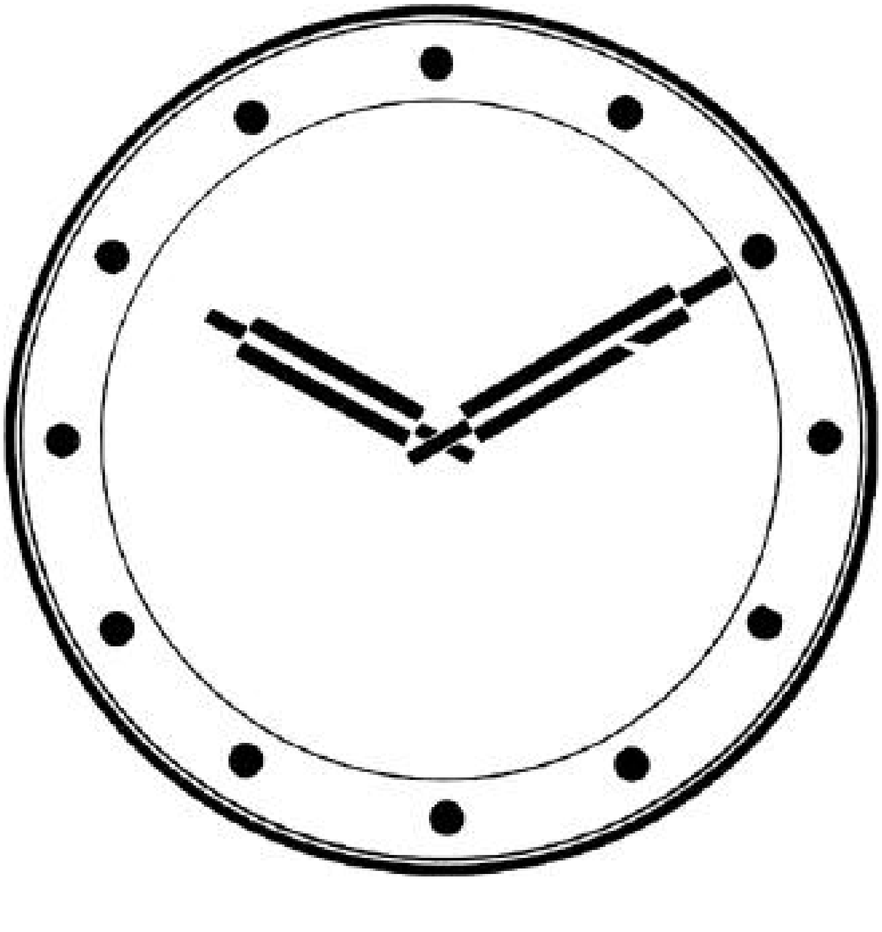
TIME II
CLOCK CARD



CONGRATULATIONS

You now own APPLIED ENGINEERING’S TIME II real time clock! APPLIED ENGINEERING is a leading manufacturer of Apple peripherals. Because of the time and care taken in the design and manufacture of your clock-, we are sure that you will enjoy the use of it for many years to come.

The authors have taken due care in preparing this manual and the programs in it. In no event shall the authors or publishers be liable for incidental or consequential damages in connection with or arising Out of the furnishing, performance or use of any of the programs herein.

If you have not already done so, please take a few minutes to complete and mail your OWNER/WARRANTY REGISTRATION CARD. This registration card will register your TIME II with the factory and include you in the list of TIME II owners. If you don’t send us this card, you will not receive any newsletters and information frequently mailed to TIME II owners. So please mail the completed card.

INSTALLING TIME II IN YOUR APPLE

The Time II real time clock simply plugs into a connector inside your Apple. Care must be exercised however, so follow these instructions exactly.

I) TURN OFF THE APPLE’S POWER SWITCH: This is very important to prevent damaging the computer as well as your Time II.

2) Remove the cover from the Apple. This is done by pulling up on the cover at the rear edge (the edge farthest from the keyboard) until the two corner fasteners pop apart. Then slide the cover backward until it comes free.

