

```

TITLE 'BOOT LOADER MODULE FOR CP/M 3.0'

; DEFINE LOGICAL VALUES:
TRUE      EQU    -1
FALSE     EQU    NOT TRUE

; DETERMINE IF FOR BANK SELECT OR NOT:
BANKED    EQU    TRUE ;FULL BLOWN VERSION

; DEFINE PUBLIC LABELS:
PUBLIC    ?INIT,?LDCCP,?RLCCP,?TIME
PUBLIC    OUT$BLOCKS
PUBLIC    ?MSG           ;NOTE A PUBLIC ROUTINE I ADDED FOR
                        ;SENDING SPEECH STRINGS TO TALKER

; EXTERNALLY DEFINED ENTRY POINTS AND LABELS:
EXTRN    ?PMSG,?CONIN
EXTRN    @CIVEC,@COVEC,@AIVEC,@AOVEC,@LOVEC
EXTRN    @CBNK,?BNKSL

IF      BANKED
EXTRN    BANKBUF           ;128 BYTE BUFFER IN MOVE MODULE FOR USE
                        ; DURING COLD AND WARM BOOTS
ENDIF

EXTRN    @SEC,@MIN,@HOUR,@DATE ;FIELDS HOLDING CURRENT TIME AND DATE

; INCLUDE Z-80 MACROS:
MACLIB    Z80

; SOME MISCELLANEOUS EQUATES:
BDOS      EQU    5
CR        EQU    13 ;ASCII CARRIAGE RETURN
LF        EQU    10 ;ASCII LINEFEED
;
PORTA     EQU    0aCH ;PORT FOR CLOCK DATA LINES
PORTB     EQU    0aDH ;PORT FOR CLOCK CONTROL AND ADD LINES
STATCLK   EQU    0aFH ;PORT FOR 8255 THAT CONTROLS CLOCK
;
; I/O FOR SD SYSTEMS I/O-8 BOARD PORT ASSIGNMENTS
;
BCTL      EQU    010H ; CHANNEL B CONTROL
ACTL      EQU    011H ; CHANNEL A CONTROL
BDTA      EQU    012H ; CHANNEL B DATA
ADTA      EQU    013H ; CHANNEL A DATA
SCCSEL    EQU    014H ; SCC SELECT
RTCRS     EQU    015H ; SD IO8 Clock status port (output here first to
select clock register)
RTC       EQU    016H ; Clock data port
SDS$STAT  EQU    017H ; SD IO8 Board status port
;
;
; WE CAN DO INITIALIZATION FROM BANKED MEMORY (IF WE HAVE IT):
IF      BANKED

```

```

DSEG ; INIT DONE FROM BANKED MEMORY
ELSE
CSEG ; INIT TO BE DONE FROM COMMON MEMORY
ENDIF

;;;; ?INIT
; HARDWARE INITIALIZATION OTHER THAN CHARACTER AND DISK I/O:
?INIT:
; ASSIGN CONSOLE INPUT AND OUTPUT TO CRT:
LXI H,8000H ;SIGNIFIES DEVICE 0
SHLD @CIVEC ;CONSOLE INPUT VECTOR
SHLD @COVEC ;CONSOLE OUTPUT VECTOR

; ASSIGN PRINTER TO LPT:
LXI H,4000H ;SIGNIFIES DEVICE 1
SHLD @LOVEC ;LIST OUTPUT VECTOR

; ASSIGN AUX TO CRT1:
LXI H,02000H ;SIGNIFIES DEVICE 2
SHLD @AIVEC ;AUXILLIARY INPUT VECTOR
SHLD @AOVEC ;AUXILLIARY OUTPUT VECTOR

; PRINT THE SIGN-ON MESSAGE:
LXI H,SIGNON$MSG ;POINT TO IT
JMP ?PMSG ;AND PRINT IT
;

;;;; ?SMSG
; ROUTINE OUTPUTS SPEECH STRING TO TALKER
IF BANKED
CSEG ;TO BE ON THE SAFE SIDE
ENDIF

?SMSG: MOV A,M
ORA A
JRZ SMSG1
MOV C,A
CALL SPEAKOUT
INX H
JR ?SMSG

SMSG1: MVI C,0DH ;MUST END STRING '0', SO END WITH CR FOR TALKER
; ;FALL THROUGH TO SPEAKOUT
;

SPEAKOUT: ;Sent output to Speach synthizer via Serial port
MVI A,3H ;Out to Serial Communications Controller #3 on SD
System Board
OUT SCCSEL ;Select SSC
SOUT1: IN BCTL ;Are we ready for a character
ANI 04H
JZ SOUT1 ;Note will not hang, will be 1 if no chip there.
MOV A,C
OUT BDTA ;Send it
RET
;
;

;;;;; OUT$BLOCKS
; ROUTINE OUTPUTS SPECIFIED # BYTES TO SPECIFIED OUTPUT PORTS:

