WEST SIDE ELECTRONICS

SUPERCLOCK II

P.O. Box 636, Chatsworth, CA 91311

SUPERCLOCK II is a trademark of West Side Electronics.

Apple, Apple II, Applesoft, and Apple PASCAL are trademarks of Apple Computer, Inc.

*Apple Clock is a trademark of Mountain Hardware, Inc.

INSTALLATION

Before installing the SUPERCLOCK II into your computer, examine the board to become familiar with the function of the dipswitches.

SWITCH	ON	OFF
SET	Clock setting enabled	Setting disabled
ADJ	Resets seconds to 00	Normal position
MODE	Selects ACE mode	Normal mode

The normal position for all switches is down or OFF.

After turning off the computer, remove the cover by gently pulling up at the back. The SUPERCLOCK II can now be placed in any slot except 0. Do not replace the cover until the clock has been set.

To set the clock, the switch marked "SET" must be moved to the ON position. Now run the clock setting program supplied. Follow the directions in the program to set the clock to the correct local time. Note that all data to and from the clock is expressed numerically. In particular, the day of week is represented by a number from 0-6 as shown below:

0 - Sunday
1 - Monday
2 - Tuesday
3 - Wednesday
4 - Thursday
5 - Friday
6 - Saturday

After setting the clock, return the "SET" switch to the OFF position to prevent accidental changes to the clock.

READING THE CLOCK

The SUPERCLOCK II contains firmware that makes reading the clock from BASIC extremely easy. In short, whenever the SUPERCLOCK's slot is selected for input (ie. IN#n in BASIC, n control-K from the monitor, etc.), it will return a string of characters containing the date and time in this format (assuming NORMAL mode):

```
                W MM/DD/YY HH;MM;SS
(day of week ↑   \date/   \time/  )
```

Semicolons are used in the time display because Applesoft cannot handle colons on input. Note that when reading the clock, the computer will echo the input (ie. print it on the screen) just as if it were being entered from the keyboard. If you want to read the clock without having it echo to the screen, use the command PR#n. This sends all output to the SUPERCLOCK where it is "dumped." Don't forget to reset normal screen output with PR#0 after reading the clock.

When inputting the clock data into a string (e.g. IN#n INPUT A$), the various components of the time can be extracted using the appropriate string functions (see Integer BASIC and Applesoft manuals). Study the sample programs supplied for further clarification on how to use the SUPERCLOCK.

±30 SECOND ADJUSTMENT

Periodically, the SUPERCLOCK II can be synchronized to the correct time by using the switch marked "ADJ." Moving this switch to the ON position will momentarily reset the seconds to 00, adding one minute if the seconds were greater than 30. Return this switch to the OFF position after adjusting the clock.

APPLE CLOCK EMULATION (ACE) MODE

To take advantage of the many programs written for the Apple Clock* from Mountain Hardware, the SUPERCLOCK II has incorporated an Apple Clock Emulation mode. This mode is selected by moving the switch marked "MODE" to the ON position. The only difference between modes is in the format of the data presentation from the clock. The Apple Clock uses the following format:

```
  MM/DD HH;MM;SS.mmm
  \   /  \    /  \ /
( date   \time/  milliseconds )
```

In the ACE mode, the SUPERCLOCK II uses this format:

```
  MM/DD HH;MM;SS.WYY
  \   /  \    /  ↑ └year       )
( date   \time/   └day of week
```

Thus the only difference between the two clocks is that the day of week and year digits appear instead of milliseconds.

INTERRUPTS

Refer to the data sheet of the 6820/6821 for complete details in interrupt handling. The following interrupt frequencies are available:

CA1 -	1024 Hz	(approx. 1 per mS)
CA2 -	1 Hz	(1 per second)
CB1 -	1/60 Hz	(1 per minute)
CB2 -	1/3600 Hz	(1 per hour)

PIA addresses can be determined as follows:

PORT A -	$C080 + $n0	
CRA -	$C081 + $n0	
PORT B -	$C082 + $n0	(n=SUPERCLOCK slot)
CRB -	$C083 + $n0	

SUPERCLOCK II AND PASCAL

Using the SUPERCLOCK II with Apple PASCAL is greatly simplified by having the clock routines in the system library. The file WSE:SYSTEM.LIBRARY contains all of routines supplied in APPLE1:SYSTEM.LIBRARY plus the addition of UNIT SUPERCLOCK. You can give your programs access to the clock by simply adding the following statement just after the PROGRAM heading:

```
USES SUPERCLOCK;
```

This line effectively adds 3 external PROCEDURES which are described below.

```
PROCEDURE READCLOCK (a,b,c,d,e,f,g);
```

