECTNO ;INCR SECT NO.
    INR A
    STA SECTNO
                                ;SEND OR REPEAT SECTOR
REPTB LXI D,SECTMSG
    CALL PRINT$MESSAGE
    LDA SECTNO
    CALL HEXO
    CALL CRLF
    MVI A,SOH
    CALL SEND

```

```

        LDA    SECTNO
        CALL   SEND
        LDA    SECTNO
        CMA
        CALL   SEND
        MVI    C,0           ;INIT CKSUM
        LXI    H,80H
SENDC  MOV    A,M
        CALL   SEND
        INX    H
        MOV    A,H
        CPI    1           ;DONE WITH SECTOR?
        JNZ   SENDC

        ;SECTOR SENT, SEND CKSUM
        MOV    A,C           ;GET CKSUM
        CALL   SEND

        ;GET ACK ON SECTOR
        MVI    B,4           ;WAIT 4 SECONDS MAX
        CALL   RECV
        JNC    SNT0         ;NO TIMEOUT
        ;TIMED OUT WAITING FOR ACK
        CALL   TOUT         ;PRINT 'TIMEOUT', ERRCT
DATERR LDA    ERRCT
        INR    A
        STA   ERRCT
        CPI    ERROR$LIMIT
        JC    REPTB         ;REPEAT SECTOR
        ;SECTOR SEND NO GOOD AFTER X TRIES
        CALL   CHECK$FOR$QUIT
        JZ    REPTB         ;KEEP ON TRYIN'
        CALL   ERXIT

        DB    'CAN''T SEND SECTOR - ABORTING',13,10,'$'
SECTMSG DB    'SENDING SECTOR $'

        ;NO TIMEOUT SENDING SECTOR
SNT0  CPI    ACK           ;ACK RECIEVED?
        JZ    SENDB         ;..YES, SEND NEXT SECT
        ;ACK NOT RECIEVED
        CALL   HEXO         ;TYPE CHR IN HEX
        LXI    D,ERR1
        CALL   PRINT$MESSAGE
        JMP    DATERR       ;GO TO DATA ERROR

ERR1  DB    'H RECEIVED, NOT ACK',13,10,'$'
OPENM DB    'FILE OPEN',13,10,'$'
;
;
; S U B R O U T I N E S
;
;OPEN FILE
OPEN$FILE LXI    D,FCB
        MVI    C,OPEN
        CALL   BDOS
        INR    A           ;OPEN OK?
        RNZ    ;GOOD OPEN
        CALL   ERXIT

```

```

        DB      'CAN'T OPEN FILE$'
; - - - - -
PRINT$MESSAGE:
        MVI     C,PRINT
        JMP     BDOS          ;PRINT MESSAGE, RETURN
; - - - - -
;EXIT PRINTING MESSAGE FOLLOWING 'CALL ERXIT'
ERXIT POP     D              ;GET MESSAGE
        CALL   PRINT$MESSAGE ;PRINT IT
EXIT  LHL     STACK         ;GET ORIGINAL STACK
        SPHL   ;RESTORE IT
        RET     ;--EXIT-- TO CP/M
; - - - - -
;MODEM RECV
;-----
RECV  PUSH    D              ;SAVE
MSEC  LXI     D,0BBBBH      ;1 SEC DCR COUNT
;
        MVI     A,55H        ;LOWER RTS
        OUT     MODEM$CTL$PORT
        MVI     A,11101010B
        OUT     MODEM$CTL$PORT
        NOP
        NOP
;
MWTI:  IN      MODEM$CTL$PORT
        ANI     MODEM$RCV$MASK
        CPI     RCV$READY
        JZ      MCHAR        ;GOT CHAR
        DCR     E              ;COUNT DOWN
        JNZ     MWTI         ;FOR TIMEOUT
        DCR     D
        JNZ     MWTI
        DCR     B              ;DCR # OF SECONDS
        JNZ     MSEC
;
;MODEM TIMED OUT RECEIVING
        POP     D              ;RESTORE D,E
        STC     ;CARRY SHOWS TIMEOUT
        RET
;
;GOT MODEM CHAR
MCHAR IN      MODEM$DATA$PORT
        POP     D              ;RESTORE DE
;CALC CHECKSUM
        PUSH   PSW
        ADD    C
        MOV    C,A
;
        MVI     A,5H          ;Raise RTS line
        OUT     MODEM$CTL$PORT ;Sel Reg 5
        MVI     A,11101000B
        OUT     MODEM$CTL$PORT
;
        POP     PSW
        ORA    A              ;TURN OFF CARRY TO SHOW NO TIMEOUT
        RET
; - - - - -

```

```

;MODEM SEND CHAR ROUTINE
;-----
;
SEND  PUSH  PSW          ;CHECK IF MONITORING OUTPUT
      ADD   C            ;CALC CKSUM
      MOV   C,A
SENDW IN   MODEM$CTL$PORT
      ANI   MODEM$SEND$MASK
      CPI   SEND$READY
      JNZ   SENDW
      POP   PSW          ;GET CHAR
      OUT   MODEM$DATA$PORT
      RET

