

# CIO Library with Note and Point for Kyan Pascal#

(\* FILELIB.I - Routines to allow the use of standard ATARI I/O calls in KYAN Pascal programs.

FileOpen - Exactly like the ATARI Basic OPEN command.  
FileClose - Exactly like the ATARI Basic CLOSE command.  
FileGet - Replace the parameters REC and RECLen with the name and length of your record and this routine will read one record from the file.  
FileGetChar - Reads one character from the file.  
FilePut - The inverse of FileGet.  
FilePutChar - The inverse of FileGetChar.  
FileNote - Exactly like the ATARI Basic NOTE command.  
FilePoint - Exactly like the ATARI Basic POINT command.

I don't recommend that you mix these routines with the standard Pascal I/O routines. They were written for use in a small database program that DEMANDED true random access files. Use them with care as none of the protection provided by Pascal is present here. <Mike Long 9/19/86> \*)

```
Procedure FileOpen(IOCbnum,IOdir,Auxbyte : Integer;  
                  Var FileName : String15);
```

```
Begin  
#a
```

```
ICHID EQU $0340  
ICDNO EQU $0341  
ICCMD EQU $0342  
ICSTA EQU $0343  
ICBAL EQU $0344  
ICBAH EQU $0345  
ICPTL EQU $0346  
ICPTH EQU $0347  
ICBLL EQU $0348  
ICBLH EQU $0349  
ICAX1 EQU $034A  
ICAX2 EQU $034B  
ICAX3 EQU $034C  
ICAX4 EQU $034D  
ICAX5 EQU $034E  
ICAX6 EQU $034F
```

```
CIOV EQU $E456
```

```
; _T = LSB of ^FileName  
; _T+1 = MSB of ^FileName  
; _T+2 = LSB of AuxByte  
; _T+3 = MSB of AuxByte (Unused)  
; _T+4 = LSB of IOdir  
; _T+5 = MSB of IOdir (Unused)  
; _T+6 = LSB of IOCbnum  
; _T+7 = MSB of IOCbnum (Unused)
```

```
TXA ;Save X register
```

```

    PHA
    LDX #7          ;Copy heap
    LDY #12
CL1  LDA (_SP),Y
    STA _T,X
    DEY
    DEX
    BPL CL1
    LDA _T+6       ;Get IOCB #
    ASL A          ;Multiply by 16
    ASL A
    ASL A
    ASL A
    TAX            ;Move to X register
    LDA #3         ;Open command
    STA ICCMD,X
    LDA _T         ;Filename address
    STA ICBAL,X
    LDA _T+1
    STA ICBAH,X
    LDA _T+4       ;Data direction
    STA ICAX1,X
    LDA _T+2       ;Aux byte
    STA ICAX2,X

    JSR CIOV       ;Do the I/O
    PLA            ;Restore X register
    TXA

```

```

#
End; (* FileOpen *)

```

```

Procedure FileClose(IOCBnum : Integer);

```

```

Begin

```

```

#a

```

```

; _T    = LSB of IOCBnum
; _T+1  = MSB of IOCBnum (Unused)

```

```

    TXA            ;Save X register
    PHA
    LDX #1         ;Copy heap
    LDY #6
CL2  LDA (_SP),Y
    STA _T,X
    DEY
    DEX
    BPL CL2
    LDA _T         ;Get IOCB #
    ASL A          ;Multiply by 16
    ASL A
    ASL A
    ASL A
    TAX            ;Move to X register
    LDA #12        ;Close command
    STA ICCMD,X
    JSR CIOV       ;Do the I/O
    PLA            ;Restore X register

```

TAX

#  
End; (\* FileClose \*)

Procedure FileGet(IOCbnum : Integer;  
                  Var Rec : RecType;  
                  RecLen : Integer);

Begin  
#a

; \_T = LSB of RecLen  
; \_T+1 = MSB of RecLen  
; \_T+2 = LSB of ^Rec  
; \_T+3 = MSB of ^Rec  
; \_T+4 = LSB of IOCbnum  
; \_T+5 = MSB of IOCbnum (Unused)

          TXA                 ;Saved X register  
          PHA  
          LDX #5               ;Copy heap  
          LDY #10  
CL3 LDA (\_SP),Y  
      STA \_T,X  
      DEY  
      DEX  
      BPL CL3  
      LDA \_T+4                 ;Get IOCb #  
      ASL A                   ;Multiply by 16  
      ASL A  
      ASL A  
      ASL A  
      TXA                     ;Move to X register  
      LDA #7                  ;Get record command  
      STA ICCMD,X  
      LDA \_T+2                 ;Data address  
      STA ICBAL,X  
      LDA \_T+3  
      STA ICBAH,X  
      LDA \_T                  ;Data length  
      STA ICBLL,X  
      LDA \_T+1  
      STA ICBLH,X  
      JSR CIOV                ;Do the I/O  
      PLA                     ;Restore X register  
      TXA

#  
End; (\* FileGet \*)

Procedure FileGetChar(IOCbnum : Integer;  
                      Var Byte : Char);