This procedure requires 7 variables of the type INTEGER. After calling READCLOCK, the variables will have the following values:

a - Day of Week	e - Hour
b - Month	f - Minutes
c - Day	g - Seconds
d - Year	

```
PROCEDURE TIMESTRING (b,c,d,e,f,g,h);
```

This procedure takes 6 variables of type INTEGER and formats them into a variable of type STRING. If the first 6 parameters have the associated values as described in READCLOCK, then the string variable (h) will be defined as

```
MM/DD/YY HH:MM:SS
```

```
PROCEDURE UPDATE;
```

This procedure first reads and displays the current time. Then it gets the time of last boot from the disk, displays a greeting message with this time, and finally updates both the disk and the PASCAL Filer.

Program AUTOBOOT uses this last procedure to automatically update a disk when booted. The code for AUTOBOOT is stored in the file WSE:SYSTEM.STARTUP. This program can be transferred to your other disks to give them this feature. Note that this program must be named SYSTEM.STARTUP for it to automatically run when booted. If you are already using a startup program, then a call to the procedure UPDATE can be added to the beginning of your existing program. Of course, for the time to be updated on the disk, it must not be write protected.

INTERNAL CONTROLS

There are six locations within the PIA accessible to the MPU data bus: two Peripheral Registers, two Data Direction Registers, and two Control Registers. Selection of these locations is controlled by the RS0 and RS1 inputs together with bit 2 in the Control Register, as shown in Table 1.

TABLE 1 – INTERNAL ADDRESSING

RS1	RS0	Control Register Bit CRA-2	Control Register Bit CRB-2	Location Selected
0	0	1	X	Peripheral Register A
0	0	0	X	Data Direction Register A
0	1	X	X	Control Register A
1	0	X	1	Peripheral Register B
1	0	X	0	Data Direction Register B
1	1	X	X	Control Register B

X = Don't Care

INITIALIZATION

A low reset line has the effect of zeroing all PIA registers. This will set PA0-PA7, PB0-PB7, CA2 and CB2 as inputs, and all interrupts disabled. The PIA must be configured during the restart program which follows the reset.

Details of possible configurations of the Data Direction and Control Register are as follows.

DATA DIRECTION REGISTERS (DDRA and DDRB)

The two Data Direction Registers allow the MPU to control the direction of data through each corresponding peripheral data line. A Data Direction Register bit set at "0" configures the corresponding peripheral data line as an input; a "1" results in an output.

CONTROL REGISTERS (CRA and CRB)

The two Control Registers (CRA and CRB) allow the MPU to control the operation of the four peripheral control lines CA1, CA2, CB1 and CB2. In addition they allow the MPU to enable the interrupt lines and monitor the status of the interrupt flags. Bits 0 through 5 of the two registers may be written or read by the MPU when the proper chip select and register select signals are applied. Bits 6 and 7 of the two registers are read only and are modified by external interrupts occurring on control lines CA1, CA2, CB1 or CB2. The format of the control words is shown in Table 2.

TABLE 2 – CONTROL WORD FORMAT

	7	6	5	4	3	2	1	0
CRA	IRQA1	IRQA2	CA2 Control			DDRA Access	CA1 Control	
CRB	IRQB1	IRQB2	CB2 Control			DDRB Access	CB1 Control	

Data Direction Access Control Bit (CRA-2 and CRB-2) – Bit 2 in each Control register (CRA and CRB) allows selection of either a Peripheral Interface Register or the Data Direction Register when the proper register select signals are applied to RS0 and RS1.

Interrupt Flags (CRA-6, CRA-7, CRB-6, and CRB-7) – The four interrupt flag bits are set by active transitions of signals on the four Interrupt and Peripheral Control lines when those lines are programmed to be inputs. These bits cannot be set directly from the MPU Data Bus and are reset indirectly by a Read Peripheral Data Operation on the appropriate section.

TABLE 3 – CONTROL OF INTERRUPT INPUTS CA1 AND CB1

CRA-1 (CRB-1)	CRA-0 (CRB-0)	Interrupt Input CA1 (CB1)	Interrupt Flag CRA-7 (CRB-7)	MPU Interrupt Request $\overline{IRQA}$ ($\overline{IRQB}$)
0	0	↓ Active	Set high on ↓ of CA1 (CB1)	Disabled — $\overline{IRQ}$ remains high
0	1	↓ Active	Set high on ↓ of CA1 (CB1)	Goes low when the interrupt flag bit CRA-7 (CRB-7) goes high
1	0	↑ Active	Set high on ↑ of CA1 (CB1)	Disabled — $\overline{IRQ}$ remains high
1	1	↑ Active	Set high on ↑ of CA1 (CB1)	Goes low when the interrupt flag bit CRA-7 (CRB-7) goes high

Notes:
1. ↑ indicates positive transition (low to high)
2. ↓ indicates negative transition (high to low)
3. The interrupt flag bit CRA-7 is cleared by an MPU Read of the A Data Register, and CRB-7 is cleared by an MPU Read of the B Data Register.
4. If CRA-0 (CRB-0) is low when an interrupt occurs (interrupt disabled) and is later brought high, $\overline{IRQA}$ ($\overline{IRQB}$) occurs after CRA-0 (CRB-0) is written to a "one".

