By Brooks Shera, W5OJM

A GPS-Based Frequency Standard

This modern and highly accurate frequency standard is something *you* can readily have!

F requency accuracy has been a topic of special interest to many amateurs and experimenters since the early days of radio. Until recently, the best frequency standard available to most hams was a crystal oscillator carefully adjusted to zero-beat with a station of known frequency, such as WWV. Unpredictable variations in ionospheric propagation make achieving high accuracy by this method an art as well as a science. In fact, until 1981, the ARRL sponsored frequency measuring tests.[1] Only the best entries were closer than 0.1 ppm. This is plenty good enough to keep your transmitter within the HF band edges, but for applications such as EME work and VHF weak-signal detection, it leaves something to be desired.[2]

Today, the potential for accurately measuring frequency is vastly improved. We now have more than two dozen atomic clocks constantly circling the Earth and typically six or more of these are in line-of-sight view all the time! I refer, of course, to the GPS satellites kindly provided by the Department of Defense (DoD). The spread-spectrum signals from these satellites are very weak and quite complex. However, the hard work necessary to receive and process these highly accurate frequency signals has already been done by the folks that design and build the little GPS navigation receivers that are now available for less than the price of a hand-held transceiver.

These GPS receivers process the signals from four or more satellites, and once every second they compute latitude, longitude and altitude. To make this computation, the receiver must also compute the current time with very high accuracy. As a side benefit of the position computation, many receivers output a timing pulse once each second for the use of people who like to know *exactly* what time it is.

Perhaps we can use this once-per-second pulse, which is derived from the orbiting atomic clocks and is typically accurate to a few tens of nanoseconds (ns), to con-

trol (discipline) the frequency of our Earth-based frequency standard in much the same way that previous generations of hams have manually adjusted crystal oscillators to WWV. If so, we might expect to achieve an accuracy of perhaps a few parts in 10^{11}, about 10,000 times better than the time-honored WWV zero beat method!

The challenge of this project is to use the GPS timing pulse—which occurs only once per second—to control the frequency of a crystal oscillator that vibrates perhaps five million times a second. The direct approach I chose was to use a phase-locked loop (PLL). Because of the long time constants involved, I built the loop using digital—rather than the traditional analog—components.

Circuit Description

As shown in the block diagram of Figure 1, the device consists of five sections: a commercial GPS receiver, a voltage-controlled crystal oscillator (VCXO), a phase-measuring circuit, a microprocessor (CPU) with a few interface chips, and a digital-to-analog converter (DAC) to control the

VCXO. Figure 2 shows the controller board, which includes almost everything except the GPS receiver and the VCXO. Figure 3 is the schematic of the controller.

VCXO

A stable, temperature-controlled oscillator is desirable because we will rely on it to keep the output on frequency between GPS pulses. Good oscillator stability also helps us overcome the slight jitter in the time of the GPS pulses that is purposely introduced by the DoD to reduce the navigational accuracy of GPS for nonmilitary users. High-quality VCXOs are widely used in cellular-telephone transmitter sites. These VCXOs typically oscillate in the 5 to 10 MHz range and are often available on the surplus market. Excellent temperature-controlled crystal oscillators have also been used for many years as components in better-quality frequency counters. One example is the HP 10811 crystal oscillator subassembly, which is often available at a reasonable price.[3] The controller circuit can accommodate any stable oscillator in the range of 0.4 to 10 MHz, but be certain you
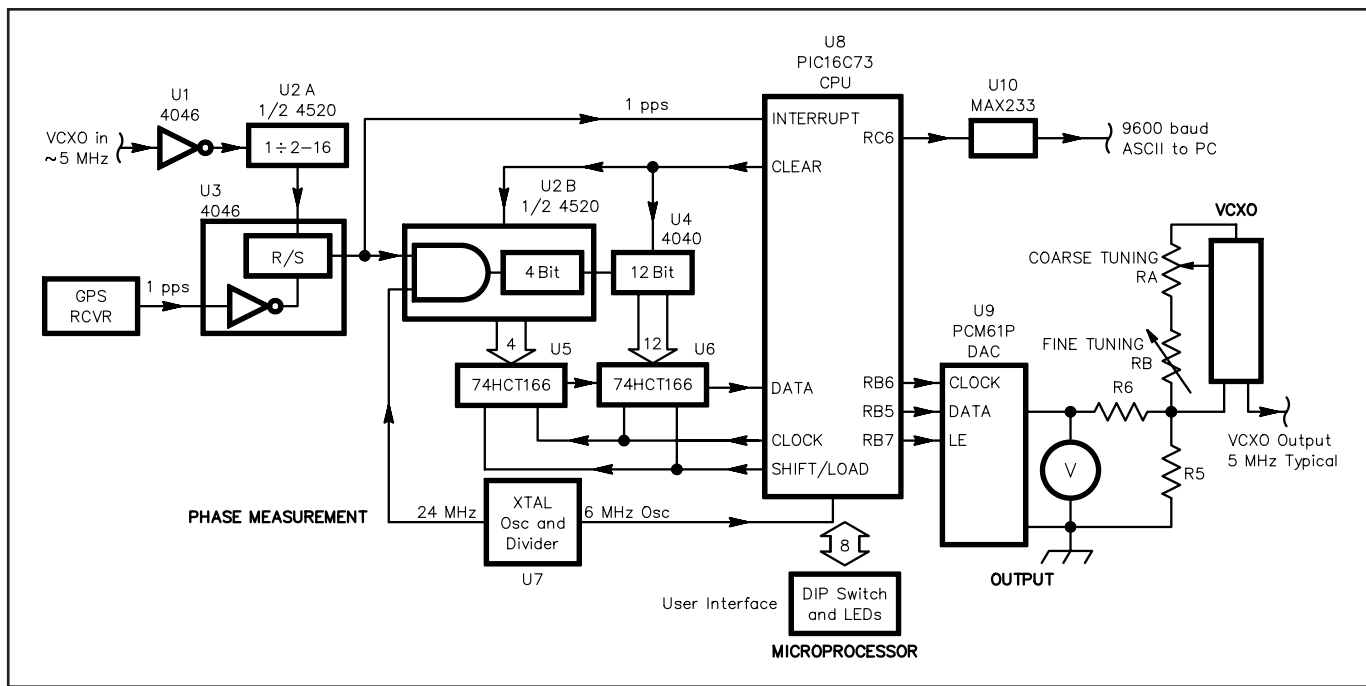
Figure 1—Block diagram of the GPS-based frequency standard.

