

```
MODULE ; XMODEM file transfer
; 2/18/86
```

```
; check out disk I/O, text and binary
; what is convention for last byte when length is multiple of 256
; figure out monitoring of keyboard etc. before/during xfer.
; sort out declarations
; decide on globals
; compile must be case sensitive if use LF here (ACSTERM has lf)
;SET $4CA=$FF; --will this provoke symbol overflow?
```

```
PROC PBlock(BYTE ARRAY block, CARD size)
  CARD j
  BYTE DSPFLG=$2FE
  DSPFLG=1 ;write control char to screen (except EOL)
  FOR j=0 TO size-1 DO
    Put(block(j)) ; block(j)=0
  OD
  DSPFLG=0
  PutE()
RETURN
```

```
MODULE; BLKIO-----
; Copyright (c) 1983, 1984, 1985 by Action Computer Services (ACS)
```

```
BYTE CIO_status
```

```
CHAR FUNC CIO=*(BYTE dev, CARD addr,
  size, BYTE cmd, aux1, aux2)
~[$29$F$85$A0$86$A1$A$A$A$A$AA$A5$A5
$9D$342$A5$A3$9D$348$A5$A4$9D$349
$A5$A6$F0$8$9D$34A$A5$A7$9D$34B$98
$9D$345$A5$A1$9D$344$20$E456
$8C CIO_status$C0$88$D0$6$98$A4$A0
$99 EOF$A085$60]
```

```
CARD FUNC ReadBlock=*(BYTE dev, CARD addr, size)
~[$48$A9$7$85$A5$A9$0$85$A6$A5$A3$5$A4
$D0$6$85$A0$85$A1$68$60$68$20 CIO
$BD$348$85$A0$BD$349$85$A1$60]
```

```
PROC WriteBlock=*(BYTE dev, CARD addr, size)
; Writes size bytes from addr to dev.
; Status is saved in CIO_status.
~[$48$A9$B$85$A5$A9$0$85$A6$A5$A3$5$A4
$D0$2$68$60$68$4C CIO]
```

```
MODULE ; part of BLOCKIO
```

```
; These will be from ACSTERM:
DEFINE modem = "5"
DEFINE file = "3"
DEFINE STRING = "CHAR ARRAY"
DEFINE ASCII = "$0"
DEFINE EOL = "$9B"
```

```
CARD ARRAY end(0)
BYTE ARRAY fbuf
BYTE baud=~[14], fmode
```

```

STRING Rdev(0)="R:"

BYTE FUNC MStatus=*( )
    BYTE QLi=$2EB; DVSTAT+1, input queue length
    XIO(modem,0,$D,0,0,Rdev)
RETURN (QLi)

PROC OpenModem(BYTE trans)
    Close(modem)
    Open(modem,Rdev,13,0)
    XIO(modem,0,36,baud,0,Rdev)
    XIO(modem,0,38,trans,0,Rdev)
    XIO(modem,0,40,0,0,Rdev) ; concurrent
RETURN

PROC MyClose(BYTE chan)
    Close(modem)
    Close(chan) ; more than this in ACSTERM?
RETURN

PROC OpenFile(STRING msg)
    BYTE ARRAY spec(30)
    Print(msg)
    InputMD(0,spec,30)
    MyClose(file)
    Open(file,spec,fmode,0)
RETURN

PROC GetKey( )
    BYTE CH=$2FC, c
    IF CH<>255 THEN
        c=GetD(7)
        IF c=EOL THEN c=$D FI
        PutD(modem,c)
    FI
RETURN

MODULE ; -----

DEFINE FALSE = "0"
DEFINE EOT = "4"
DEFINE SOH = "1"
DEFINE ACK = "6"
DEFINE LF = "$A"
DEFINE CR = "$D"
DEFINE NAK = "$15"
DEFINE SUB = "$1A"
DEFINE TIMEOUT = "$FFFF"
DEFINE RETRYMAX = "10"
DEFINE ERRORMAX = "10"

BYTE j, CheckSum, SectNum, TotErr, Errors
BYTE transx=~[0], xeof
CHAR ARRAY block(128)
INT dbufp ; # of data bytes in fbuf, index of next
CARD ibuf
CARD BLen=~[2000]; length of buffer (must be >2*128?)

CARD FUNC Receive(BYTE wait)