Control of CA1 and CB1 Interrupt Input Lines (CRA-0, CRB-0, CRA-1, and CRB-1) – The two lowest order bits of the control registers are used to control the interrupt input lines CA1 and CB1. Bits CRA-0 and CRB-0 are used to enable the MPU interrupt signals $\overline{IRQA}$ and $\overline{IRQB}$, respectively. Bits CRA-1 and CRB-1 determine the active transition of the interrupt input signals CA1 and CB1 (Table 3).

TABLE 4 – CONTROL OF CA2 AND CB2 AS INTERRUPT INPUTS
CRA5 (CRB5) is low

CRA-5 (CRB-5)	CRA-4 (CRB-4)	CRA-3 (CRB-3)	Interrupt Input CA2 (CB2)	Interrupt Flag CRA-6 (CRB-6)	MPU Interrupt Request $\overline{IRQA}$ ($\overline{IRQB}$)
0	0	0	↓ Active	Set high on ↓ of CA2 (CB2)	Disabled — $\overline{IRQ}$ remains high
0	0	1	↓ Active	Set high on ↓ of CA2 (CB2)	Goes low when the interrupt flag bit CRA-6 (CRB-6) goes high
0	1	0	↑ Active	Set high on ↑ of CA2 (CB2)	Disabled — $\overline{IRQ}$ remains high
0	1	1	↑ Active	Set high on ↑ of CA2 (CB2)	Goes low when the interrupt flag bit CRA-6 (CRB-6) goes high

Notes: 1. ↑ indicates positive transition (low to high)
2. ↓ indicates negative transition (high to low)
3. The interrupt flag bit CRA-6 is cleared by an MPU Read of the A Data Register and CRB-6 is cleared by an MPU Read of the B Data Register.
4. If CRA-3 (CRB-3) is low when an interrupt occurs (Interrupt disabled) and is later brought high, $\overline{IRQA}$ ($\overline{IRQB}$) occurs after CRA-3 (CRB-3) is written to a "one".

TABLE 5 – CONTROL OF CB2 AS AN OUTPUT
CRB-5 is high

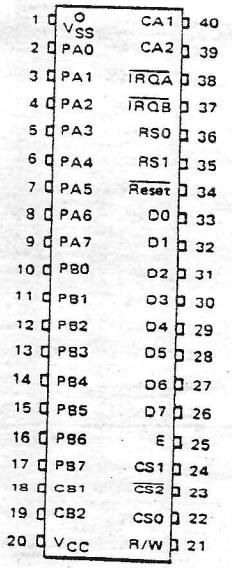
CRB-5	CRB-4	CRB-3	CB2 Cleared	CB2 Set
1	0	0	Low on the positive transition of the first E pulse following an MPU Write "B" Data Register operation.	High when the interrupt flag bit CRB-7 is set by an active transition of the CB1 signal.
1	0	1	Low on the positive transition of the first E pulse after an MPU Write "B" Data Register operation.	High on the positive edge of the first "E" pulse following an "E" pulse which occurred while the part was deselected.
1	1	0	Low when CRB-3 goes low as a result of an MPU Write in Control Register "B".	Always low as long as CRB-3 is low. Will go high on an MPU Write in Control Register "B" that changes CRB-3 to "one".
1	1	1	Always high as long as CRB-3 is high. Will be cleared when an MPU Write Control Register "B" results in clearing CRB-3 to "zero".	High when CRB-3 goes high as a result of an MPU Write into Control Register "B".

Control of CA2 and CB2 Peripheral Control Lines (CRA-3, CRA-4, CRA-5, CRB-3, CRB-4, and CRB-5) – Bits 3, 4, and 5 of the two control registers are used to control the CA2 and CB2 Peripheral Control lines. These bits determine if the control lines will be an interrupt input or an output control signal. If bit CRA-5 (CRB-5) is low, CA2 (CB2) is an interrupt input line similar to CA1 (CB1) (Table 4). When CRA-5 (CRB-5) is high, CA2 (CB2) becomes an output signal that may be used to control peripheral data transfers. When in the output mode, CA2 and CB2 have slightly different characteristics (Tables 5 and 6).

