

```

TITLE 'CPM3 MODULE FOR Seagate 6531 Hard disk Drive or CF Card.'

; Assume a BANKED version of CPM3
;
; 12/24/09      V1.1   Corrected the High/Low byte sector R/W error
; 03/15/2011   V1.2   Single Reset pulse to IDE Board

; DEFINE LOGICAL VALUES:
TRUE      EQU      -1
FALSE     EQU      NOT TRUE
;
BELL      EQU      07H
CR        EQU      0DH
LF        EQU      0AH

;Ports for 8255 chip. Change these to specify where the 8255 is addressed,
;and which of the 8255's ports are connected to which IDE signals.
;The first three control which 8255 ports have the control signals,
;upper and lower data bytes. The last one is for mode setting for the
;8255 to configure its ports, which must correspond to the way that
;the first three lines define which ports are connected.

IDEportA   EQU      030H      ;lower 8 bits of IDE interface
IDEportB   EQU      031H      ;upper 8 bits of IDE interface
IDEportC   EQU      032H      ;control lines for IDE interface
IDEportCtrl EQU      033H      ;8255 configuration port

READcfg8255 EQU      10010010b ;Set 8255 IDEportC out, IDEportA/B input
WRITEcfg8255 EQU      10000000b ;Set all three 8255 ports output

;IDE control lines for use with IDEportC. Change these 8
;constants to reflect where each signal of the 8255 each of the
;IDE control signals is connected. All the control signals must
;be on the same port, but these 8 lines let you connect them to
;whichever pins on that port.

IDEa0line  EQU      01H      ;direct from 8255 to IDE interface
IDEa1line  EQU      02H      ;direct from 8255 to IDE interface
IDEa2line  EQU      04H      ;direct from 8255 to IDE interface
IDEcs0line EQU      08H      ;inverter between 8255 and IDE interface
IDEcs1line EQU      10H      ;inverter between 8255 and IDE interface
IDEwrline  EQU      20H      ;inverter between 8255 and IDE interface
IDErdline  EQU      40H      ;inverter between 8255 and IDE interface
IDERstline EQU      80H      ;inverter between 8255 and IDE interface
;
;Symbolic constants for the IDE Drive registers, which makes the
;code more readable than always specifying the address pins

REGdata    EQU      IDEcs0line
REGgerr    EQU      IDEcs0line + IDEa0line
REGsecCnt  EQU      IDEcs0line + IDEa1line
REGsector  EQU      IDEcs0line + IDEa1line + IDEa0line
REGcylinderLSB EQU      IDEcs0line + IDEa2line
REGcylinderMSB EQU      IDEcs0line + IDEa2line + IDEa0line
REGshd     EQU      IDEcs0line + IDEa2line + IDEa1line      ; (0EH)
REGcommand EQU      IDEcs0line + IDEa2line + IDEa1line + IDEa0line ; (0FH)
REGstatus  EQU      IDEcs0line + IDEa2line + IDEa1line + IDEa0line
REGcontrol EQU      IDEcs1line + IDEa2line + IDEa1line
REGastatus EQU      IDEcs1line + IDEa2line + IDEa1line + IDEa0line

;IDE Command Constants. These should never change.

COMMANDrecal EQU      10H
COMMANDread  EQU      20H
COMMANDwrite EQU      30H
COMMANDinit  EQU      91H
COMMANDid    EQU      0ECH
COMMANDspindown EQU      0E0H
COMMANDspinup EQU      0E1H
;
MAXSEC      EQU      3DH      ;Sectors per track for CF my Memory drive, Kingston CF 8G. (CPM format, 0-3D)
;
; Note this will also work with a Seagate 6531 IDE drive
;
; IDE Status Register:
; bit 7: Busy 1=busy, 0=not busy
; bit 6: Ready 1=ready for command, 0=not ready yet
; bit 5: DF 1=fault occurred insIDE drive
; bit 4: DSC 1=seek complete

