HOW TO GET NIST-TRACEABLE TIME ON YOUR COMPUTER

Michael A. Lombardi
NIST
Time and Frequency Division
325 Broadway
Boulder, CO 80303

ABSTRACT

Many computer applications require accurate timekeeping. Accurate time is required by computer systems that make and record measurements so that the data is timestamped correctly. Plus, accurate time is often needed to synchronize computer networks. This paper describes a number of different ways that you can use to get accurate time that is traceable to NIST on your computer system. This includes receiving time signals by radio, by telephone, or through the Internet. The cost and performance of each technique is also described.

INTRODUCTION

You probably know that PC-compatible computers contain a built-in clock. If you've ever relied on a PC clock for timekeeping, you probably also know that it isn't very accurate. Your wristwatch and most of the clocks in your home probably keep better time than the clock in your PC. Of course, most of us aren't particularly concerned if the clock is off by a few seconds (or even minutes or hours). If we can tell when we made the most recent revision to a file, we'll probably be satisfied with our PC's timekeeping ability.

Some PC applications, however, require accurate timekeeping (to the nearest second or better). For example, the computers in a financial institution must keep very accurate records of when transactions were completed, for legal or other reasons. Computer systems that make physical measurements and acquire scientific data need to know exactly when the measurements were made. Software used on a manufacturing floor may need to turn a piece of equipment on or off at a specified time. And any system involved with synchronous communications must keep accurate time. For example, radio and TV stations may need computers that can switch feeds or link up with remotes at the exact right time.

This article describes several methods that make accurate PC timekeeping possible. Before looking at these methods, let's look at how the PC keeps time.

The Clocks Inside Your PC

Two time-of-day clocks reside in every AT-compatible computer (and in many PC and XT-compatible computers as well). These clocks go by several different names, but for simplicity, we'll call them the software and hardware clocks. The software clock runs only when the computer is turned on. It stops when the computer is turned off. The hardware clock uses a battery and runs even while the computer is turned off.

Every computer that runs DOS has a built-in software clock. On PC or XT-compatibles, this clock is driven by the Intel 8253 timer-counter chip or an equivalent device. AT-compatible computers use a device equivalent to an Intel 8254 timer-counter. The timer-counter is programmed by the BIOS to generate an interrupt every 54.936 milliseconds, or about 18.206 times per second. Another BIOS routine counts the interrupt requests and generates a time-of-day clock that can be read or set by other software programs. For example, DOS uses the information from the software clock when it date and time stamps files.

The software clock is useful, but it has several limitations. First, the software clock is a poor timekeeper. Its accuracy is limited by the stability of the interrupt requests. Any change in the interrupt request rate causes the clock to gain or lose time. If you leave your computer turned on for long periods, the software clock might be off by large amounts, perhaps a minute or more for every day that the computer was left turned on. It's also possible for an ill-behaved software program to use the timer-counter for another purpose and change its interrupt rate. This could cause the clock to rapidly gain or lose time.

Another problem with the software clock is that it cannot display all possible time-of-day values. The resolution of the clock is limited to the interval between interrupts, or about 55 milliseconds as stated earlier. Only times that are even multiples of this interval can be displayed. For example, 00:00:01.00 could never be displayed by the software clock. The closest possible values it can display are 00:00:00.98 and 00:00:01.04.

The single biggest limitation of the software clock, however, is that when the computer is turned off the clock stops running. On the original IBM-PC, this meant that you manually had to set the clock each time you turned the computer on. You could purchase an optional battery-backed clock for the PC, but there were several different standards, and not all of them worked with all software packages. This problem was addressed with the introduction of the IBM-AT in 1984, which included a battery-backed hardware clock as standard equipment. An AT-compatible hardware clock is included with every 286, 386, and 486 computer produced today.

The hardware clock is based on the Motorola 146818 Real Time Clock Chip, or a functionally equivalent device. The clock is supported by the AT BIOS, and BIOS services are available that let software programs read and set the clock.

The hardware clock is a CMOS device that consumes very little power. When the computer is turned off, it runs on batteries. When the computer is turned back on, the software clock starts running again and sets itself (within 1 second) to the hardware clock. Although the two clocks are synchronized at start-up, they may run at very different rates and will probably gain or lose time relative to each other while the computer is running.