TABLE 6 – CONTROL OF CA-2 AS AN OUTPUT
CRA-5 is high

CRA-5	CRA-4	CRA-3	CA2 Cleared	CA2 Set
1	0	0	Low on negative transition of E after an MPU Read "A" Data operation.	High when the interrupt flag bit CRA-7 is set by an active transition of the CA1 signal.
1	0	1	Low on negative transition of E after an MPU Read "A" Data operation.	High on the negative edge of the first "E" pulse which occurs during a deselect.
1	1	0	Low when CRA-3 goes low as a result of an MPU Write to Control Register "A".	Always low as long as CRA-3 is low. Will go high on an MPU Write to Control Register "A" that changes CRA-3 to "one".
1	1	1	Always high as long as CRA-3 is high. Will be cleared on an MPU Write to Control Register "A" that clears CRA-3 to a "zero".	High when CRA-3 goes high as a result of an MPU Write to Control Register "A".

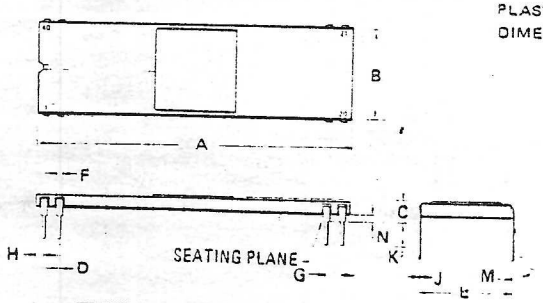
PIN ASSIGNMENT



PACKAGE DIMENSIONS

CASE 715-02
(CERAMIC)

SEE PAGE 165 FOR PLASTIC PACKAGE DIMENSIONS.



DIM	MILLIMETERS MIN	MILLIMETERS MAX	INCHES MIN	INCHES MAX
A	50.29	51.31	1.980	2.020
B	14.86	15.62	0.585	0.615
C	2.54	4.19	0.100	0.165
D	0.38	0.53	0.015	0.021
F	0.76	1.40	0.030	0.055
G	2.54 BSC		0.100 BSC	
H	0.76	1.78	0.030	0.070
J	0.20	0.33	0.008	0.013
K	2.54	4.19	0.100	0.165
L	14.60	15.37	0.575	0.605
M	–	10°	–	10°
N	0.51	1.52	0.020	0.060

NOTE:
1. LEADS, TRUE POSITIONED WITHIN 0.25 mm (0.010) DIA (AT SEATING PLANE), AT MAX. MAT'L CONDITION.