```

```

; bit 3: DRQ 1=data request ready, 0=not ready to xfer yet
; bit 2: CORR 1=correctable error occurred
; bit 1: IDX vendor specific
; bit 0: ERR 1=error occurred
;
; DEFINE PUBLIC LABELS:
PUBLIC DPH0 ;DISK PARAMETER HEADERS

; DEFINE EXTERNAL LABELS:
EXTRN @ADRV,@RDRV
EXTRN @DMA,@TRK,@SECT
EXTRN @CBNK
EXTRN @DBNK ;BANK FOR DMA OPERATION
EXTRN @ERMDE ;BDOS ERROR MODE
EXTRN ?WBOOT ;WARM BOOT VECTOR
EXTRN ?PMSG ;PRINT MESSAGE @<HL> UP TO 00, SAVES
; [BC] AND [DE]
EXTRN ?PDERR ;PRINT BIOS DISK ERROR HEADER
EXTRN ?CONIN,?CONO ;CONSOLE IN AND OUT
EXTRN ?CONST ;CONSOLE STATUS
EXTRN ?BNKSL ;SELECT PROCESSOR MEMORY BANK
EXTRN ?SMSG ;MY ROUTINE TO SPEAK A MESSAGE

; INCLUDE CP/M 3.0 DISK DEFINITION MACROS:
MACLIB CPM3

; INCLUDE Z-80 MACRO LIBRARY:
MACLIB Z80

DSEG ;PUT IN OP SYS BANK IF BANKING

; EXTENDED DISK PARAMETER HEADER FOR DRIVE 0:
DW HDWRT ;HARD DISK WRITE ROUTINE
DW HDRD ;HARD DISK READ ROUTINE
DW HDLOGIN ;HARD DISK LOGIN PROCEDURE
DW HDINIT ;HARD DISK DRIVE INITIALIZATION ROUTINE
DB 0 ;RELATIVE DRIVE 0 ON THIS CONTROLLER
DB 0 ;MEDIA TYPE:
; HI BIT SET : DRIVE NEEDS RECALIBRATING
DPH0: DPH 0,IDEHD$DPB,0,

; MAKE SURE DPB'S ARE IN COMMON MEMORY:
CSEG

; 512 BYTE SECTORS ON HARD DISK (512X61X256) = 7,995,392 Bytes)
; with 16 heads this is less than 16 drive cylinders!

IDEHD$DPB:
; DPB 512,32,512,2048,1024,2,8000H
; DPB 256,32,1001,2048,1024,2,8000H
; DPB 512,60,256,2048,1024,1,8000H
; DPB 512,62,252,2048,1024,1,8000H
DPB 512,61,256,2048,1024,1,8000H

DSEG ;CAN SET BACK TO BANKED SEGMENT IF BANKING

;;;; HDINIT:
HDINIT RET ;DO NOT INITILIZE HARD DISK YET
;

;;;; HDLOGIN
;-----INITILIZE THE IDE HARD DISK -----
HDLOGIN: ;Initilze the 8255 and drive then do a hard reset on the drive,
MVI A,READcfg8255 ;Config 8255 chip (10010010B), read mode on return
OUT IDEportCtrl ;Config 8255 chip, READ mode

;Hard reset the disk drive
;For some reason some CF cards need to the RESET line
;pulsed very carefully. You may need to play around
;with the pulse length. Symptoms are: incorrect data comming
MVI A,IDERstline
OUT IDEportC ;back from a sector read (often due to the wrong sector being read)
;I have a (negative)pulse of 2.7uSec. (10Mz Z80, two IO wait states).
MVI B,20H ;Which seem to work for the 5 different CF cards I have.
ResetDelay:
DCR B