3) Inside the Apple, across the rear of the main circuit board, there is a row of eight long narrow sockets called “slots”. You can plug your Time II into any slot except slot 0. Slot 7 is the preferred slot. Insert the ‘fingers” of the circuit .board into the slot you want. The fingers will enter the slot with some friction, and will seat firmly.

4) Replace the Apple’s cover by sliding the front edge into place, then press down on the two rear corners until they pop into place.

5) Now turn on your Apple and continue.

HOW TO READ SECONDS FOR SLOT -7-*

Let’s start with a simple program.

```
1Ø HOME                    clear screen
2Ø A = 49394               A = Slot 7 input port address
3Ø B = 49393               B = Slot 7 output port address
4Ø POKE B,16                Set bold line high (This is done so that one
                           number doesn’t change while we’re reading
                           another number. Actually, it isn’t even
                           necessary here, but this practice will
                           develop good programming habits for later on.)
5Ø POKE A,32               Set unit seconds address (see programming table)
6Ø VTAB (12)               Move cursor
7Ø PRINT PEEK (A)          Print data from input port (seconds—units)
8Ø POKE 9,Ø                Set bold line low (this will let the clock
                           continue counting by putting a 9 in the
                           output port)
9Ø GOTO 4Ø                 Do it all again
```

* You can use this program in any slot by subtracting 16 from A and B for each slot back from 7 you go. For example, in slot 5 use A ——40378; B = 49377

Program For Any Slot

ADD these lines to your read seconds program.

```
5   HOME
1Ø  INPUT “WHAT SLOT IS THE CLOCK IN” ;S
```

```
2Ø A = 49282 + (S+16)

3Ø B = 49281 + (S*16)
```

Now run your new program. This program is said to be portable because it is NOT slot independent.

Now lets take a look at a program that will set the hours, minutes and seconds as well as display them.

```
1Ø   HONE

2Ø   INPUT “WHAT SLOT IS THE CLOCK IN?”; S

3Ø   A = 49282 + (8*16)

4Ø   B = 49281 + (S+16)

5Ø   PRINT “HIT D TO DISPLAY TINE”

6Ø   PRINT “HIT S TO SET TIME”

7Ø   INPUT A$

8Ø   IF A$=”D” GOTO 130

9Ø   lF A$=“S” GOTO 500

1ØØ  PRINT

11Ø  PRINT “YOU HIT THE WRONG KEY”   PRINT

12Ø  GOTO 80

13Ø  POKE B,16         Set hold line high

14Ø  POKE A,32         Set seconds units address

15Ø  SU = PEEK (A)     Make SU=Data in register
                       defined by the address
                       above (32)

16Ø  POKE A,33         Set seconds tens address

17Ø ST = PEEK (A)      Make STData in register
                       defined by the address
                       above (33)

18Ø POKE A,34  Set minutes units address
```

```
19Ø MU = PEEK (A)                          Hake MU=Data in register
                                           defined by the address
                                           above (34)

2ØØ POKE A35                               Set minutes tens address

21Ø MT = PEEK (A)                          Hake MT=Data in register
                                           defined by the address
                                           above (35)

22Ø POKE A,36                              Set hours units address

23Ø HU = PEEK (A)                          MAKE HU=Data in register
                                           defined by the address
                                           above (36)

24Ø POKE A,37                              Set hours tens address

25Ø HT = PEEK (A)                          Make HT=Data in register
                                           defined by the address
                                           above (37)

26Ø  VTAB (1)                              Move Cursor

27Ø  PRINT HT;HU;” :“;MT;MU;”:” ;ST;SU     Print time

28Ø  PRINT

29Ø  PRINT “HIT ANY KEY TO END”

3ØØ  IF PEEK (-16384)>127 THEN             Has key been hit?
     GOTO 35Ø

31Ø  POKE -16368,Ø                         Reset hit key check

32Ø  PRINT

33Ø  POKE B,Ø                              Lower hold line

34Ø  GOTO 13Ø                              Do it all again

35Ø  POKE B,Ø                              Lower hold line

36Ø  END                                   End

5ØØ  POKE B,16                             Set hold line high

51Ø  INPUT “HOUR TENS” HT                  Input hours tens

52Ø  POKE A,5                              Write register to address
                                           input part
```

```
53Ø  POKE B,HT + 16                       Write the data plus 16
                                          to data output port
                                          (you need the 16 to keep
                                          the hold line high)

54Ø  POKE A,5 +- 16                       Write register plus 16 to
                                          address input port (this
                                          raises the write line)

55Ø POKE A,5                              Write register to input
                                          port (this lowers the write
                                          line)

56Ø  INPUT “HOUR UNITS”;HU

57Ø  POKE A,4

58Ø  POKE B,HU + 16

59Ø  POKE A,4 + 16

6ØØ  POKE A,4

61Ø  INPUT ‘MINUTE TENS” ;MT

62Ø  POKE A, 3

63Ø  POKE B,MT + 16

64Ø  POKE A,3 + 16

65Ø  POKE A,3

66Ø  INPUT ‘MINUTE UNITS’;HU

67Ø  POKE A,2

68Ø  POKE B,MU + 16

69Ø  POKE A.2 + 16

7ØØ  POKE A,2

71Ø  POKE B,Ø                             Set hold line low

72Ø  HOME

73Ø  GOTO l3Ø                             Go print time just set
```