```

```

CARD CDTMV3=$21C ; system timer counts down to 0
BYTE CONSOL=$D01F
CDTMV3 = 60*wait
DO
  GetKey()
  IF MStatus() THEN
    RETURN (GetD(modem))
  FI
  IF CONSOL!7 THEN CDTMV3 = 60 FI ; force timer to 1 sec.
UNTIL CDTMV3=0 OD
RETURN (TIMEOUT)

PROC Send(CHAR c)
  PutD(modem,c)
RETURN

PROC PurgeLine(BYTE wait)
  DO UNTIL Receive(wait)=TIMEOUT OD
RETURN

; -----
PROC WBuf() ; 128 bytes from block to disk buffer
; Must set dbufp=xeof=0 before 1st call.
; Caller must open and close file.
; Can't write out a block until next call, because don't know
; a block is last until EOT is received instead of next block.
CARD j, len
  IF xeof THEN ; preceeding block was final.
;print("EOF")
    IF transx=ASCII THEN
      FOR j=dbufp-128 TO dbufp-1 DO
        IF fbuf(j)=SUB THEN EXIT FI
      OD
      dbufp = j
    ELSE
      dbufp ==- (128-fbuf(dbufp-1))
    FI
  ELSE
;print("not EOF")
    FOR j=0 TO 127 DO
      fbuf(dbufp) = block(j)
      dbufp ==+ 1
    OD
  FI
;PBlock(block,dbufp)
;printf("dbufp=%U%E",dbufp)
  IF dbufp>Blen-128 OR xeof<>0 THEN ; flush buffer to disk
    IF transx=ASCII THEN
      ;replace CR-LF by EOL
      ;Don't touch a trailing CR, as LF might be in next block.
      ibuf = 0
      FOR j=0 TO dbufp-1 DO
        IF j<=dbufp-2 THEN
          IF (fbuf(j)&$7F)=CR AND fbuf(j+1)=LF THEN
            j ==+ 1
            fbuf(j) = EOL
          FI
        FI
      FI
      fbuf(ibuf) = fbuf(j)

```

```

        ibuf ==+ 1
    OD
    dbufp = ibuf
FI
Close(modem)
IF xeof THEN len = dbufp ELSE len = dbufp-128 FI
WriteBlock(file,fbuf,len)
OpenModem(32)
;Move remaining dbufp-len bytes to front of fbuf
FOR j=len TO dbufp-1 DO
    fbuf(j-len) = fbuf(j)
OD
dbufp ==- len
FI
RETURN ; WBuf

PROC RecFile()
    CARD ch, FirstChar
    BYTE SectCurr, ErrorFlag
    BYTE CONSOL=$D01F

    fmode = 8
    OpenFile("XMODEM download to file: ")
    dbufp = 0 : xeof = 0
    OpenModem(32)

    SectNum = 0
    Errors = 0 ; on current sector
    TotErr = 0 ; on file

    PurgeLine(0)
    Send(NAK)

DO
    ErrorFlag = FALSE
    DO
        FirstChar = Receive(10)
        IF SectNum=0 OR (CONSOL&2)=0 THEN
            Put(FirstChar) ;** debug -FOX can type to screen
        FI
        UNTIL FirstChar=SOH OR FirstChar=EOT OR FirstChar=TIMEOUT OD
        IF FirstChar=TIMEOUT THEN
            ErrorFlag = 'T
; ELSEIF FirstChar=EOT THEN
;     EXIT
        ELSEIF FirstChar=SOH THEN
            SectCurr = Receive(1)
            IF (SectCurr + Receive(1))=$FF THEN ; good sector number
                IF SectCurr=(SectNum+1) THEN
                    CheckSum = 0
                    FOR j=0 TO 127 DO
                        ch = Receive(1)
                        IF ch=TIMEOUT THEN
                            ErrorFlag = 'T
                            EXIT
                        FI
                        block(j) = ch
                        CheckSum = CheckSum+ch
                    OD

```

```

        IF CheckSum=Receive(1) THEN
            SectNum = SectCurr
            Printf("Rec'd %U after %U tries%E%C",SectNum,Errors,$1C)
            Errors = 0
            WBuf()
            Send(ACK)
        ELSE ; bad checksum ***or timeout in block
            ErrorFlag = 'C
        FI
    ELSEIF SectCurr=SectNum THEN ; already received this
        PurgeLine(1)
        Send(ACK)
    ELSE ; lost a sector
        ErrorFlag = 'S
    FI
ELSE ; bad header
    ErrorFlag = 'H
FI
FI
IF ErrorFlag THEN
    Errors ==+ 1
    IF SectNum THEN TotErr ==+ 1 FI
    PurgeLine(1)
    Printf("Awaiting %U (try=%U, Errs=%U, type %C)%E",
        SectNum,Errors,TotErr,ErrorFlag)
    Send(NAK)
FI
UNTIL FirstChar=EOT OR Errors=ERRORMAX OD
IF FirstChar=EOT AND Errors<ERRORMAX THEN
    Send(ACK)
    xeof=1
    WBuf() ; write buffer, close file
    Printf("%EDone")
ELSE
    Printf("%EAborting")
FI
MyClose(file)
RETURN ; RecFile