```

```

JNZ     ResetDelay           ;Delay (reset pulse width)

XRA     A
OUT     IDEportC             ;No IDE control lines asserted (just bit 7 of port C)
CALL    DELAY$32

MVI     D,11100000b         ;Data for IDE SDH reg (512bytes, LBA mode,single drive,head 0000)
                                ;For Trk,Sec,head (non LBA) use 10100000
                                ;Note. Cannot get LBA mode to work with an old Seagate Medalist 6531 drive.
                                ;have to use teh non-LBA mode. (Common for old hard disks).

MVI     E,REGshd            ;00001110,(0EH) for CS0,A2,A1,
CALL    IDEwr8D              ;Write byte to select the MASTER device
CALL    IDEwr8D              ;Write byte to select the MASTER device
;
MVI     B,080H              ;<<< May need to adjust delay time
WaitInit:
MVI     E,REGstatus         ;Get status after initilization
CALL    IDErd8D              ;Check Status (info in [D])
BIT     7,D
JZ      DoneInit            ;Return if ready bit is zero
;
                                ;Delay to allow drive to get up to speed
PUSH    B
LXI     B,0FFFFH
DELAY2: MVI     D,2           ;May need to adjust delay time to allow cold drive to
DELAY1: DCR     D             ;to speed
JNZ     DELAY1
DCX     B
MOV     A,C
ORA     B
JNZ     DELAY2
POP     B
;
DJNZ   WaitInit
CALL    SetErrorFlag        ;Ret with NZ flag set if error (probably no drive)
LXI     H,MSG$INIT$ERR     ;RESTORE FAILED
CALL    ?PMSG
ORI     1
RET

DoneInit:
LXI     H,SAYSIGNON
CALL    ?SMSG               ;SPEAK MESSAGE
XRA     A                   ;RETURN WITH NO ERROR
RET
;
;
;----- SECTOR WRITE ROUTINE -----
; ROUTINE WRITES 1 SECTOR TO THE DISK:
;
HDWRT: XRA     A
STA     ERFLG
CALL    wrlba               ;Send to drive the sector we want to read. Converting
                                ;CPM TRK/SEC info to Drive TRK/SEC/Head
                                ;send before error check so info is updated
CALL    IDEwaitnotbusy     ;make sure drive is ready
JC      SetErrorFlag

MVI     D,COMMANDwrite
MVI     E,REGcommand
CALL    IDEwr8D            ;tell drive to write a sector
CALL    IDEwaitdrq         ;wait unit it wants the data
JC      SetErrorFlag       ;If problem abort
;
LHLD   @DMA                ;DMA address
JMP    ADJBNNN

CSEG
=====
ADJBNNN: LDA    @CBNK
PUSH   PSW
LDA    @DBNK
CALL   ?BNKSL

MVI    A,WRITEcfg8255
OUT    IDEportCtrl
MVI    B,0                ;256X2 bytes

WRSEC1: MOV    A,M
INX    H

```

```

OUT    IDEportA      ;LOW byte on A (Was wrong order in earlier versions)
MOV    A,M
INX    H
OUT    IDEportB      ;Then HIGH byte on B
MVI    A,REGdata
PUSH   PSW
OUT    IDEportC      ;Send write command

ORI    IDEwrline     ;Send WR pulse
OUT    IDEportC
POP    PSW
OUT    IDEportC
DJNZ   WRSEC1

MVI    A,READcfg8255 ;Set 8255 back to read mode
OUT    IDEportCtrl

POP    PSW
CALL   ?BNKSL
JMP    CHECK$RW

;=====
DSEG

CHECK$RW:
MVI    E,REGstatus   ;Check R/W status when done
CALL   IDErd8D
MOV    A,D
ANI    01H
STA    ERFLG         ;Ret Z if All OK
RZ

SetErrorFlag:
;For now just return with error flag set
XRA    A
DCR    A
STA    ERFLG         ;Ret NZ if problem
RET

;
;
;----- SECTOR READ ROUTINE -----
; ROUTINE READS 1 PHYSICAL SECTOR FROM THE DRIVE:
;
HDRD:  XRA    A
STA    ERFLG         ;CLEAR THE ERROR FLAG
CALL   wrlba         ;Send to drive the sector we want to read. Converting
;CPM TRK/SEC info to Drive TRK/SEC/Head
;Send before error check so info is updated
CALL   IDEwaitnotbusy ;make sure drive is ready
JC     SetErrorFlag  ;Returned with NZ set if error

MVI    D,COMMANDread
MVI    E,REGcommand
CALL   IDEwr8D       ;Send sec read command to drive.
CALL   IDEwaitdrq    ;Wait until it's got the data
JC     SetErrorFlag  ;If problem abort

;
LHLD   @DMA          ;DMA address
JMP    ADJBNKS

CSEG                ;MUST HAVE THE FOLLOWING CODE IN COMMON
;=====
ADJBNKS:LDA    @CBNK
PUSH   PSW
LDA    @DBNK        ;MUST HAVE THIS CODE IN COMMON
CALL   ?BNKSL       ;NOW DMA ADDRESS IS AT THE CORRECT BANK

MVI    B,0
MoreRD16:
MVI    A,REGdata     ;REG regsiter address
OUT    IDEportC

ORI    IDErdline     ;08H+40H, Pulse RD line
OUT    IDEportC

IN     IDEportA      ;read the LOW byte
MOV    M,A
INX    H
IN     IDEportB      ;THEN read the HIGH byte
MOV    M,A
INX    H