Begin  
#a

```

; _T    = LSB of ^Byte
; _T+1 = MSB of ^Byte
; _T+2 = LSB of IOCBnum
; _T+3 = MSB of IOCBnum

        TXA                ; Save X register
        PHA
        LDX #3             ; Copy heap
        LDY #8
CL4     LDA (_SP),Y
        STA _T,X
        DEY
        DEX
        BPL CL4
        LDA _T+2           ;Get IOCB #
        ASL A              ;Multiply by 16
        ASL A
        ASL A
        ASL A
        TAX                ;Move to X register
        LDA #7             ;Get Record command
        STA ICCMD,X
        LDA #0             ;Single byte get
        STA ICBLH,X
        STA ICBLH,X
        JSR CIOV           ;Do the I/O
        LDY #0             ;Store the byte in
        STA (_T),Y        ; 'Byte'
        PLA                ;Restore X register
        TAX

```

```

#
End; (* FileGetChar *)

```

```

Procedure FilePut(IOCbnum : Integer;
                  Var Rec : RecType;
                  RecLen : Integer);

```

```

Begin
#a

```

```

; _T    = LSB of RecLen
; _T+1 = MSB of RecLen
; _T+2 = LSB of ^Rec
; _T+3 = MSB of ^Rec
; _T+4 = LSB of IOCBnum
; _T+5 = MSB of IOCBnum

        TXA                ;Save X register
        PHA
        LDX #5             ;Copy heap
        LDY #10
CL5     LDA (_SP),Y
        STA _T,X
        DEY
        DEX

```

```

BPL CL5
LDA _T+4      ;Get IOCB #
ASL A        ;Multiply by 16
ASL A
ASL A
ASL A
TAX          ;Move to X register
LDA #11      ;Put record command
STA ICCMD,X
LDA _T+2      ;Data address
STA ICBAL,X
LDA _T+3
STA ICBAH,X
LDA _T        ;Data length
STA ICBLL,X
LDA _T+1
STA ICBLH,X
JSR CIOV     ;Do the I/O
PLA          ;Restore X register
TAX

```

```

#
End; (* FilePut *)

```

```

Procedure FilePutChar(IOCbnum : Integer;
                     Byte : Char);

```

```

Begin

```

```

#a

```

```

; _T    = Byte
; _T+1  = LSB of IOCBnum
; _T+2  = MSB of IOCBnum

```

```

TXA          ; Save X register
PHA
LDX #3       ; Copy heap
LDY #8

```

```

CL6 LDA (_SP),Y
STA _T,X
DEY
DEX
BPL CL6
LDA _T+1     ;Get IOCB #
ASL A       ;Multiply by 16
ASL A
ASL A
ASL A
TAX         ;Move to X register
LDA #11     ;Put Record command
STA ICCMD,X
LDA #0      ;Single byte put
STA ICBLL,X
STA ICBLH,X
LDA _T      ;Put Byte in ACC
JSR CIOV    ;Do the I/O
PLA        ;Restore X register

```

TAX

#

End; (\* FilePutChar \*)

Procedure FileNote(IOCbnum : Integer;  
                  Var Sector,Byte : Integer);

Begin

#a

; \_T = LSB of ^Byte  
; \_T+1 = MSB of ^Byte  
; \_T+2 = LSB of ^Sector  
; \_T+3 = MSB of ^Sector  
; \_T+4 = LSB of IOCbnum  
; \_T+5 = MSB of IOCbnum (Unused)

TXA                   ;Save X register

PHA

LDX #5               ;Copy heap

LDY #10

CL7 LDA (\_SP),Y

STA \_T,X

DEY

DEX

BPL CL7

LDA \_T+4             ;Get IOCb #

ASL A               ;Multiply by 16

ASL A

ASL A

ASL A

PHA                 ;Save for later

TAX                 ;Move to X register

LDA #38             ;Note command

STA ICCMD,X

JSR CIOV            ;Do the I/O

PLA                 ;Get IOCb # X 16

TAX                 ;Move to X register

LDY #0

LDA ICAX5,X         ;Get LSB of Byte

STA (\_T),Y

LDA ICAX3,X         ;Get LSB of Sector

STA (\_T+2),Y

INY

LDA #0              ;MSB of Byte

STA (\_T),Y

LDA ICAX4,X         ;Get MSB of Sector

STA (\_T+2),Y

PLA                 ;Restore X register

TAX

#

End; (\* FileNote \*)

Procedure FilePoint(IOCbnum,Sector,Byte : Integer);

Begin

#a

```
; _T    = LSB of Byte
; _T+1  = MSB of Byte (Unused)
; _T+2  = LSB of Sector
; _T+3  = MSB of Sector
; _T+4  = LSB of IOCBnum
; _T+5  = MSB of IOCBnum (Unused)
```

```
TXA          ;Save X register
PHA
LDX #5       ;Copy heap
LDY #10
```

```
CL8 LDA (_SP),Y
STA _T,X
DEY
DEX
BPL CL8
LDA _T+4     ;Get IOCB #
ASL A       ;Multiply by 16
ASL A
ASL A
ASL A
TAX         ;Move to X register
LDA #37     ;Point command
STA ICCMD,X
LDA _T+2     ;LSB of Sector
STA ICAX3,X
LDA _T+3     ;MSB of Sector
STA ICAX4,X
LDA _T       ;LSB of Byte
STA ICAX5,X
JSR CIOV    ;Do the I/O
PLA        ;Restore X register    TAX
```

#

End; (\* FilePoint \*)