```

```

        IF      BANKED
        CSEG                      ;WE WANT THIS ROUTINE IN COMMON MEMORY
        ENDIF
OUT$BLOCKS:
        MOV     A,M                ;GET A BYTE FROM THE BLOCK
        ORA     A                  ;END OF OUTPUT BLOCK ?
        RZ                      ;THEN DONE!!
        MOV     B,A                ;ELSE PUT # BYTES TO SEND OUT IN [B]
        INX     H                  ;POINT TO PORT TO SEND TO
        MOV     C,M                ;GET IT TO [C]
        INX     H                  ;POINT TO 1ST BYTE OF BLOCK TO SEND OUT
        OUTIR                      ;Z-80 BLOCK OUTPUT
        JR      OUT$BLOCKS

        ;;;; ?LDCCP
        ; THIS ROUTINE IS ENTERED TO LOAD THE CCP.COM FILE INTO THE TPA BANK
        ; AT SYSTEM COLD START:
?LDCCP:
        ; SET UP THE FCB FOR THE FILE OPERATION:
        XRA     A                  ;ZERO EXTENT
        STA     CCP$FCB+15
        LXI     H,0                ;START AT BEGINNING OF FILE
        SHLD   FCB$NR

        ; TRY TO OPEN THE CCP.COM FILE:
        LXI     D,CCP$FCB         ;POINT TO FCB
        CALL    OPEN              ;ATTEMPT THE OPEN OPERATION
        INR     A                  ;WAS IT ON THE DISK ?
        JRNZ    CCP$FOUND         ;YES -- GO LOAD IT

        ; WE ARRIVE HERE WHEN CCP.COM FILE WASN'T FOUND:
        LXI     H,CCP$MSG         ;REPORT THE ERROR
        CALL    ?PMSG
        CALL    ?CONIN            ;GET A RESPONSE
        JR      ?LDCCP           ;AND TRY AGAIN

        ; FILE WAS OPENED OK -- READ IT IN:
CCP$FOUND:
        LXI     D,0100H          ;LOAD AT BOTTOM OF TPA
        CALL    SETDMA           ;BY SETTING THE NEXT DMA ADDRESS
        LXI     D,128            ;SET MULTI SECTOR I/O COUNT
        CALL    SETMULTI        ; TO ALLOW UP TO 16K BYTES IN ONE OPERATION
        LXI     D,CCP$FCB       ;POINT TO THE FCB
        CALL    READ             ;AND READ THE CCP IN

        ; FOLLOWING CODE FOR BANKED SYSTEMS -- MOVES CCP IMAGE TO BANK 2
        ; FOR LATER RELOADING AT WARM STARTS:
        IF      BANKED
        LXI     H,0100H          ;GET CCP IMAGE FROM START OF TPA
        MVI     B,25             ;TRANSFER 25 LOGICAL SECTORS
        LDA     @CBNK           ;GET CURRENT BANK
        PUSH    PSW             ;AND SAVE IT
LD$1:
        PUSH    B                ;SAVE SECTOR COUNT
        MVI     A,1             ;SELECT TPA BANK
        CALL    ?BNKSL