The hardware clock is updated once per second and cannot display fractions of a second. For this reason, it cannot be read or set within better than a second. The accuracy of the hardware clock is determined by the quality of its timebase oscillator (typically a 32.768 kHz crystal). These crystals are economical, costing less than $1 in single quantities.

However, they offer only marginal timekeeping performance. They are sensitive to temperature and other factors and are often not calibrated at the factory. Even under the best conditions, these oscillators are not likely to be stable to better than 1 part per million (about 0.1 seconds per day). In actual operation, most hardware clocks seem to gain or lose time at a rate of about 1 to 15 seconds per day, with 5 or 6 seconds per day being typical. Although the hardware clock usually outperforms the software clock by a considerable amount, its performance often pales in comparison to even a low-cost wristwatch.

As you can tell by now, neither the software or hardware clock is suitable for accurate timekeeping. Fortunately, however, there are several ways you can keep accurate time on your PC if your application demands it. Let's start by looking at ways to set your computer's clock on time.

Getting Time by Modem

Since 1988, it has been quite easy to set your computer's clock to precise time from the National Institute of Standards and Technology (NIST). That year marked the beginning of NIST's Automated Computer Time Service (ACTS). Using ACTS requires only a modem and some simple software. When a computer connects to ACTS by telephone, it receives an ASCII timecode. The information in this timecode is then used to set the computer clock to the correct time. [1]

ACTS is usable at either 300 or 1200 baud, with 8 data bits, 1 stop bit, and no parity. The 1200 baud timecode is transmitted every second, and contains more information than the 300 baud timecode, which is transmitted once every 2 seconds. A description of the 1200 baud timecode is given in Figure 1.

When you connect to ACTS, the last character in the timecode is an asterisk (*). The asterisk is called the on-time marker (OTM). The time values sent by the timecode refer to the arrival time of the OTM. In other words, if the timecode says it is 12:45:45, this means it is 12:45:45 when the OTM arrives.

Of course, there is some delay between the time the OTM leaves NIST and the time it arrives at your computer. Part of this delay is the actual data transmission time, and part of it is the time it takes your modem to process the incoming data and feed it to your computer. For phone calls within the continental United States, the data transmission delay is usually smaller than the modem processing delay. If the phone call goes through a satellite link, however, the data transmission time will account for most of the delay.

Since ACTS knows that the OTM will be delayed between the time it leaves NIST and the time it gets to your computer, it sends the OTM out 45 milliseconds early. This 45 milliseconds includes the 8 milliseconds that it takes to send the OTM at 1200 baud, 7 milliseconds transmission time to allow for travel from NIST to the average user in the United States, and 30 milliseconds to compensate for the modem processing delay (typical for several modem brands).

JJJJJ YR-MO-DA HH:MM:SS TT L UT1 msADV UTC(NIST) <OTM>

JJJJJ is the Modified Julian Date (MJD). The MJD is the last five digits of the Julian Date, which is simply a count of the number of days since January 1, 4713 B.C. To get the actual Julian Date, add 2.4 million to the MJD.

YR-MO-DA is the date. It shows the last two digits of the year, the month, and the current day of month.

HH:MM:SS is the time in hours, minutes, and seconds. The time is always sent as Coordinated Universal Time (UTC). An offset needs to be applied to UTC to obtain local time in the United States. For example, Mountain Time in the United States is 7 hours behind UTC during Standard Time, and 6 hours behind UTC during Daylight Saving Time.

TT is a two-digit code (00 to 99) that indicates whether the United States is on Standard Time (ST) or Daylight Saving Time (DST). It also indicates when ST or DST is approaching. This code is set to 00 when ST is in effect, or to 50 when DST is in effect. About 48 days prior to a time change, the code starts counting the days until the change. When ST is in effect, the code counts down from 99 to 51. When DST is in effect, the code counts down from 49 to 01.

L is a one-digit code that indicates whether a leap second will be added or subtracted at midnight on the last day of the current month. If the code is 0, no leap second will occur this month. If the code is 1, a positive leap second will be added at the end of the month. This means that the last minute of the month will contain 61 seconds instead of 60. If the code is 2, a second will be deleted on the last day of the month. Leap seconds occur at a rate of about one per year. They are used to correct for irregularity in the earth's rotation.