Let’s try a different technique in our next program.

Type in Program #4

```
1Ø   HOME

2Ø   DIM TIME (6)

3Ø   INPUT “WHAT SLOT IS THE CLOCK
     IN?”;S

4Ø   HOME

5Ø   A = 49292 + (S*16)

6Ø   B = 49281 + (S*16)

7Ø   POKE B16                              Set hold line high

8Ø   N = Ø

9Ø   FOR C 37 to 32 STEP —1

1ØØ  POKE A,C                              Read time

11Ø  TIME (N) = PEEK (A)

12Ø  N = N + 1

13Ø  NEXT C

14Ø  D = TIME (Ø)

15Ø  IF D>7 THEN A$ —

16Ø  IF TIME (Ø)>7 THEN TIME (Ø) =         12 or 24 hour format
     TIME (Ø) — 8                          AM or PM

165  IF D<8 AND TIME (Ø)<4 THEN
     A$ = "A.M."

17Ø  IF 0<8 AND TIME (Ø)>3 THEN
     A$ = "P.M."

18Ø  IF TIME (Ø)>3 THEN TIME (Ø)-
     TIME (Ø) — 4

19Ø  VTAB (1)

2ØØ  PRINT THE TIME IS” ;TIME(Ø)
;TIME:(1):":":TIME(2);TIML(3)              Print time
;":":TIME($);TIME(%);":"A$
```

```
21Ø   PRINT
22Ø   PRINT
23Ø   PRINT
24Ø   IF PEEK (-16384)>127 THEN             Has key been hit
      GOTO 35Ø
25Ø   POKE -16368,9                         Reset key check
26Ø   PRINT
27Ø   POKE B,9                              Set hold low
28Ø   GOTO 70
39Ø   POKE B,9                              Set hold low
36Ø   END
```

Check your typing and RUN it.

Your screen should look like this:

THE TIME IS HH:MM:SS:AM or PH

Don’t be concerned if AM or PM isn’t right, because as of yet you have had no way to set them.

The preceding program was quite a bit different from program #3 due mostly to the use of ARRAYS. If you are not familiar with ARRAYS, have a look at pages 108-111 in your APPLESOFT TUTORIAL.

Notice how bits 4 and 8 were set to Ø before it is read at line 16Ø and l8Ø. If this were not done, the HOUR TENS number would be wrong.