```

; -----

```

BYTE FUNC RBuf() ; read 128 bytes into block.
BYTE i
; N.B.! set ibuf=dbufp=xeof=0 before 1st call
IF xeof THEN RETURN(0) FI ; no more blocks
i = 0
WHILE i<128 DO
    IF ibuf=dbufp THEN ; no more data
        IF EOF(file) THEN ; already got EOF
            xeof = 1 ; flag for NEXT call to RBuf
            EXIT
        ELSE
            Close(modem)
            dbufp = ReadBlock(file,fbuf,BLen-1) ; could be zero
            ibuf = 0
            IF CIO_status=$88 THEN ; EOF
                CIO_status = 1 ; indicate OK
            IF transx=ASCII THEN
                fbuf(dbufp) = SUB ; CP/M & MSDOS EOF

```

```

        dbufp ==+ 1
    FI
    FI
    OpenModem(32)
;PrintF("read %U bytes,xeof=%U,EOF=%U%E%E",dbufp,xeof,EOF(file))
    IF dbufp=0 THEN EXIT FI
    FI
    FI
    IF fbuf(ibuf)=EOL AND transx=ASCII THEN
        block(i) = CR
        fbuf(ibuf) = LF ; to send next
    ELSE
        block(i) = fbuf(ibuf)
        ibuf ==+ 1
    FI
    i ==+ 1 ; could make this a FOR loop??
OD
j = i
WHILE i < 128 DO ; fill out last block with number of data bytes
    block(i) = j
    i ==+ 1
OD
RETURN (1) ;RBuf

PROC SndFile()
    BYTE attempts=Errors, ch
    BYTE CONSOL=$D01F

    fmode = 4
    OpenFile("XMODEM upload of file: ")
    ibuf = 0 : dbufp = 0 : xeof=0
    OpenModem(32)

    PurgeLine(0)
    attempts = 0
    TotErr = 0

    PrintE("Await NAK or press start")
    WHILE Receive(10)<>NAK AND attempts<8 AND CONSOL=7 DO ; await initial NAK
        attempts ==+ 1
        PrintF("%CTimeout %U%E", $1C, attempts)
    OD
    IF attempts=8 THEN
        PrintE("Timed out before initial NAK")
        MyClose(file)
        RETURN
    FI
    attempts = 0
    SectNum = 1

    WHILE RBuf()<>0 AND attempts<RETRYMAX DO ; blocks
        IF CIO_status<>1 THEN
            PrintF("DOS error %U%E", CIO_status)
            EXIT
        FI
        attempts = 0
        DO ; send block
;        PrintF("%Cblock %U%E", $1C, SectNum)
            Send(SOH)

```

```

    Send(SectNum)
    Send($FF-SectNum)
    CheckSum = 0
    FOR j = 0 TO 127 DO
        ch=block(j)
        Send(ch)
        CheckSum = CheckSum+ch
    OD
    Send(CheckSum)
    PurgeLine(0)
    attempts ==+ 1
    TotErr == +1
    UNTIL Receive(10)=ACK OR attempts=RETRYMAX OD
    SectNum ==+ 1
    TotErr ==- 1
OD ; loop on blocks
IF attempts=RETRYMAX THEN
    PrintE("No ACK on sector")
ELSE
    attempts=0
    DO
        Send(EOT)
        PurgeLine(0)
        attempts ==+ 1
    UNTIL Receive(10)=ACK OR attempts=RETRYMAX OD
    IF attempts=RETRYMAX THEN
        PrintE("No ACK on EOT")
    FI
FI
MyClose(file)
PrintF("Done with %U retries%E",TotErr)
RETURN ; SndFile

```

```

; -----
PROC Main()
    BYTE ch
    fbuf = end
    transx = ASCII
;transx=1 ; BINARY
    ch = GetD(7)&$DF
    IF ch='R THEN
        RecFile()
    ELSEIF ch='T OR ch='S THEN
        SndFile()
    FI
    Close(modem)
RETURN

```