UT1 is a correction factor for converting UTC to an older form of universal time that is still used in navigation. It is always a number ranging from -0.8 to +0.8 seconds. This number is added to UTC to obtain UT1.

msADV is a five-digit code that displays the number of milliseconds that NIST advances the time code. It is originally set to 45.0 milliseconds. If you return the on-time marker (OTM) four consecutive times, it will change to reflect the actual one way line delay.

The label UTC(NIST) is contained in every timecode. It indicates that you are receiving Coordinated Universal Time (UTC) from the National Institute of Standards and Technology (NIST).

The on-time marker (OTM) is a single character sent at the end of each timecode. The OTM is originally an asterisk (*) and changes to a pound sign (#) if ACTS has successfully measured the round trip line delay.

Figure 1 - The 1200 Baud ACTS timecode

The 45 millisecond advancement of the OTM typically removes most of the delay. For example, if you are calling from Chicago and the actual delay is 50 milliseconds, the OTM will arrive at your computer only 5 milliseconds late with about 90% of the delay already removed. However, if you are making an overseas call or any call that goes through a satellite, the delay can be as large as 300 milliseconds or more.

Fortunately, ACTS lets you measure the actual line delay, so you can remove as much of the delay as possible. To do so, your software has to return the OTM to ACTS after it receives it. Each time the OTM is returned, ACTS measures the actual delay. After 4 consecutive measurements have been made, ACTS will begin advancing the OTM by the actual delay amount. For example, if the actual delay is 50.4 milliseconds, ACTS will send the OTM out 50.4 (instead of 45) milliseconds early. Once ACTS begins using the actual measured delay, the OTM changes from an asterisk to a pound sign (#). At this point, the OTM is arriving at your computer within 1 millisecond of the correct time.

In very rare instances, ACTS won't be able to measure the actual delay. This can happen if the phone call goes by satellite in one direction (long delay) and by land in the other direction (short delay). If this happens, the standard 45 millisecond advancement will be used, even if your software returns the OTM.

There are a number of software packages available (both commercial and shareware) that call ACTS, including a recently published Windows utility. [2] The manufacturer's list at the end of this paper lists the names and addresses of some known vendors.

There are some instances where you may need to write your own ACTS software. For example, you may need to write an ACTS routine to embed in another application. If so, you'll find that writing a rudimentary ACTS program is quite simple. Figure 2 is a listing of a simple ACTS program written in QBASIC, the language included with DOS (versions 5.0 and higher). This program should work on most PC-compatibles. This program dials ACTS, sends back the OTM, and uses the QBASIC functions DATE$ and TIME$ to set the date and time on your computer.

The program in Figure 2 is included to show how simple that ACTS software can be, but it does have some limitations. For example, it can only set a clock to Coordinated Universal Time (UTC). A small routine would have to be added to convert UTC to your local time. Also, it only works with COM1 or COM2, since these are the only two COM ports supported by QBASIC. It would have to be modified (using routines written in another language) to work with COM3 or COM4.