```

```

LXI    B,128          ;TRANSFER 128 BYTES TO TEMPORARY BUFFER
LXI    D,BANKBUF     ;TEMPORARY BUFFER ADDR IN [DE]
PUSH   H              ;SAVE SOURCE ADDRESS
PUSH   D              ;AND DESTINATION
PUSH   B              ;AND COUNT
LDIR   ;BLOCK MOVE SECTOR TO TEMPORARY BUFFER
MVI    A,2           ;SELECT BANK TO SAVE CCP IN
CALL   ?BNKSL
POP    B              ;GET BACK COUNT
POP    H              ;LAST DESTINATION WILL BE NEW SOURCE ADDR
POP    D              ;LAST SOURCE WILL BE NEW DESTINATION
LDIR   ;BLOCK MOVE SECTOR FROM BUFFER TO ALTERNATE
        ; BANK
XCHG   ;NEXT ADDR WILL BE NEW SOURCE ADDR
POP    B              ;GET BACK SECTOR COUNT
DJNZ   LD$1          ;DROP SECTOR COUNT AND LOOP TILL DONE...
POP    PSW            ;WHEN DONE -- RESTORE ORIGINAL BANK
JMP    ?BNKSL
ELSE

; IF NON-BANKED WE RETURN THROUGH HERE:
RET
ENDIF

; ; ; ; ?RLCCP
; ROUTINE RELOADS CCP IMAGE FROM BANK 2 IF BANKED SYSTEM OR FROM THE
; DISK IF NON-BANKED VERSION:
?RLCCP:
IF     BANKED
; FOLLOWING CODE FOR BANKED VERSION:
LXI    H,0100H        ;GET CCP IMAGE FROM START OF ALTERNATE BUFFER
MVI    B,25           ;TRANSFER 25 LOGICAL SECTORS
LDA    @CBNK          ;GET CURRENT BANK
PUSH   PSW            ;AND SAVE IT
RL$1:
PUSH   B              ;SAVE SECTOR COUNT
MVI    A,2           ;SELECT ALTERNATE BANK
CALL   ?BNKSL
LXI    B,128          ;TRANSFER 128 BYTES TO TEMPORARY BUFFER
LXI    D,BANKBUF     ;TEMPORARY BUFFER ADDR IN [DE]
PUSH   H              ;SAVE SOURCE ADDRESS
PUSH   D              ;AND DESTINATION
PUSH   B              ;AND COUNT
LDIR   ;BLOCK MOVE SECTOR TO TEMPORARY BUFFER
MVI    A,1           ;PUT CCP TO TPA BANK
CALL   ?BNKSL
POP    B              ;GET BACK COUNT
POP    H              ;LAST DESTINATION WILL BE NEW SOURCE ADDR
POP    D              ;LAST SOURCE WILL BE NEW DESTINATION
LDIR   ;BLOCK MOVE SECTOR FROM BUFFER TO TPA BANK
XCHG   ;NEXT ADDR WILL BE NEW SOURCE ADDR
POP    B              ;GET BACK SECTOR COUNT
DJNZ   RL$1          ;DROP SECTOR COUNT AND LOOP TILL DONE...
POP    PSW            ;GET BACK LAST CURRENT BANK #
JMP    ?BNKSL        ;SELECT IT AND RETURN
ELSE

```

```

; FOLLOWING CODE IS FOR NON-BANKED VERSIONS:
JMP ?LDCCP          ;JUST DO LOAD AS THOUGH COLD BOOT
ENDIF

; ; ; ; ?TIME
; ROUTINE SETS/GETS TIME For SD Systems 8 I/O Board
; Their code as in manual directly
?TIME:
; SETTING OR GETTING TIME ?

PUSH H
PUSH D
MOV A,C
LXI H,TIMOUT
PUSH H
MVI C,RTC
MVI B,1          ;Count for port I/O
ORA A
JZ GETTM        ;Get Time
SETTM:
XRA A          ;Set Time
STA @SEC
MVI A,15H
OUT RTCRS
OUTP B

MVI A,03H
OUT RTCRS
LHLD @HOUR
OUTP H

MVI A,04H
OUT RTCRS
OUTP L

UPDT: MVI A,06H          ;SET D-O-M TO L
OUT RTCRS
OUTP B

MVI A,07H
OUT RTCRS
OUTP B

LHLD @DATE
MVI A,0BH
OUT RTCRS
OUTP L

MVI A,0CH
OUT RTCRS
OUTP H
RET

GETTM:
MVI A,06H          ;Get day from clock