PROGRAM "SET CLOCK"

```
]LIST
   10 REM   ** SUPERCLOCK II SETTING PROGRAM **
   20 CALL -936
   30 GOSUB 5000
   40 PRINT "      YOUR SUPERCLOCK IS IN SLOT ";SLOT
   50 F24=1: REM  F24=0 FOR 12 HOUR
   60 PM=0: REM PM=1 FOR 12 HOUR & PM
   70 LY=0: REM  LY=1 FOR LEAP YEAR
   80 DIM A(14),B(14),A$(20)
   90 B(12)=0:B(13)=0
  100 A(1)=22:A(2)=26:A(3)=25:A(4)=24:A(5)=23
  110 A(6)=28:A(7)=27:A(8)=21:A(9)=20:A(10)=19
  120 A(11)=18:A(12)=17:A(13)=16
  130 PA=-16256+SLOT*16
  140 CRA=PA+1
  150 PB=CRA+1
  160 GOSUB 180: REM   READ CLOCK
  170 GOTO 240: REM   SET CLOCK
  180 PRINT " IN#";SLOT: REM   CTRL-D IN QUOTES FOR DOS
  190 POKE 50,63: VTAB 23: PRINT "       PRESS ANY KEY TO CONTINUE       "
      ;: POKE 50,255: VTAB 4: PRINT
  200 INPUT "THE PRESENT TIME IS ",A$
  210 PRINT " IN#0": REM   CTRL-D HERE TOO
  220 IF PEEK (-16384)<128 THEN 180: POKE -16368,0
  230 RETURN
  240 POKE 50,63: VTAB 23: PRINT "     MAKE SURE 'SET' SWITCH IS ON     "
      ;: POKE 50,255: VTAB 9
  250 INPUT "ENTER DAY OF WEEK (0-6) ",B(1): IF B(1)<0 OR B(1)>6 THEN 250

  260 INPUT "ENTER MONTH (1-12)      ",M: IF M<1 OR M>12 THEN 260:B(2)=M/
      10:B(3)=M MOD 10
  270 INPUT "ENTER DAY (1-31)        ",D: IF D<1 OR D>31 THEN 270:B(4)=D/
      10+4*LY:B(5)=D MOD 10
  280 INPUT "ENTER YEAR (0-99)       ",Y: IF Y<0 OR Y>99 THEN 280:B(6)=Y/
      10:B(7)=Y MOD 10
  290 IF NOT F24 THEN 310
  300 INPUT "ENTER HOUR (0-23)       ",H: IF H<0 OR H>23 THEN 300:B(8)=H/
      10+8:B(9)=H MOD 10: GOTO 320
  310 INPUT "ENTER HOUR (1-12)       ",H: IF H<1 OR H>12 THEN 310
  320 INPUT "ENTER MINUTE (0-59)     ",N: IF N<0 OR N>59 THEN 320:B(10)=N/
      10:B(11)=N MOD 10
  330 IF F24 THEN 350
  340 INPUT "ENTER AM/PM             ?",A$: IF A$(1,1)="P" THEN PM=1:B(8)
      =H/10+4*PM:B(9)=H MOD 10
  350 VTAB 23: POKE 50,63: PRINT "    PRESS (RETURN) TO SET CLOCK   ";: POKE
      50,255: INPUT A$
  360 POKE CRA,0
  370 POKE PA,255
  380 POKE CRA,4
  390 FOR I=1 TO 13
  400 POKE PB,A(I)
  410 POKE PA,B(I)
  420 POKE PB,A(I)+64
  430 POKE PB,A(I)
  440 NEXT I
  450 POKE PB,0
  460 POKE CRA,0
  470 POKE PA,240
  480 POKE CRA,4
  490 GOSUB 180
  500 VTAB 23: POKE 50,63: PRINT "    TURN 'SET' SWITCH TO OFF          "
      ;: POKE 50,255: END
 5000 REM  SUPERCLOCK II FINDER
 5010 I= PEEK (-12289): REM  KILL ALL ROMS
 5020 I=1
 5030 IF PEEK (-16350+I*256)=248 AND PEEK (-16349+I*256)=5 AND PEEK (-16348
      +I*256)=104 THEN 5060
 5040 I=I+1: IF I<8 THEN 5030
 5050 PRINT "I CANNOT FIND A SUPERCLOCK II": POP :I= PEEK (-12289): END
 5060 SLOT=I
 5070 I= PEEK (-12289): RETURN
```