The program may also have to be modified slightly to work with high speed modems that use data compression and/or error-correcting protocols. It should work without any problems if you have a 2400 baud or slower modem. If you have a faster modem, you need to make sure that all the data compression schemes and error correcting protocols are turned off, or else you may have difficulty in connecting to ACTS. Some modems automatically compensate when they connect to slower speed systems. Other modems require you to send them one or more specific commands. If the program doesn't work, find the appropriate commands in your modem manual, and add them to the modem initialization string at the beginning of the program (by modifying the variable named ATTENTION$).

```
DECLARE SUB StripCRLF (x$)
' ***********************************************************************
' Software to access the NIST Automated Computer Time Service
' (Works with QBASIC compiler included with MS-DOS 5.0)
' ***********************************************************************
'       Define Modem Initialization String
' ***********************************************************************

        attention$ = "ATZX3"

' ***********************************************************************
'       Define number of timecodes to receive
' ***********************************************************************

        receivelines% = 10

' ***********************************************************************
'       COM Port Setting (set to either COM1: or COM2:)
' ***********************************************************************

        comport$ = "COM2:"

' ***********************************************************************
'       Baud Rate Setting (set to 1200 for 1200 baud, or 300 for 300 baud)
' ***********************************************************************

        baudrate$ = "1200"

' ***********************************************************************
'       Dialing Method (set to T for tone dialing, P for pulse dialing)
' ***********************************************************************

        pulse$ = "T"

' ***********************************************************************
'       Phone Number (eliminate area code, add access codes if necessary)
' ***********************************************************************

        phone$ = "1-303-494-4774"

' ***********************************************************************
'       housekeeping functions
' ***********************************************************************

        ON ERROR GOTO Handler                   ' trap errors
        CLS                                     ' clear screen
        IF baudrate$ = "1200" THEN codefix% = 6 ELSE codefix% = 0

' ***********************************************************************
'       print opening text on screen
' ***********************************************************************
```

```
        PRINT "NIST Automated Computer Time Service (ACTS)"
        PRINT
        PRINT "Dialing ACTS, please wait ..."
        PRINT

' ***************************************************************************
'       set up modem for 8 data bits, 1 stop bit, and no parity
' ***************************************************************************

        communicate$ = comport$ + baudrate$ + ",N,8,1"

' ***************************************************************************
'       call the Automated Computer Time Service (ACTS)
' ***************************************************************************

        OPEN communicate$ FOR RANDOM AS #1 LEN = 256

                PRINT #1, attention$                ' initialize modem
                g = TIMER
                DO
                LOOP UNTIL ABS(TIMER - g) > 3       ' pause for 3 seconds
                PRINT #1, "ATD" + pulse$ + phone$   ' dial number
                printline% = 1                      ' count first line

        DO

                timecode$ = ""                      ' clear timecode$
                DO                                  ' read in characters
                        s$ = INPUT$(1, #1)
                        timecode$ = timecode$ + s$
                LOOP UNTIL s$ = "#" OR s$ = "*" OR s$ = CHR$(13)
                CALL StripCRLF(timecode$)           ' clean up timecode

                SELECT CASE timecode$               ' check for errors
                        CASE "NO DIALTONE", "BUSY", "NO CARRIER", "NO ANSWER"
                                CLOSE #1
                                GOTO Handler
                END SELECT

                otm$ = s$                           ' check for on-time marker

                IF otm$ = "*" OR otm$ = "#" THEN

                        PRINT #1, otm$              ' send OTM back to NIST
                        PRINT timecode$             ' print timecode on screen
                        printline% = printline% + 1 ' increment counter

                END IF

        LOOP UNTIL printline% > receivelines%      ' get next timecode
```

```
        t$ = MID$(timecode$, 10 + codefix%, 8)       ' get time from timecode
        TIME$ = t$                                    ' set time on computer
        d$ = MID$(timecode$, 4 + codefix%, 5) + "-"   ' get date from timecode
        d$ = d$ + MID$(timecode$, 1 + codefix%, 2)
        DATE$ = d$                                    ' set date on computer
        PRINT #1, "ATH"                               ' hang up telephone
        CLOSE #1

        PRINT
        PRINT "The computer's clock has been set."    ' print message on screen

        END

' ****************************************************************************
Handler:                                              ' print error message

        PRINT
        PRINT "Communications Error  -  Check Phone Line and COM Port setting."
        PRINT

END
' ****************************************************************************

SUB StripCRLF (x$) STATIC

        ' strips all carriage returns and line feeds from a string

        y$ = ""
        FOR i% = 1 TO LEN(x$)
                z$ = MID$(x$, i%, 1)
                IF ASC(z$) <> 10 AND ASC(z$) <> 13 THEN
                        y$ = y$ + z$
                END IF
        NEXT i%
        x$ = y$

END SUB
```

Figure 2 - Sample ACTS software

A call to ACTS takes just seconds (the sample program keeps you on-line for about 10 seconds at 1200 baud). Since the time setting process is so quick, ACTS limits your on-line to 56 seconds, or just 28 seconds if all lines are busy.

ACTS is a very accurate source of time. In fact, it provides far more accuracy than the typical PC clock can handle. If you measure the line delay, ACTS is capable of setting a computer clock to within 1 millisecond of NIST time. Of course, as discussed earlier, the software clock in a PC ticks about once every 55 milliseconds. Even if the OTM arrives within

1 millisecond of the correct time, you can only guarantee that the clock has been set within 55 milliseconds. And even then, the clock won't stay set for very long. For example, a typical PC clock that gains 5 seconds per day will gain 1 millisecond every 17 seconds. Although millisecond time setting is a moot point on a PC, some computer clocks can take full advantage of ACTS potential accuracy.