DATA NAME	REG	+16	+32	DATA BITS			
				1	2	4	8
Seconds Units	0	16	32	*	*	*	*
Secoxida Tens	1	17	33	*	*	*	
Minutes Units	2	18	34	*	*	*	
Minutes Tens	3	19	3h	*	*	*	
Hours Units	4	20	36	*	*	*	*
Hours Tens (AM/PM)	5	21	37		*	Ta	Tb
Day of Week	6	22	38	*	*	*	
Date Units	7	23	39	*	*	*	
Date Tens (Leap Yr)	S	24	40	*	*		
Month UnIts	9	25	41	*	*	*	*
Month Tens	10	26	42	*			
Year Units	11	27	43	*	*	*	*
Tea Tens	12	28.	44	*	*	*	*
Interupt**	15	31	47	*	*	*	*

Seconds units & tens are reset to zero irrespective of input data when a write instruction is executed

Ta = Ø for AM, 1 for PM
Tb = Ø for 12 hour format, T for 24 hr.format

Tc = Ø for 28 days in month 2, 1 for 29 days In month 2 ***

*   Bits Used

**   Interupts are only active during a read operation and hold line set low. (1Ø24 HZ interrupt is not hold state dependent

*** If Te Previously set to “1", upon completion of month 2 (February) day 29, Tc will automatically be reset to “Ø"

12 OR 24 HOUR FORMAT, AM OR PM

To start off with, if you select the 24 hour format you need not be concerned with AM & PM.

Let’s look at the part of our programming chart concerned with 12 and 24 hour format and A.M., P.M.

DATA NAME	REG	+16	+32	DATA BITS			
				1	2	4	8
HOURS TENS	5	21	37	*	*		

If Bit 4 is high, then it is P.M.

If Bit 4 is low, then it is A.M.

If Bit 8 is high. then 24 hour format
If Bit 8 is low, **then** 12 hour format

LEAP YEAR?

Looking at the part of our programming chart concerned with leap year:

DATA NAME	REG	+16	+32	DATA BITS			
				1	2	4	8
HOURS TENS	8	24	40	*	*		

If Bit 4 is high, (1) then there are 29 days in month 2 (February)
If Bit 4 is low, (Ø) then there are 28 days in month 2 (February)

Upon completion of month 2, day 29, bit 4 will be reset low (Ø)

+-30 SECOND ADJUST

TIME II clock-calendar has the ability to adjust the seconds ± 30.

Try this program:

```
1Ø  B = 49393
2Ø  POKE B,32
3Ø  FOR DELAY = 1 to 25
4Ø  NEXT DELAY
5Ø  POKE B,O
6Ø  END
```

To predict what this program will do, let’s look at some examples.

If the time is 12:16:24 when the above program is run, the time will be at back 24 seconds. The clock will continue running from 12:16:00.

If the time is 12:16:31 when the above program is run, the time will be let ahead 29 seconds to 12:17:00.

Line 20 writes 32 to the date output port and sets the ±30 ADJUST line HIGH(1). Lines 3D and 40 give the necessary time delay for the clock chip to read it. (31.25 milliseconds minimum is required.) Line 50 resets the ±30 second line LOW (0) completing the SET routine.

If You wish, you can add this routine to the programs you already have.

