

```

;
; PCPUT.COM This CP/M program will send a file to a PC sent via a serial
; port and write it as an MSDOS file. The program on the PC
; should receive the file in a XModem format/protocol. (Use Absolute Telnet).
;
; The program seems to work up to at least 38,400 Baud fine.
; Note this is just the gutted Ward Christenson Modem program,
; This program can be assembled to utilize the serial ports of the SD-Systems
; Serial IO board or the S100Computers/N8VEM Serial-IO Board.
; It can be easily modified for most other serial ports.
;
;      John Monahan   2/8/2013                (monahan@vitasoft.org)
;
;
;DEFINE ASCII CHARACTERS USED
SOH          EQU    1
EOT          EQU    4
ACK          EQU    6
NAK          EQU   15H
LF           EQU   10
CR           EQU   13

; 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

TRUE        EQU   0FFH
FALSE       EQU   NOT TRUE

SD$SYSTEMS  EQU   FALSE      ;<---True if SD Systems Serial 8-IO Board

IF SD$SYSTEMS
BASE$PORT   EQU   010H      ;>>> SETUP FOR SD SYSTEMS I/O8 Board <<<
MODEM$SSC$SELECT EQU   14H      ;Port to select 1 of 4 SSC's on the board

ELSE
BASE$PORT   EQU   0A1H      ;>>> SETUP FOR S100Computers Board <<<
SPEECH$CTL$PORT EQU   0A0H      ;Serial Ctrl port A for speech synthesizer
SPEECH$DATA$PORT EQU  0A2H      ;Serial Data port A for speech synthesizer

ENDIF

MODEM$CTL$PORT EQU   BASE$PORT      ;A1H or 010H
MODEM$SEND$MASK EQU    4
SEND$READY    EQU    4      ;VALUE WHEN READY
MODEM$RCV$MASK EQU    1
RCV$READY    EQU    1      ;BIT ON WHEN READY
MODEM$DATA$PORT EQU   BASE$PORT+2    ;A3H or 012H

ERROR$LIMIT  EQU    5      ;MAX ALLOWABLE ERRORS
EXIT$CHAR    EQU   'C'-40H    ;CHAR TO EXIT FROM T OR C

      ORG    100H

```



```

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 SNTO ;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$QUIT
JZ REPTB ;KEEP ON TRYIN'
CALL ERXIT
DB 'CAN'T SEND SECTOR - ABORTING',13,10,'$'
;NO TIMEOUT SENDING SECTOR
SNTO CPI ACK ;ACK RECIEVED?
JZ SENDB ;..YES, SEND NEXT SECT
;ACK NOT RECIEVED
CALL HEXO ;TYPE CHR IN HEX
LXI D,ERR1
CALL PRINT$MESSAGE
JMP DATERR ;GO TO DATA ERROR
;
;
;----- S U B R O U T I N E S -----
;
;OPEN FILE
OPEN$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 FROM STACK
      CALL  PRINT$MESSAGE ;PRINT IT
EXIT   LHLD  STACK        ;GET ORIGINAL STACK
      SPHL                ;RESTORE IT
      JMP   0H            ;EXIT -- TO CP/M
```

```
;-----
; SERIAL PORT GET CHARACTER ROUTINE
;-----
```

```
;
RECV   PUSH   D            ;SAVE
      MVI   A,5H          ;Lower RTS line
      OUT  MODEM$CTL$PORT ;Sel Reg 5
      MVI   A,11101010B   ;EAH
      OUT  MODEM$CTL$PORT
      NOP
      NOP
MSEC:  LXI   D,0BBBBH     ;1 SEC DCR COUNT
MWTI:  IN    MODEM$CTL$PORT
      ANI  MODEM$RECV$MASK
      CPI  RECV$READY
      JZ   MCHAR          ;GOT CHAR
      DCR  E              ;COUNT DOWN
      JNZ  MWTI           ;FOR TIMEOUT
      DCR  D
      JNZ  MWTI
      DCR  B              ;DCR # OF SECONDS
      JNZ  MSEC           ;MODEM TIMED OUT RECEIVING
      POP  D              ;RESTORE D,E
      STC                ;CARRY SHOWS TIMEOUT
      RET
                                     ;GOT MODEM CHAR
MCHAR  IN    MODEM$DATA$PORT
      POP  D              ;RESTORE DE
      PUSH PSW            ;CALC CHECKSUM
      ADD  C
      MOV  C,A
      POP  PSW
      ORA  A              ;TURN OFF CARRY TO SHOW NO TIMEOUT
      RET
```

```
;-----
; SERIAL PORT SEND CHARACTER ROUTINE
;-----
```

```
;
SEND   PUSH   PSW          ;CHECK IF MONITORING OUTPUT
      ADD   C              ;CALC CKSUM
      MOV   C,A
SENDW  IN    MODEM$CTL$PORT ;Don't worry PC is always fast enough!
      ANI  MODEM$SEND$MASK
      CPI  SEND$READY
      JNZ  SENDW
      POP  PSW            ;GET CHAR
      OUT  MODEM$DATA$PORT
;
                                     ;Raise RTS line to prevent the next character arriving
      MVI  A,5H          ;while the Z80 is busy processing info
      OUT  MODEM$CTL$PORT ;Sel Reg 5
      MVI  A,11101000B   ;E8H
      OUT  MODEM$CTL$PORT
      RET
```

```
IF NOT SD$SYSTEMS
```

```
;-----
; SPEECH PORT, SEND CHARACTER ROUTINE
;-----
```

```
;
SPEAK  PUSH   PSW          ;CHECK IF MONITORING OUTPUT
```

```

SPEAKW IN    SPEECH$CTL$PORT
        ANI    MODEM$SEND$MASK
        CPI    SEND$READY
        JNZ    SPEAKW
        POP    PSW          ;GET CHAR
        OUT    SPEECH$DATA$PORT
        RET

MSG:    MOV    A,M          ;Speak string at [HL] up to '$'
        INX    H
        CPI    CR          ;Note CR ends string AND initializes V-Stamp chip to speak
        JZ     DONE$SP
        CALL   SPEAK
        JMP    MSG
DONESP: CALL   SPEAK
        RET
ENDIF

```

```

;PRINT TIMEOUT MESSAGE
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

```

```
;----- 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

```

```
;TIMEOUT ON EOT
```

```

EOTTOT CALL   TOUT
JMP    EOTERR

```

```
;
```

```
;READ ERROR
```

```

RDERR  CALL   ERXIT
DB     '++FILE READ ERROR$'

```

```
;DONE - CLOSE UP SHOP
```

```
XFER$CPLT:
```

```
IF NOT SD$SYSTEMS
```

```

LXI    H,FINISH$MSG ;Speak downloading finished
CALL   SMSG

```

```
ENDIF
```

```

CALL   ERXIT
DB     13,10,'TRANSFER COMPLETE$'

```

```
;----- DATA AREA -----
```

```
;Initilization table for SCC registers
```

```

SCCINIT:DB    04H    ;1, Point to WR4
DB    44H    ;2, X16 clock,1 Stop,NP

```

```
;
```

```

DB    03H    ;3, Point to WR3

```