ACTS isn't the only source of accurate time by modem. The United States Naval Observatory (USNO) in Washington, DC operates a computer time service similar to ACTS. The USNO service limits your on-line time to 60 seconds and works at 1200 baud only. There are several commercial and shareware software packages that call the USNO service (some call both ACTS and the USNO), and at least one sample source code listing has been published. [3]

The USNO service is slightly less capable than ACTS. Its timecode contains less information than the ACTS timecode, and it is not capable of measuring line delay or of advancing the on-time marker. If your modem supports remote digital loopback tests you can measure the line delay by echoing characters back to the USNO, and using software routines to estimate the delay. You can then add the delay correction to the timecode before you set the clock. However, the results will be less accurate than those obtained with ACTS.

Getting Time through the Internet

You can also set your computer clock to NIST time if your PC is connected to the Internet computer network. The Internet serves millions of users. It is an international network that reaches many universities, government agencies, and military installations.

The file server for the Internet time service is located at NIST in Boulder, Colorado. The server is named TIME_A.TIMEFREQ.BLDRDOC.GOV, and its Internet address is 132.163.135.130. Users can access the server and obtain the software and documentation needed (through anonymous FTP) to use the Internet time service. Once you have the software, you can instantly set the clock on any computer that has a TCP/IP Internet connection.

The Internet service provides less accuracy than the modem services, since it is more difficult to estimate network delays than it is to estimate phone line delays. However, this service is well suited for keeping your clock set to the nearest second.

The modem and network services each represent an excellent way to set your PC's clock on time. However, these services will not improve your clock's performance. If you need to keep your PC's clock on the correct second, you may need to access one of these services frequently (perhaps several times a day or more). The following sections describe some alternative ways of keeping your PC's clock on time.

The Smart Clock Technology

A recent development that should have a huge impact on computer timekeeping is the Smart Clock technology invented at NIST (both foreign and domestic patents were applied for in 1992). Although the specifics of Smart Clock technology can get quite involved, the basic premise is simple. If a clock knows the rate at which it gains or loses time, it can correct itself. Clocks usually drift at about the same rate from day to day. For example, if you call ACTS at the same time each day, you might find that your PC's clock is always fast by 4 seconds. Using this information, a memory resident software program could gradually move the time back 4 seconds per day (1/6 second every hour, for instance). Used in conjunction with a time-setting service like ACTS, a Smart Clock algorithm could make a computer clock keep very accurate, self-correcting time. [4]

PC software based on a concept similar to the Smart Clock is starting to appear. This memory resident software will automatically adjust the PC clock to keep it on time (within 0.5 seconds per week). Of course, you'll still have set the clock occasionally (using ACTS or another service) so the software can "learn" about the PC clock's performance and guarantee that the clock stays on the right second. However, this software can greatly improve your PC's ability to stay on time. [5]

Using a Radio Clock

One of the shortcomings of modem services like ACTS is that they require making a phone call (often long distance) each time you need to set your clock. If you need to keep your PC's clock on the right second all the time, this could involve a substantial number of phone calls (and substantial expenses). You may find that your application demands continuous access to an accurate time source, without making phone calls.

You can get accurate time, all the time, by using a radio clock. There are a number of different types of radio clocks that receive digital time codes transmitted by radio from several different time and frequency stations. The costs vary widely, from less than $500 to $5000 or more. Radio clocks come in several different forms. Some are standalone devices with a digital time display. These are often interfaced to the PC (or to other types of computers) using an interface like the RS-232, RS-422, or IEEE-488. Others are available on plug-in expansion cards that work on the PC or AT bus. Your application can use a radio clock to constantly set the PC clock, or it can get all of its timing information from the radio clock and ignore the PC clock entirely.

All radio clocks provide enough accuracy for PC timekeeping (most provide more accuracy than ACTS). You need to make sure, however, that the radio signal of the clock you choose is usable in your area. You also need to make sure that you can mount the type of antenna you need to receive the radio signal. Many radio clocks require an outdoor antenna to operate properly.

Radio clocks are designed to receive and decode a specific radio time signal. There are at least four different time signals you can use, but not all of them are usable in all areas. All four signals are described briefly in the following paragraphs.

