```
;
;  PCGET  This CP/M program will obtain a file from a PC sent via a USB
;  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 assumes the presence of the DLP-USB Controller on
;  the S100Computes Serial IO Board.
;

;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

USB$DATAPORT      EQU   0ACH        ;Data port for the DLP Controller
USB$STATUS$PORT   EQU   0AAH        ;Status port for DLP Controller (Port C
of 8255)
USB$RXE           EQU   80H         ;If bit 7 = 0, data available
USB$TXE           EQU   40H         ;If Bit 6 = 0, data can be written to
chip for transmission

SPEECH$CTL$PORT   EQU   0A0H        ;Serial Ctrl port A for speech
synthesizer
SPEECH$DATA$PORT EQU    0A2H        ;Serial Data port A for speech
synthesizer
SPEECH$SEND$MASK EQU    4
SPEECH$SEND$READY EQU   4           ;VALUE WHEN READY

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

```
      ORG   100H

START LXI   D,SIGNON    ;GET SIGNON MESSAGE
      MVI   C,PRINT
      CALL  BDOS        ;PRINT 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

                        ;GOBBLE UP GARBAGE CHARS FROM THE LINE
      MVI   B,1         ;TIMEOUT DELAY
      CALL  RECV
;
      JMP   RECV$FILE   ;<<<<<<<<<<<<< Force recieve file
;
                        ;INITITIALIZE THE SCC SERIAL PORT


;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$LOOP
      XRA   A           ;GET 0
      STA   FCB+32            ;ZERO RECORD #
      RET
;
;**************RECEIVE FILE****************
;
RECV$FILE:
      CALL  ERASE$OLD$FILE
      CALL  MAKE$NEW$FILE
      PUSH  H
      LXI   H,DOWNLOAD$MSG    ;Speak downloading file
      CALL  SMSG
      POP   H

RECV$LOOP:
      XRA   A           ;GET 0
      STA   ERRCT       ;INIT ERROR COUNT
RECV$HDR:
      LXI   D,RMSG
      CALL  PRINT$MESSAGE
      LDA   SECTNO
      INR   A
      CALL  HEXO
      CALL  CRLF
```

```
      MVI   B,5         ;5 SEC TIMEOUT
      CALL  RECV
      JNC   RHNTO       ;NO TIMEOUT
RECV$HDR$TIMEOUT:
      CALL  TOUT        ;PRINT TIMEOUT

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$HDR
      CALL  ERXIT
      DB    '++UNABLE TO GET VALID HEADER',0DH,0AH,'$'

                        ;GOT CHAR - MUST BE SOH
RHNTO 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

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


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


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       ;We are done
      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$MESSAGE
      PUSH  H
      LXI   H,ERASE$MSG ;Speak, erased old file
      CALL  SMSG
      POP   H
ERAY: CALL  CRLF
      LXI   D,FCB
      MVI   C,ERASE
      CALL  BDOS
      RET


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 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
MSEC: LXI   D,0BBBBH    ;1 SEC DCR COUNT
MWTI: IN    USB$STATUS$PORT
      ANI   USB$RXE
      JZ    MCHAR       ;GOT CHAR
      DCR   E           ;COUNT DOWN
      JNZ   MWTI        ;FOR TIMEOUT
      DCR   D
      JNZ   MWTI
      DCR   B           ;DCR # OF SECONDS
      JNZ   MSEC        ;MODEM TIMED OUT RECEIVING
      POP   D           ;RESTORE D,E
      STC               ;CARRY SHOWS TIMEOUT
      RET
                        ;GOT MODEM CHAR
MCHAR IN    USB$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    USB$STATUS$PORT   ;Don't worry PC is always fast enough!
      ANI   USB$TXE
      JNZ   SENDW
      POP   PSW         ;GET CHAR
      OUT   USB$DATA$PORT
      RET

;-----------------------------------------------------
;   SPEECH PORT, SEND CHARACTER ROUTINE
;-----------------------------------------------------
;
SPEAK PUSH  PSW         ;CHECK IF MONITORING OUTPUT
SPEAKW      IN    SPEECH$CTL$PORT
      ANI   SPEECH$SEND$MASK
      CPI   SPEECH$SEND$READY
      JNZ   SPEAKW
      POP   PSW         ;GET CHAR
      OUT   SPEECH$DATA$PORT
      RET

SMSG: MOV   A,M         ;Speak string at [HL] up to '$'
      INX   H
```

```
      CPI   CR          ;Note CR ends string AND initilizes V-Stamp chip to
speak
      JZ    DONE$SP
      CALL  SPEAK
      JMP   SMSG
DONESP:     CALL  SPEAK
      RET

;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$CPLT:
      LXI   H,FINISH$MSG      ;Speak downloading finished
      CALL  SMSG
      CALL  ERXIT
      DB    13,10,'TRANSFER COMPLETE$'

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

SIGNON:     DB    'Get a File from a PC using the USB Port of the
S100Computers '
      DB    'Serial IO Board',13,10
      DB    'Zilog SCC Ports A0H & A2H. 8255 Port AAH, USB Port ACH.',13,10
      DB    '(Note assumes all ports are already initilized.)',13,10,'$'

DOWNLOAD$MSG DB   'Downloading file Started.',CR
SPEED$MSG    DB   '38,400 Baaud.',CR
FINISH$MSG   DB   'Down loading of file complete.  No Errors',CR
ERASE$MSG    DB   'Old file erased',CR

RMSG  DB    'WAITING FOR SECTOR #$'
ERRSOH      DB    'H RECEIVED, NOT SOH',0DH,0AH,'$'
ERR2  DB    '++BAD SECTOR # IN HDR',0DH,0AH,'$'
ERR3  DB    '++BAD CKSUM ON SECTOR',0DH,0AH,'$'
EXIST DB    '+++NOTE OLD FILE HAS BEEN ERASED+++$'
TOUTM DB    'TIMEOUT $'
QUITM DB    0DH,0AH,'++MULTIPLE ERRORS ENCOUNTERED.'
      DB    0DH,0AH,'TYPE Q TO QUIT, R TO RETRY:$'
FSENTM      DB    13,10,'FILE SENT, SENDING EOT''S',10,13,'$'
ERR1  DB    'H RECEIVED, NOT ACK',13,10,'$'


      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
;
; END
```