PROGRAM "SET CLOCK - APPLESOFT"

```
]LIST

10  REM   ** SUPERCLOCK II SETTING PROGRAM (APPLESOFT) **
15  DEF  FN MOD(I) = I - 10 *  INT (I / 10)
20  HOME
30  GOSUB 5000
40  PRINT "      YOUR SUPERCLOCK IS IN SLOT ";SLOT
50 F24 = 1: REM    F24=0 FOR 12 HOUR
60 PM = 0: REM   PM=1 FOR 12 HOUR & PM
70 LY = 0: REM    LY=1 FOR LEAP YEAR
80  DIM A(14),B(14)
90 B(12) = 0:B(13) = 0
100 A(1) = 22:A(2) = 26:A(3) = 25:A(4) = 24:A(5) = 23
110 A(6) = 28:A(7) = 27:A(8) = 21:A(9) = 20:A(10) = 18
120 A(11) = 18:A(12) = 17:A(13) = 16
130 PA =  - 16256 + SLOT * 16
140 CA = PA + 1
150 PB = CA + 1
160  GOSUB 180: REM     READ CLOCK
170  GOTO 240: REM     SET CLOCK
180  PRINT " IN#";SLOT: REM  CTRL-D IN QUOTES FOR DOS
190  INVERSE : VTAB 23: PRINT "        PRESS ANY KEY TO CONTINUE          ";
     : NORMAL : VTAB 4: PRINT
200  INPUT "THE PRESENT TIME IS ";A$
210  PRINT " IN#0": REM  CTRL-D HERE TOO
220  IF  PEEK ( - 16384) < 128 THEN 180:
225  POKE  - 16368,0
230  RETURN
240  INVERSE : VTAB 23: PRINT "      MAKE SURE 'SET' SWITCH IS ON       ": NORMAL
     : VTAB 9
250  INPUT "ENTER DAY OF WEEK (0-6) ";B(1): IF B(1) < 0 OR B(1) > 6 THEN
     250
260  INPUT "ENTER MONTH (1-12)       ";M:B(2) =  INT (M / 10):B(3) =  FN M
     OD(M): IF M < 1 OR M > 12 THEN 260
270  INPUT "ENTER DAY (1-31)         ";D:B(4) =  INT (D / 10 + 4 * LY):B(5
     ) =  FN MOD(D): IF D < 1 OR D > 31 THEN 270
280  INPUT "ENTER YEAR (0-99)        ";Y:B(6) =  INT (Y / 10):B(7) =  FN M
     OD(Y): IF Y < 0 OR Y > 99 THEN 280
290  IF  NOT F24 THEN 310
300  INPUT "ENTER HOUR (0-23)        ";H:B(8) =  INT (H / 10 + 8):B(9) =  FN
     MOD(H): IF H < 0 OR H > 23 THEN 300
305  GOTO 320
310  INPUT "ENTER HOUR (1-12)        ";H: IF H < 1 OR H > 12 THEN 310
320  INPUT "ENTER MINUTE (0-59)      ";N:B(10) =  INT (N / 10):B(11) =  FN
     MOD(N): IF N < 0 OR N > 59 THEN 320
330  IF F24 THEN 350
340  INPUT "ENTER AM/PM              ";A$: IF  MID$ (A$,1,1) = "P" THEN PM
      = 1
345 B(3) =  INT (H / 10 + 4 * PM):B(9) =  FN MOD(H)
350  VTAB 23: INVERSE : PRINT "    PRESS <RETURN> TO SET CLOCK    ";: NORMAL
     : INPUT A$
360  POKE CA,0
370  POKE PA,255
380  POKE CA,4
390  FOR I = 1 TO 13
400  POKE PB,A(I)
410  POKE PA,B(I)
420  POKE PB,A(I) + 64
430  POKE PB,A(I)
440  NEXT I
450  POKE PB,0
460  POKE CA,0
470  POKE PA,240
480  POKE CA,4
490  GOSUB 130
500  VTAB 23: INVERSE : PRINT "    TURN 'SET' SWITCH TO OFF          ";
     : NORMAL : END
5000  REM   SUPERCLOCK II FINDER
5010 I =  PEEK ( - 12289): REM   KILL ALL ROMS
5020 I = 1
5030  IF  PEEK ( - 16350 + I * 256) = 248 AND  PEEK ( - 16349 + I * 256) =
     5 AND  PEEK ( - 16348 + I * 256) = 104 THEN 5060
5040 I = I + 1: IF I < 8 THEN 5030
5050  PRINT "I CANNOT FIND A SUPERCLOCK II": I =  PEEK ( - 12239): END
5060 SLOT = I
5070 I =  PEEK ( - 12289): RETURN
```