The most common (and lowest priced) radio clocks receive signals from NIST radio stations WWV and WWVH. WWV is located in Colorado, and WWVH is in Hawaii. You can tune in to either station with an ordinary shortwave radio (you'll hear the closest one) on 2.5, 5, 10, or 15 MHz. You can also hear WWV on 20 MHz. During the broadcast, you'll hear audio tones that sound like the ticking of a clock. At the beginning of each minute, a voice announces the current time. You can use this information to set your PC's clock manually, if you wish.

In addition to the voice information, WWV and WWVH also broadcast a binary coded decimal (BCD) timecode on a 100 Hz subcarrier that can be read and decoded by a radio clock. The timecode provides the current hour, minute, second, month, day, year, and other information. The time is accurate to within 1 to 50 milliseconds.

WWV and WWVH radio clocks are an economical source of PC time. The signals cover a large area and are usable throughout the United States (and in many other countries). However, they have some shortcomings. Since they use shortwave radio signals, reception may become difficult during some parts of the day. For example, it may be difficult to receive a 15 MHz signal 24 hours a day in some locations. Some radio clocks get around this problem by taking advantage of the fact that WWV and WWVH broadcast on several frequencies. They scan these frequencies and use the one that currently provides the best reception. Another potential shortcoming with using a WWV/WWVH radio clock is that a large, outdoor antenna may be required to get good reception.

Another type of radio clock receives signals from WWVB, a NIST radio station located in Colorado. WWVB is a low frequency (LF) station that broadcasts on 60 kHz. There is nothing to listen to on WWVB, since no voice announcements are made. However, WWVB broadcasts a timecode capable of 0.1 millisecond accuracy. The coverage area of WWVB is smaller than that of WWV/WWVH. However, you can receive WWVB signals in nearly all parts of the continental United States with only a small antenna and equipment that is easy to set up and use.

You can also get NIST time from the GOES satellites operated by NOAA (The National Oceanic and Atmospheric Administration). GOES stands for Geostationary Operational Environmental Satellite. GOES broadcasts from two different satellites on a frequency of about 468 MHz. Only a small antenna is necessary to receive the signals, which are usable in most parts of North and South America. The signal includes a timecode accurate to about 0.1 milliseconds. [6]

The most accurate radio clocks currently available receive signals from the GPS (Global Positioning System) satellites. GPS is a satellite navigation system developed by the U. S. Department of Defense that provides worldwide coverage. Using only a small outdoor antenna, a GPS radio clock can receive time accurate to within less than 1 microsecond, or over 1000 times the accuracy you can get from ACTS. [7]

The price of GPS radio clocks has fallen dramatically over the past few years, and more and more people are using them as timing sources for computer and telecommunication networks. Many new GPS products have been released recently, ranging from handheld navigation receivers to receivers that plug in to the PC or AT expansion bus. In fact, a recent survey lists over 50 manufacturers of GPS receivers. [8] Although not all GPS receivers are suitable for PC timing applications, they should be the dominant type of radio clock for years to come.

Using a More Stable Clock

You can also keep accurate time on a PC by replacing its clock with a better clock. We mentioned that the hardware clock on a PC uses a very low cost (less than $1) timebase oscillator. If you use a clock with a higher quality timebase oscillator, you can obviously keep better time.

Precision real time clock boards are available that plug into the PC bus and effectively replace the hardware clock. These boards often include a better timebase oscillator than the one inside the PC, plus they allow you to use an external oscillator to get even better results. This means that if you have access to a frequency standard (like a quartz, rubidium, or cesium oscillator) you can use it as the timebase.

Precision clock boards greatly increase the stability of the PC clock, but you will still need to set the clock on time (using ACTS or another service) and check it occasionally. However, if you use a good oscillator as a timebase and the system is undisturbed, these boards will keep the correct time for many years. In fact, you may never need to set the PC clock again!

Precision clock boards are expensive, costing about $1000-$2000 for the board and perhaps thousands of dollars more for a frequency standard. However, their potential performance is unequaled. If you work in a laboratory or industrial setting where a frequency standard is already available, they may be the method of choice for getting accurate time on a PC.

Summary

As we have seen in this discussion, you no longer need to be satisfied with the marginal timekeeping abilities of your PC clock. There are a large number of hardware and software solutions you can use to keep accurate time on a PC.

References

[1] Levine, Judah, Weiss, Marc A., Davis, Dick D., Allan, David W., and Sullivan, Donald. B., "The NIST Automated Computer Time Service", Journal of Research of the National Institute of Standards and Technology, Volume 94, Number 5, September-October 1989.

