

```

;
; PCGET This CP/M program will obtain a file from a PC sent via a serial
; port and write it to file on the CP/M system. 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    'Get a File from 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

```



```

RECV$SECT$ERR:                ;PURGE THE LINE OF INPUT CHARS
    MVI    B,1                ;1 SEC W/NO CHARS
    CALL   RECV
    JNC    RECV$SECT$ERR      ;LOOP UNTIL SENDER DONE
    MVI    A,NAK
    CALL   SEND                ;SEND NAK
    LDA    ERRCT
    INR    A
    STA    ERRCT
    CPI    ERROR$LIMIT
    JC     RECV$HDR
    CALL   CHECK$FOR$QUIT
    JZ     RECV$HDR
    CALL   ERXIT

    DB     '++UNABLE TO GET VALID HEADER',0DH,0AH,'$'
RMSG    DB     'WAITING FOR SECTOR #'$'

                                ;GOT CHAR - MUST BE SOH
RHNT0   CPI    SOH
    JZ     GOT$SOH
    ORA    A                    ;00 FROM SPEED CHECK?
    JZ     RECV$HDR
    CPI    EOT
    JZ     GOT$EOT

                                ;DIDN'T GET SOH -
    CALL   HEXO
    LXI    D,ERRSOH
    CALL   PRINT$MESSAGE
    JMP    RECV$SECT$ERR

ERRSOH   DB     'H RECEIVED, NOT SOH',0DH,0AH,'$'
GOT$SOH:
    MVI    B,1
    CALL   RECV
    JC     RECV$HDR$TIMEOUT
    MOV    D,A                    ;D=BLK #
    MVI    B,1
    CALL   RECV                ;GET CMA'D SECT #
    JC     RECV$HDR$TIMEOUT
    CMA
    CMP    D                    ;GOOD SECTOR #?
    JZ     RECV$SECTOR
                                ;GOT BAD SECTOR #
    LXI    D,ERR2
    CALL   PRINT$MESSAGE
    JMP    RECV$SECT$ERR

ERR2    DB     '++BAD SECTOR # IN HDR',0DH,0AH,'$'
;
RECV$SECTOR:
    MOV    A,D                    ;GET SECTOR #
    STA    RECVD$SECT$NO
    MVI    C,0                    ;INIT CKSUM
    LXI    H,80H                ;POINT TO BUFFER
RECV$CHAR:
    MVI    B,1                    ;1 SEC TIMEOUT

```

```

CALL RECV          ;GET CHAR
JC RECV$HDR$TIMEOUT
MOV M,A           ;STORE CHAR
INR L            ;DONE?
JNZ RECV$CHAR

                                ;VERIFY CHECKSUM
MOV D,C          ;SAVE CHECKSUM
MVI B,1         ;TIMEOUT
CALL RECV       ;GET CHECKSUM
JC RECV$HDR$TIMEOUT
CMP D          ;CHECK
JNZ RECV$CKSUM$ERR
;
;GOT A SECTOR, WRITE IF = 1+PREV SECTOR
;
LDA RECVD$SECT$NO
MOV B,A        ;SAVE IT
LDA SECTNO    ;GET PREV
INR A        ;CALC NEXT SECTOR #
CMP B        ;MATCH?
JNZ DO$ACK

;GOT NEW SECTOR - WRITE IT
LXI D,FCB
MVI C,WRITE
CALL BDOS
ORA A
JNZ WRITE$ERROR
LDA RECVD$SECT$NO
STA SECTNO    ;UPDATE SECTOR #
DO$ACK MVI A,ACK
CALL SEND
JMP RECV$LOOP
;
WRITE$ERROR:
CALL ERXIT

DB '++ERROR WRITING FILE',0DH,0AH,'$'
;
RECV$CKSUM$ERR:
LXI D,ERR3
CALL PRINT$MESSAGE
JMP RECV$SECT$ERR

ERR3 DB '++BAD CKSUM ON SECTOR'
DB 0DH,0AH,'$'
;
GOT$EOT:
MVI A,ACK    ;ACK THE EOT
CALL SEND
LXI D,FCB
MVI C,CLOSE
CALL BDOS
INR A
JNZ XFER$CPLT
CALL ERXIT

```

```

        DB      '++ERROR CLOSING FILE$'
;
ERASE$OLD$FILE:
        LXI    D,FCB
        MVI    C,SRCHF          ;SEE IF IT EXISTS
        CALL   BDOS
        INR    A                ;FOUND?
        RZ     ;NO, RETURN
        LXI    D,EXIST
        CALL   PRINT$MESSAGE
ERAY:   CALL   CRLF
        LXI    D,FCB
        MVI    C,ERASE
        CALL   BDOS
        RET
;
EXIST DB      '+++NOTE OLD FILE HAS BEEN ERASED+++$'
;
MAKE$NEW$FILE:
        LXI    D,FCB
        MVI    C,MAKE
        CALL   BDOS
        INR    A                ;FF=BAD
        RNZ   ;OPEN OK

;DIRECTORY FULL - CAN'T MAKE FILE
        CALL   ERXIT
        DB      '++ERROR - CAN''T MAKE FILE',0DH,0AH
        DB      '++DIRECTORY MUST BE FULL',0DH,0AH,'$'
;
; S U B R O U T I N E S
;
;OPEN FILE
OPEN$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  LHLD  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,5H          ;Lower RTS line
OUT    MODEM$CTL$PORT ;Sel Reg 5
MVI    A,11101010B
OUT    MODEM$CTL$PORT
NOP
NOP

;
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$PORT
POP    D          ;RESTORE DE
;CALC CHECKSUM
PUSH  PSW
ADD   C
MOV   C,A
;
MVI    A,5H          ;Raise RTS line
OUT    MODEM$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$',10,13
FSENTM  DB    13,10,'FILE SENT, SENDING EOT''S',10,13,'$'
ERR1    DB    'H RECEIVED, NOT ACK',13,10,'$'
;
;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

```