In this program, we take advantage of most of the TIME II features. Enter it and experiment with it. When you have finished, be sure to save it on tape or d~sk.

```
1Ø   TEXT

2Ø   HOME

3Ø   INPUT “WHAT SLOT IS THE CLOCK            What slot?
     IN?”; S

4Ø   PRINT

5Ø   A = 49282 + (S416)

6Ø B = 49281 + (S*16)

7Ø Dliii X(13)

8Ø DIII Y$(7)                                 Dimension array X,Y,Z

9Ø DIM Z$(13)

1ØØ  DATA “SUNDAY “, “MONDAY",
     "TUESDAY “, “WEDNESDAY“,
     THURSDAY “, “FRIDAY“,
     "SATURDAY"

11Ø  DATA “SECOND UNITS”, “SEC
     OND TENS”, “MINUTE UNITS”
     ,“MINUTE TENS", “HOUR UNIT
     S”, “HOUR TENS”                          Enter data

12Ø DATA “DAY OF WEEK Ø = SUN,
     1 = MON, ETC.”, “DATE UNITS
     “,“DATE TENS’

13Ø  DATA “MONTH UNITS”, "MONTH
     TENS”, “YEAR UNITS", “YEAR
     TENS”

14Ø  DATA “SECONDS TENS”, “SECON
     D UNITS”

15Ø  FOR C = Ø TO 6

16Ø  READ Y$(C)                               Load Y$ Array

17Ø  NEXT C

18Ø  FOR D = Ø TO 12

19Ø  READ Z$(D)                               Load Z$ Array

2ØØ  NEXT 'D
```

```
21Ø   PRINT "TYPE D TO DISPLAY DAT
      E-TIME”

22Ø   PRINT “TYPE S TO SET DATE-TI
      ME”

23Ø   PRINT TYPE A TO ADJUST TIME
      +- 30 SEC.”

24Ø   PRINT "TYPE E TO END PROGRAM
      "

25Ø   INPUT Q$

26Ø   PRINT

27Ø   HOME                                      What Subroutine?

28Ø   IF Q$ = “D” THEN GOTO 360

29Ø   IF Q$ = “5” THEN GOTO  640

3ØØ   IF Q$ = “A" THEN GOTO 1030

31Ø   IF Q$ = “E” THEN END

32Ø   PRINT

33Ø   PRINT “YOU HIT THE WRONG KEY
      !"

34Ø   PRINT

35Ø   GOTO 210

36Ø   I = Ø

37Ø   POKE B,16

38Ø   FOR E 44 to 32 STEP —1

39Ø   POKE A,E                                  Load date time

4ØØ   X(I) = PEEK (A)

41Ø   1= I + 1

42Ø   NEXT E

43Ø   IF F>7 THEN P$ = " "

44Ø   IF X(4)>3 THEN X(4) = X(4)
      -4                                        12 or 24 hour format

45Ø F = X(7)                                    AM or PM
```

```
46Ø   IF F>7 THEN P$ = ""

47Ø   IF X(7)>7 THEN X(7) = X(7)
      —8

48Ø   IF F<6 AND X(7)<4 THEN P                  12 or 24 hour format
      $ = “AM”                                  AM or PM

49Ø   IF F<8 AND X(7)>3 THEN P
      $= "PM"

5ØØ   IF X(7)>3 THEN X(7) = X(7)
      -4

51Ø-  VTAB (1)

52Ø   PRINT "THE DATE IS “;X(2);X(
      3);"/";X(4);X(5);"/";X(Ø);X(
      1)

53Ø   PRINT “TODAY IS “;Y$(X(6))                Print time

54Ø   PRINT “THE TIME IS “;X(7);X(
      8);":";X(9);X(Ø);":"X(11);
      X(12) P$

55Ø   PRINT

56Ø   PRINT “HIT ANY KEY TO END”

57Ø   IF PEEK (—16384)>L27 THEN
      GOTO 620                                  End program?

58Ø   POKE -16368,0

59Ø   PRINT

6ØØ   POKE B,0                                  Set hold line low

61Ø   GOTO 369                                  Get new time

62Ø   POKE B,W

63Ø   END                                       End program

64Ø   G = 12

65Ø   POKE B,16

66Ø   FOR G = 12 to 0 STEP -1                   Set date/time

67Ø   PRINT Z$(G)
```

```
68Ø  INPUT N
69Ø  X(G) = N
7ØØ  NEXT G
71Ø  INPUT ‘IS THIS A LEAP YEAR”
     ;Y$
72Ø  IF Y$ = “Y” GOTO 78Ø
73Ø  IF Y$ = “N” GOTO 78Ø
74Ø  PRINT
75Ø  PRINT “TYPE Y OR N PLEASE!”
76Ø  PRINT
77Ø  GOTO 71Ø
78Ø  INPUT “12 OR 24 HOUR FORMAT
     ";Y
79Ø IF Y = 24 THE$ X(5) = X(5) +
     8
8ØØ  IF Y = 12 GOTO 86Ø

81Ø  IF Y = 24 GOTO 94Ø
82Ø  PRINT
83Ø  PRINT “TYPE 12 or 24 PLEASE!
84Ø  PRINT
85Ø  GOTO 780
86Ø  INPUT “AM OR PM”
87Ø  IF X$ = “PM” THEN X(5)= X(5
     )+4
88Ø  IF X$ - “PH” THEN GOTO 94Ø
89Ø  IF X$ = “AM” THEN GOTO 94Ø
9ØØ  PRINT
91Ø PRINT “TYPE AM OR PM PLEASE!
     "
```