[2] Deurbrouck, John, "Keeping Your PC's Clock Right to The Second with WTIME", PC Magazine, Volume 11, Number 18, October 27, 1992, pp. 361-366.

[3] Winer, Ethan and Cramer, Phil, PC Magazine BASIC Lab Notes, Ziff-Davis Press: Emeryville, CA, 1993, pp. 21-40.

[4] Weiss, Marc A., Allan, David W., Davis, Dick D., and Levine, Judah, "Smart Clock: A New Time", IEEE Transactions on Instrumentation and Measurement, Volume 41, Number 6, December 1992, pp. 915-919.

[5] Becker, G. Thomas, "RighTime, A Real Time Clock Correcting Program for MS-DOS Based Computer Systems", Proc. 24th Annual Precise Time and Time Interval Meeting, December 1992.

[6] Beehler, Roger E. and Lombardi, Michael A., "NIST Time and Frequency Services", NIST Special Publication 432, U. S. Government Printing Office, June 1991.

[7] Kamas, George and Lombardi, Michael A., "Time and Frequency User's Manual", NIST Special Publication 559, U. S. Government Printing Office, September 1990.

[8] "GPS World Receiver Survey", GPS World, Volume 4, Number 1, January 1993, pp. 52-65.

This paper was previously published in an abbreviated and slightly altered form:

Lombardi, Michael A., "Keeping Time on Your PC", BYTE Magazine, Volume 18, Number 11, October 1993.

COMPUTER TIME MANUFACTURERS LIST

Although we have made every effort to include all manufacturers, the following list may be incomplete. Inclusion on the list implies no endorsement by NIST. If you know of other products that should be included on future lists, please contact NIST.

Time Setting Software

ACTS Software
NIST Office of Standard Reference Materials
B311 Chemistry Building
NIST
Gaithersburg, MD 20899-0001
(301) 975-6776
(calls ACTS, includes source code)

TIMECHECKER
Zephyr Services
1900 Murray Avenue
Pittsburgh, PA 15217
(800) 533-6666
(calls USNO)

TIMESET
Life Sciences Software
8925 271st N.W., Suite 112
P.O. Box 1560
Stanwood, Washington 98292
(206) 387-9788
(calls ACTS or USNO)

PCCLOCK Digital Clock
M. J. Sadaway
Box 128-P
South Walpole, MA 02071
(calls ACTS)

TIME-SYNC
SolaCamp Software
P. O. Box 4464
Lutherville, MD 21094
(calls ACTS or USNO)

WTIME
Katherine West, Utilities
PC Magazine
One Park Avenue
New York, NY 10016
FAX: (212) 503-5799
(calls ACTS, for Windows)

Self Correcting Clock Software

RIGHTIME
Air System Technologies, Inc.
14232 Marsh Lane, Suite 339
Dallas, TX 75234-3899
(214) 402-9660

PRECISION TIME
CrystalLogic
2525 Perimeter Place Drive, Suite 121
Nashville, TN 37214
(800) 915-6442
(615) 391-9100
(also calls ACTS and USNO)

Radio Clock Manufacturers (for WWV/WWVH, WWVB and GOES)

Spectracom Corporation
101 Despatch Drive
East Rochester, NY 14445
(716) 381-4827
(WWVB)

CHRONO-LOG Corporation
2 West Park Road
Havertown, PA 19083-4691
(215) 853-1130
(WWV)

Franklin Instrument Company
233 Railroad Drive
Warminster, PA 18974
(215) 355-7942
(WWVB)

Odetics
Precision Time Division
1515 South Manchester Avenue
Anaheim, CA 92802-2907
(714) 758-0400
(WWV, also GPS)

Radio Clock Manufacturers (continued)

TrueTime
3243 Santa Rosa Avenue
Santa Rosa, CA 95407
(707) 528-1230
(WWV, WWVB, GOES, also GPS)

Arbiter Systems, Inc.
1324 Vendels Circle, Suite 121
Paso Robles, CA 93446
(805) 232-3831
(GOES)

Precision Clock Boards

Guide Technology
920 Saratoga Avenue, Suite 215
San Jose, CA 95129
(408) 246-9905

Bancomm
Division of Datum Inc.
541 Via del Oro
San Jose, CA 95119
(408) 578-4161