PROGRAM "TIMEFILER"

```
>LIST
   10 DIM A$(20)
   20 CALL -936
   30 GOSUB 220
   40 PRINT " OPEN TIMEFILE": REM CTRL-D INCLUDED IN ALL DOS COMMANDS
   50 PRINT " READ TIMEFILE"
   60 INPUT A$
   70 PRINT " CLOSE"
   80 PRINT "      SUPERCLOCK II DEMO PROGRAMS"
   90 VTAB 4: PRINT "THIS DISK WAS"
  100 PRINT "LAST BOOTED ON: ";A$
  110 PRINT " IN#";SLOT
  120 VTAB 7: INPUT "CURRENT TIME:    ",A$
  130 PRINT " OPEN TIMEFILE"
  140 PRINT " WRITE TIMEFILE"
  150 PRINT A$
  160 PRINT " CLOSE"
  170 PRINT "IN#0"
  180 VTAB 12: INPUT "C(ATALOG OR S(ET CLOCK ",A$
  190 IF A$(1,1)="S" THEN PRINT " RUN SET CLOCK"
  200 PRINT " CATALOG"
  210 END
 5000 REM   SUPERCLOCK II SLOT FINDER
 5010 I= PEEK (-12289): REM  KILL ALL ROMS
 5020 I=1
 5030 IF PEEK (-16350+I*256)=248 AND PEEK (-16349+I*256)=5 AND PEEK (-16348
      +I*256)=104 THEN 5060
 5040 I=I+1: IF I<8 THEN 5030
 5050 PRINT "I CANNOT FIND A SUPERCLOCK II": POP :I= PEEK (-12289): END
 5060 SLOT=I
 5070 I= PEEK (-12289): RETURN

>

>

>LOADTIMEFILER - APPLESOFT

]
]


]LIST

 20  HOME
 30  GOSUB 5000
 40  PRINT " OPEN TIMEFILE": REM  CTRL-D INCLUDED IN ALL DOS COMMANDS
 50  PRINT " READ TIMEFILE"
 60  INPUT A$
 70  PRINT " CLOSE"
 80  PRINT "      SUPERCLOCK II DEMO PROGRAMS"
 90  VTAB 4: PRINT "THIS DISK WAS"
100  PRINT "LAST BOOTED ON: ":A$
110  PRINT " IN#";SLOT
120  VTAB 7: INPUT "CURRENT TIME:  ";A$
130  PRINT " OPEN TIMEFILE"
140  PRINT " WRITE TIMEFILE"
150  PRINT A$
160  PRINT " CLOSE"
170  PRINT "IN#0"
180  VTAB 12: INPUT "C(ATALOG OR S(ET CLOCK ";A$
190  IF A$ = "S" THEN  PRINT " RUN SET CLOCK - APPLESOFT"
200  PRINT " CATALOG"
220  END
5000  REM   SUPERCLOCK II FINDER
5010 I =  PEEK ( - 12289): REM   KILL ALL ROMS
5020 I = 1
5030  IF  PEEK ( - 16350 + I * 256) = 248 AND  PEEK ( - 16349 + I * 256) =
     5 AND  PEEK ( - 16348 + I * 256) = 104 THEN 5060
5040 I = I + 1: IF I < 8 THEN 5030
5050  PRINT "I CANNOT FIND A SUPERCLOCK II":I =  PEEK ( - 12289): END
5060 SLOT = I
5070 I =  PEEK ( - 12289): RETURN
```

```
                0001  ;******************
                0002  ;*                *
                0003  ;* SUPERCLOCK II *
                0004  ;*  I/O  DRIVER  *
                0005  ;*                *
                0006  ;******************
                0007  ;
                0008  ; J. MAZUR 8/9/80
                0009  ;
                0010  ;
                0011  BASL    EQU 0028
                0012  STACK   EQU 0100
                0013  CSWL    EQU 0036
                0014  CSWH    EQU 0037
                0015  KSWL    EQU 0038
                0016  KSWH    EQU 0039
                0017  ASAVE   EQU 0478
                0018  PSAVE   EQU 04F8
                0019  YSAVE   EQU 0578
                0020  XSAVE   EQU 05F8
                0021  N0SAVE  EQU 0678
                0022  TBLPTR  EQU 06F8
                0023  SLOT    EQU 07F8
                0024  PORTA   EQU C080
                0025  CRA     EQU C081
                0026  PORTB   EQU C082
                0027  CRB     EQU C083
                0028  IORTS   EQU FF58
                0029  ;
                0030  ;
                0031          ORG C300
08              0032          PHP             ;SAVE P
78              0033          SEI             ;DISABLE INT
2C58FF          0034          BIT IORTS       ;SET V FLAG
5000            0035          BVC ENTR        ;ALWAYS TAKEN
                0036          ORG C306        ;BYTE SAVER
B8              0037          CLV             ;RE-ENTRY
48              0038  ENTR    PHA
8A              0039          TXA
48              0040          PHA
2058FF          0041          JSR IORTS
BA              0042          TSX
BD0001          0043          LDA STACK,X     ;$CN
D003            0044          BNE SKIP        ;SKIP OVER
D7D3C5          0045          DFD "WSE"
8DF807          0046  SKIP    STA SLOT
290F            0047          AND #$0F        ;$0N
AA              0048          TAX
98              0049          TYA             ;SAVE REG
9D7805          0050          STA YSAVE,X
68              0051          PLA
9DF805          0052          STA XSAVE,X
68              0053          PLA
9D7804          0054          STA ASAVE,X
5004            0055          BVC CONT
68              0056          PLA
9DF804          0057          STA PSAVE,X
8A              0058  CONT    TXA
0A              0059          ASL
0A              0060          ASL
0A              0061          ASL
0A              0062          ASL
9D7806          0063          STA N0SAVE,X
505A            0064          BVC READ