```

```

OUT    RTCRS
INP    A

LXI    H,@DATE          ;Save it and clear A
MOV    M,A
XRA    A
RLD

ADD    A
MOV    E,A
ADD    A
ADD    A
ADD    E
MOV    E,A
XRA    A
RLD                ;X10

ADD    E
DCR    A
MOV    E,A
MVI    A,0BH
OUT    RTCRS
INP    L            ;read date stored msb

MVI    A,0CH
OUT    RTCRS
INP    H

MOV    A,E
ADD    L
MOV    L,A
MOV    A,H
ACI    0
MOV    H,A
MVI    A,07H
OUT    RTCRS        ;get month from clock
INP    A

PUSH   PSW
ANI    0F0H
XCHG
LXI    H,MONTBL
JRZ    DOMON

LXI    H,MONTBL+18
POP    PSW
INR    A
PUSH   PSW
DOMON:
POP    PSW
ANI    0FH
DMON1: DCR    A
JZ     STUFF
INX    H
INX    H
JR     DMON1
STUFF:

```

```

MOV    A,M
INX    H
MOV    H,M
MOV    L,A
DAD    D
SHLD   @DATE
CALL   UPDT

RERD:  MVI    A,04H ;Read hours
      OUT    RTCRS
      INP    L

      MVI    A,03H
      OUT    RTCRS
      INP    H

      SHLD   @HOUR

      MVI    A,02H ;read seconds
      OUT    RTCRS
      INP    D

      MVI    A,01H
      OUT    RTCRS
      IN     RTC
      ANI    0F0H
      JRZ   RERD

      MOV    A,D
      STA   @SEC
      RET

TIMOUT:
      POP    D
      POP    H
      RET

;
MONTBL:
      DW    0,31,59,90,120,151,181,212,243,273,304,334
;
;
      IF    BANKED
      CSEG
      ENDIF

      ; ; ; ;
      ; CP/M BDOS FUNCTION INTERFACES

      ; OPEN FILE:
OPEN:  MVI    C,15 ! JMP BDOS          ; OPEN FILE CONTROL BLOCK

      ; SET DMA ADDRESS:
SETDMA:
      MVI    C,26 ! JMP BDOS          ; SET DATA TRANSFER ADDRESS

      ; SET MULTI SECTOR I/O COUNT:
SETMULTI:

```

```

MVI C,44 ! JMP BDOS          ; SET RECORD COUNT

; READ FILE RECORD:
READ:
MVI C,20 ! JMP BDOS          ; READ RECORDS

; CCP NOT FOUND ERROR MESSAGE:
CCP$MSG:
DB    CR,LF,'BIOS ERR ON A: NO CCP.COM FILE',0

; FCB FOR CCP.COM FILE LOADING:
CCP$FCB:
DB    1          ;AUTO-SELECT DRIVE A
DB    'CCP      COM'      ;FILE NAME AND TYPE
DB    0,0,0,0
DS    16
FCB$NR:  DB    0,0,0

; SYSTEM SIGN-ON MESSAGE:
SIGNON$MSG:
DB    CR,LF,LF,'64K CP/M VERSION 3.0 (John Monahan 12/24/2009)'
DB    CR,LF,'A: = IDE CF Memory Disk, B: & C: = 8" Floppys, '
DB    CR,LF,'D: = 5" Floppy, E: = Mdisk, F: = ST506 Hard Disk.'
DB    CR,LF,'Note SD & DD CP/M 8" disk formats are supported.'
DB    CR,LF,0

END

```