Set date/time

```
92Ø   PRINT
93Ø   GOTO 86Ø
94Ø   FOR H = 12 TO Ø STEP -1
95Ø   N = X(H)
96Ø   GOSUB 1Ø2Ø
97Ø   NEXT H                              Set date/time
98Ø   POKE B,9
99Ø   PRINT
1ØØØ  HOME
1Ø1Ø  GOTO 36Ø
1Ø2Ø  POKE A,H: POKE B,N +16: POKE
      A,H + 16: A,N: RETURN
1Ø3Ø  POKE B,32
1Ø4Ø  FOR J = 1 TO 25
1Ø5Ø  NEXT J                              ±30 second adjust
1Ø6Ø  POKE B,Ø
1Ø7Ø  GOTO 36Ø
```

Let's review what we've learned so far:

1) Each digit has its own address.

2) Set hold line high when reading data.

3) Add 32 to register when reading it.

4) Set hold line low after reading data.

5) The 12 and 24 hour & AM/P.M. bits must be set low when READING the DATE TENS DATA.

7) The DAY of WEEK is a number from Ø to 6.

8) Write 32 to data output port to set ± 30 Second adjust 111GB.

9) Write 0 to data output port to reset ± 30 second adjust.

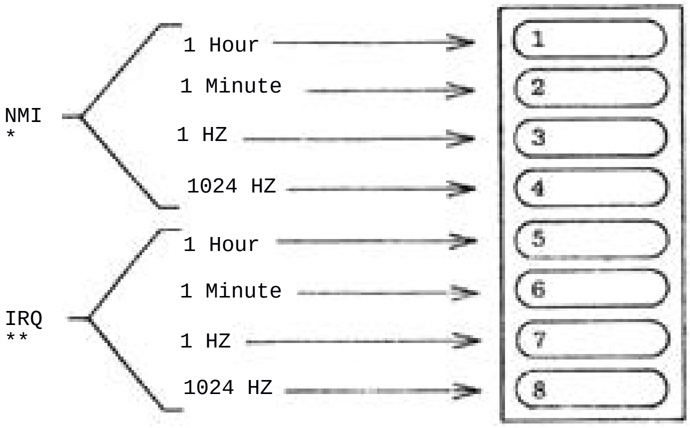
10) When setting the time: a) SET HOLD LINE hig~h (see line 500 in program #3) b) write the register of the DATA you want to the address input port (see line 520 in Program #3) c) write DATA +16 to output port (see line 530) d) write register +16 to address input port (see line 550). Repeat parts b, c, d and e until all registers are loaded. Set HOLD line LOW (see line 710 in program #3)

TIME II INTERRUPTS

There are 4 interrupts lines, running at 1024 HZ, 1 HZ, 1 Minute and 1 Hour. These interrupt lines are driven low (Ø) when a read is done on address 15. Any of the four signals can be switched to either NMI (nonsaskable interrupt) or IRQ (interrupt request). In fact, it is possible to set one interrupt to NMI and another to IRQ!

The interrupt lines are driven by open collector HAND GATES and are driven low for 122.1 aS (except the 1024 HZ signal which has a 50% duty cycle).

Refer to the below drawing to set the INTERRUPTS.



*NMI Non-maskable interrupt. When this line is pulledlow (Ø) the Apple begins an interrupt cycle and jumps to an interrupt handling routine at location $3FB.

**IRQ Interrupt request. When this line is pulled low (Ø) the Apple begins an interrupt cycle ONLY if the 6502's (interrupt disable) flag is not set. If so, the 6502 will jump to the interrupt

handling routine whose address is stored in locations $3FE and $3FF.

TIME BASE CALIBRATION

Your TIME II has a quartz crystal time base which oscillates at 32,768 HZ (2^{15}). This frequency can be adjusted up or down approximately 2 HZ by the trimmer capacitor which is next to the dip switches at the rear of the board. Your TIME II was calibrated at the factory to 32,768.0 HZ ± .0002%.