```

```

MVI    A,REGdata      ;Deassert RD line
OUT    IDEportC

DJNZ   MoreRD16

POP    PSW
CALL   ?BNKSL
JMP    CHECK$RW
;=====
DSEG
;
; ----- SUPPORT ROUTINES -----
;
wrlba:                                ;Write the logical block address to the drive's registers
LDA    @SECT            ;Get CPM requested sector
INR    A                ;Disk sectors 1...MAXSEC
MOV    D,A
MVI    E,REGsector     ;Send info to drive
CALL   IDEwr8D

LHLD   @TRK            ;We have are in LBA mode so send directly
MOV    D,L              ;Send Low TRK#
MVI    E,REGcylinderLSB
CALL   IDEwr8D

MOV    D,H              ;Send High TRK#
MVI    E,REGcylinderMSB
CALL   IDEwr8D

MVI    D,1              ;For now, one sector at a time
MVI    E,REGsecCnt
CALL   IDEwr8D
RET

IDEwaitnotbusy:                    ;Drive READY if 01000000
MVI    B,0FFH
MVI    C,0FFH          ;Delay, must be above 80H for 4MHz Z80. Leave longer for slower drives
MoreWait:
MVI    E,REGstatus     ;wait for RDY bit to be set
CALL   IDErd8D
MOV    A,D
ANI    11000000B
XRI    01000000B
JZ     DoneNotbusy
DJNZ   MoreWait
DCR    C
JNZ    MoreWait
STC
ret
DoneNotBusy:
ORA    A                ;Clear carry it indicate no error
RET

;Wait for the drive to be ready to transfer data.
;Returns the drive's status in Acc

IDEwaitdrq:
MVI    B,0FFH
MVI    C,0FFH          ;Delay, must be above 80H for 4MHz Z80. Leave longer for slower drives
MoreDRQ:
MVI    E,REGstatus     ;wait for DRQ bit to be set
CALL   IDErd8D
MOV    A,D
ANI    10001000B
CPI    00001000B
JZ     DoneDRQ
DJNZ   MoreDRQ
DCR    C
JNZ    MoreDRQ
STC
RET
DoneDRQ:
ORA    A                ;Clear carry
RET

DELAY$32: MVI    A,40          ;DELAY ~32 MS (DOES NOT SEEM TO BE CRITICAL)
DELAY3: MVI     B,0

```

```

M0:   DJNZ   M0
      DCR   A
      JNZ   DELAY3
      RET

```

```

;-----
; Low Level 8 bit R/W to the drive controller.  These are the routines that talk
; directly to the drive controller registers, via the 8255 chip.
; Note the 16 bit I/O to the drive (which is only for SEC R/W) is done directly
; in the routines HDRD & HDWRT for speed reasons.
;
IDEr8D:                                ;READ 8 bits from IDE register in [E], return info in [D]
      MOV   A,E
      OUT   IDEportC                    ;drive address onto control lines

      ORI   IDErdline                    ;RD pulse pin (40H)
      OUT   IDEportC                    ;assert read pin

      IN    IDEportA
      MOV   D,A                          ;return with data in [D]

      XRA   A
      OUT   IDEportC                    ;Zero all port C lines
      ret

;
;
IDEwr8D:                                ;WRITE Data in [D] to IDE register in [E]
      MVI   A,WRITEcfg8255              ;Set 8255 to write mode
      OUT   IDEportCtrl

      MOV   A,D
      OUT   IDEportA                    ;Get data put it in 8255 A port

      MOV   A,E
      OUT   IDEportC                    ;select IDE register

      ORI   IDEwrline                    ;lower WR line
      OUT   IDEportC
      NOP

      XRA   A
      OUT   IDEportC                    ;Deselect all lines including WR line

      MVI   A,READcfg8255                ;Config 8255 chip, read mode on return
      OUT   IDEportCtrl
      RET

```

```

;-----
MSG$INIT$ERR DB      'Initilization of IDE drive failed.',CR,LF,0
SAYSIGNON:   DB      '8 MEGA BYTE I D E HARD DISK ON DRIVE C',0
;
ERFLG:       DB      0H

```