;
;-----
;PRINT TIMEOUT MESSAGE
;-----
;
TOUTM DB  'TIMEOUT $'
TOUT  LXI  D,TOUTM
      CALL PRINT$MESSAGE
PRINT$ERRCT:
      LDA  ERRCT
      CALL HEXO          ;FALL INTO CR/LF

;
CRLF  MVI  A,13
      CALL TYPE
      MVI  A,10

;
TYPE  PUSH  PSW
      PUSH  B
      PUSH  D
      PUSH  H
      MOV   E,A
      MVI  C,WRCON
      CALL  BDOS
      POP   H
      POP   D
      POP   B
      POP   PSW
      RET

;
;HEX OUTPUT
;
HEXO  PUSH  PSW
      RAR
      RAR
      RAR
      RAR
      CALL  NIBBL
      POP   PSW
NIBBL ANI  0FH
      CPI  10
      JC   ISNUM
      ADI  7
ISNUM ADI  '0'
      JMP  TYPE

```

```

;
;MULTIPLE ERRORS, ASK IF TIME TO QUIT
;
CHECK$FOR$QUIT:
    XRA    A            ;GET 0
    STA    ERRCT       ;RESET ERROR COUNT
    LXI    D,QUITM
    CALL   PRINT$MESSAGE
    MVI    C,RDCON
    CALL   BDOS
    PUSH   PSW         ;SAVE CHAR
    CALL   CRLF
    POP    PSW
    CPI    'R'
    RZ                    ;RETURN IF RETRY
    CPI    'r'
    RZ
    CPI    'Q'         ;QUIT?
    JNZ    LCQ
    ORA    A           ;TURN OFF ZERO FLAG
    RET
LCQ:   CPI    'q'
    JNZ    CHECK$FOR$QUIT
    ORA    A           ;TURN OFF ZERO FLAG
    RET

QUITM DB    0DH,0AH,'++MULTIPLE ERRORS ENCOUNTERED.'
       DB    0DH,0AH,'TYPE Q TO QUIT, R TO RETRY:$'
;
;FILE READ ROUTINE
;
READ$SECTOR:
    LXI    D,FCB
    MVI    C,READ
    CALL   BDOS
    ORA    A
    RZ
    DCR    A           ;EOF?
    JNZ    RDERR
                    ;EOF

    XRA    A
    STA    ERRCT
    LXI    D,FSENTM    ;FILE SENT MESSAGE
    CALL   PRINT$MESSAGE
SEOT   MVI    A,EOT
    CALL   SEND
    MVI    B,5         ;WAIT 5 SEC FOR TIMEOUT
    CALL   RECV
    JC     EOTTOT      ;EOT TIMEOUT
    CPI    ACK
    JZ     XFER$CPLT
                    ;ACK NOT RECIEVED

    CALL   HEXO
    LXI    D,ERR1
    CALL   PRINT$MESSAGE
EOTERR LDA    ERRCT
    INR    A

```

```

        STA  ERRCT
        CPI  ERROR$LIMIT
        JC   SEOT
        CALL ERXIT

        DB   'NO ACK RECIEVED ON EOT$',10,13
FSENTM  DB   13,10,'FILE SENT, SENDING EOT''S',10,13,'$'
;
;TIMEOUT ON EOT
;
EOTTOT  CALL  TOUT
        JMP  EOTERR
;
;READ ERROR
;
RDERR  CALL  ERXIT
        DB   '++FILE READ ERROR$'
; - - - - -
;DONE - CLOSE UP SHOP
XFER$CPLT:
        CALL  ERXIT
        DB   13,10,'TRANSFER COMPLETE$'
        DS   40      ;STACK AREA
STACK  DS   2       ;STACK POINTER
RECVD$SECT$NO  DB   0
SECTNO  DB   0      ;CURRENT SECTOR NUMBER
ERRCT  DB   0      ;ERROR COUNT
;
; BDOS EQUATES (VERSION 2)
;
RDCON  EQU  1
WRCON  EQU  2
PRINT  EQU  9
CONST  EQU  11     ;CONSOLE STAT
OPEN   EQU  15     ;0FFH=NOT FOUND
CLOSE  EQU  16     ;   "  "
SRCHF  EQU  17     ;   "  "
SRCHN  EQU  18     ;   "  "
ERASE  EQU  19     ;NO RET CODE
READ   EQU  20     ;0=OK, 1=EOF
WRITE  EQU  21     ;0=OK, 1=ERR, 2=?, 0FFH=NO DIR SPC
MAKE   EQU  22     ;0FFH=BAD
REN    EQU  23     ;0FFH=BAD
STDMA  EQU  26
BDOS   EQU  5
REIPL  EQU  0
FCB    EQU  5CH    ;SYSTEM FCB
;
; END

```