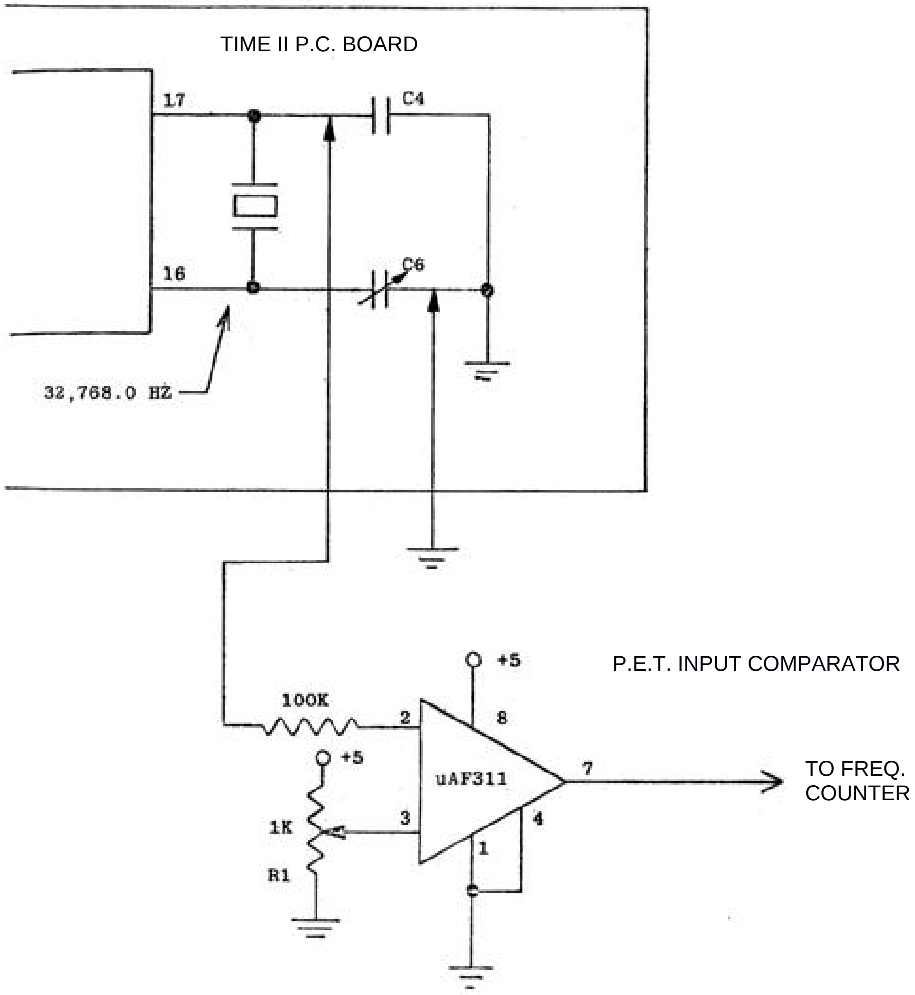
The manufacturer of the crystal specifies that the frequency may age .0005% or 5 parts per million in one year.

RECALIBRATING TIME BASE:
You can recalibrate the time base with a frequency counter; however, most frequency counters have a input capacitance that is too high and a input resistance that is too low. Check your owners manual. The input capacitance must be less than 2pF and input resistance should be greater than 50 M ohms. If it is not, you can make a simple circuit to buffer the signal before it goes into your frequency counter. (See page 21 for schematic.)

If the above discussion seems like a lot of work (and it is, however it is very unlikely that this will be required for several years). But since we developed this circuit, we wanted you to have access to it (it never hurts to over explain).

You can adjust the frequency by turning the trimmer capacitor C6 so that the clock matches a known time standard. REMEMBER, YOUR TIME II WAS CALIBRATED AT THE FACTORY SO DON'T TURN THE TRIMMER UNLESS YOU CAN DO YOUR OWN CALIBRATION.

BUFFER CIRCUIT FOR FREQUENCY COUNTER



ADJUST R1 FOR STABLE OUTPUT.