A536            0065          LDA CSWL        ;DETERMINE
D022            0066          BNE CONT1       ;IF ENTERED
ACF807          0067          LDY SLOT        ;AS A READ
C437            0068          CPY CSWH        ;OR WRITE
D013            0069          BNE CONT1
A0C9            0070  WRITE   LDY #$C9        ;RETN LOC
8436            0071          STY CSWL
A438            0072          LDY KSWL
D007            0073          BNE NOREAD
ACF807          0074          LDY SLOT
C439            0075          CPY KSWH
F00C            0076          BEQ CONT1
BDF804          0077  NOREAD  LDA PSAVE,X
48              0078          PHA
BD7804          0079          LDA ASAVE,X
2C58FF          0080          BIT IORTS
705F            0081          BVS EXIT        ;ALWAYS TAKEN
BC7806          0082  CONT1   LDY N0SAVE,X    ;INITIALIZE
A900            0083          LDA #$00        ;P'IA
9981C0          0084          STA CRA,Y
9983C0          0085          STA CRB,Y
```

```
C369  A9F0      0086          LDA #$F0
C36B  9980C0    0087          STA PORTA,Y
C36E  A9FF      0088          LDA #$FF
C370  9982C0    0089          STA PORTB,Y
C373  A904      0090          LDA #$04
C375  9981C0    0091          STA CRA,Y
C378  9983C0    0092          STA CRB,Y
C37B  A92F      0093          LDA #$2F
C37D  9982C0    0094          STA PORTB,Y
C380  A9E4      0095          LDA #$E4        ;SET COUNTER
C382  9DF806    0096          STA TBLPTR,X    ;TO TABLE-7
C385  A906      0097          LDA #$06        ;SET KSW FOR
C387  8538      0098          STA KSWL        ;RE-ENTRY
C389  BC7806    0099          LDY N0SAVE,X
C38C  A910      0100          LDA #$10        ;SET HOLD
C38E  9982C0    0101          STA PORTB,Y
C391  B8        0102          CLV
C392  BD7804    0103  READ    LDA ASAVE,X
C395  BC7805    0104          LDY YSAVE,X
C398  9128      0105          STA (BASL),Y
C39A  FEF806    0106          INC TBLPTR,X
C39D  BCF806    0107          LDY TBLPTR,X
C3A0  B138      0108          LDA (KSWL),Y
C3A2  1026      0109          BPL RDCLOK
C3A4  C9FF      0110          CMP #$FF        ;DONE?
C3A6  D015      0111          BNE EXIT        ;IF NOT, EXIT
C3A8  A900      0112          LDA #$00        ;RESTORE KSW
C3AA  8538      0113          STA KSWL
C3AC  BDF804    0114          LDA PSAVE,X     ;RECALL INIT.
C3AF  48        0115          PHA             ;P AND PUSH
C3B0  BC7806    0116          LDY N0SAVE,X
C3B3  A92F      0117          LDA #$2F        ;RESTORE INT-
C3B5  9982C0    0118          STA PORTB,Y
C3B8  A98D      0119          LDA #$8D        ;ASCII CR
C3BA  2C58FF    0120          BIT IORTS
C3BD  48        0121  EXIT    PHA
C3BE  BC7805    0122          LDY YSAVE,X     ;RESTORE REG
C3C1  BDF805    0123          LDA XSAVE,X
C3C4  AA        0124          TAX
C3C5  68        0125          PLA
C3C6  5001      0126          BVC RETN        ;RESTORE P ON
C3C8  28        0127          PLP             ;FINAL EXIT
C3C9  60        0128  RETN    RTS
C3CA  BC7806    0129  RDCLOK  LDY N0SAVE,X
C3CD  9982C0    0130          STA PORTB,Y
C3D0  EA        0131          NOP             ;WAIT A
C3D1  EA        0132          NOP             ;SHORT WHILE
C3D2  B980C0    0133          LDA PORTA,Y
C3D5  48        0134          PHA
C3D6  BDF806    0135          LDA TBLPTR,X
C3D9  C9EB      0136          CMP #$EB
C3DB  F009      0137          BEQ MASK        ;MASK D10 BIT
C3DD  C9F1      0138          CMP #$F1
C3DF  F005      0139          BEQ MASK        ;MASK H10 BIT
C3E1  68        0140          PLA
C3E2  09B0      0141  NOMASK  ORA #$B0        ;APPLE ASCII
C3E4  30D7      0142          BMI EXIT
C3E6  68        0143  MASK    PLA
C3E7  2903      0144          AND #$03        ;REMOVE
C3E9  10F7      0145          BPL NOMASK      ;STATUS BITS
                0146  ;
                0147  ;       TABLE
C3EB  A036A03A39AF3837AF3C
C3F5  3BA03534BB3332BB3130FF
```

ACE MODE CHANGES

```
C313  2CFF68    0045          DFD 2CFF68      ;ACE BYTES

C3D9  C9E9      0136          CMP #$E9
C3DB  F009      0137          BEQ MASK        ;MASK D10 BIT
C3DD  C9EC      0138          CMP #$EC
C3DF  F005      0139          BEQ MASK        ;MASK H10 BIT
C3E1  68        0140          PLA
C3E2  09B0      0141  NOMASK  ORA #$B0        ;APPLE ASCII
C3E4  30D7      0142          BMI EXIT
C3E6  68        0143  MASK    PLA
C3E7  2903      0144          AND #$03        ;REMOVE
C3E9  10F7      0145          BPL NOMASK      ;STATUS BITS
                0146  ;
                0147  ;       TABLE
C3EB  A03A39AF3837A03534BB
C3F5  3332BB3130AE363C3BFFFF
```