get a unit that has a voltage-control input that can vary the frequency over a narrow range. In the following discussion, I will assume we are using a 5 MHz VCXO.

Phase Detector

To control the VCXO, we need to measure the phase difference between its output and the 1 pps GPS pulses. This can be done with less ambiguity if the measurement is done at a frequency lower than 5 MHz, say 300 kHz. Therefore, after amplification and buffering by U1 (see Figure 1), the 5 MHz signal from the VCXO is divided by 16 in a 4-bit counter, U2A, to produce an output near 312 kHz. The exact frequency is unimportant. It is only necessary that it be phase-related to the VCXO. The phase difference between the GPS and the VCXO is measured by counting the number of pulses (U2B/U4) from a separate, garden-variety 24 MHz crystal oscillator (U7) that occur during the time between the GPS pulse and the *next* VCXO pulse from U2A. Each count of the 24 MHz signal indicates a phase difference of 42 ns. If this count is constant, we know the phase difference is constant, and thus, that our LO is synchronized to the GPS atomic clocks.

Interestingly, it is desirable to have the frequency of U7 drift slightly rather than being synchronized with the VCXO. A slight random drift averages out the ±1 count ambiguity that is inherent in any pulse-counting device. My measurements indicate that the simple phase-measuring circuit I use is consistently accurate to 2 or 3 ns (for a 30-second measurement), while without drift, the resolution would be limited to 42 ns. The $5 crystal oscillator mod-

ule drifts adequately and also provides a divided 6 MHz output to clock the CPU.

DAC

It is the microcontroller's (U8) job to read the count from U2B/U4 and adjust the control voltage applied to the VCXO to keep this count constant. U8 controls the VCXO by sending a message to the DAC (U9). The DAC I chose is a relatively low-cost 18-bit unit designed for digital audio applications such as high-quality CD players. It can control the VCXO over a range of ±3 V in steps as small as 23 µV. The DAC output voltage is attenuated by the resistor network R6 and R5 before it is applied to the VCXO control input. The resistor network serves to further decrease the voltage step size and provides a convenient way to adapt the controller circuit to VCXOs that have different control-input sensitivities.

CPU

The CPU is the brains of the controller and its interfacing and software dominated the design effort that I devoted to this project. The inexpensive PIC16C73 micro I chose is quite versatile but, like most small micros, it has a limited number of input/output pins. Therefore, I have used serial communications between the PIC and the other controller components. The '16C73 is well suited for such communications because it includes two built-in serial ports. One of these ports is used to send ASCII messages to an external PC so the performance of the VCXO and the PLL can be monitored. (This ASCII port also made debugging the hardware and software much easier.)

Serial communication is also used to

read the phase count value from U2A/U4. Whenever the CPU needs to get the current phase, it sends a load pulse to the parallel-in/serial-out shift registers, U5 and U6. This pulse loads the count data from U2A/U4 into the shift registers, and a reset pulse is then sent to clear the counters so they commence to count from zero when the next GPS pulse arrives. The CPU uses its synchronous serial port to upload the phase data from the shift registers.

The PCM-61A DAC (U9) is also a serial-input device. The CPU's software manipulates the three output pins that are connected to the DAC to download the data one bit at a time. After a full 18-bit data value is loaded, the DAC automatically latches this value into an internal register and sets its output voltage accordingly. The voltage is held indefinitely, or until a new data value is loaded. The voltage-hold feature makes it possible to arrange the DAC and the VCXO as a separate detachable unit that is connected to the controller only when the VCXO needs to be set on frequency. After setting, the VCXO-DAC unit can be moved to wherever a precise frequency reference is needed.

Two more bits of hardware should be mentioned since their appearance in the circuit may be confusing. The controller uses two '4046 PLL ICs, but neither of them is used for their designed purpose. U1 is used only as a sensitive, high-input-impedance amplifier section to buffer the VCXO, while U3 buffers the GPS input line and supplies a fast RS flip-flop that gates the phase counter. The 4046 is a readily available, inexpensive IC that has several uses and is worth exploring.[4]

Software

As mentioned earlier, the primary task of the CPU is to monitor the phase difference between GPS and the VCXO and respond appropriately. If this phase difference begins to drift, the CPU must make a correction to the VCXO frequency. The tricky part is to make a correction of the *right size*. If the correction is too large, the VCXO frequency will consistently overshoot the mark and jitter around the correct value. Conversely, too small a correction will cause a sluggish, overdamped response. The CPU should also smooth over the small second-to-second and minute-to-minute GPS timing fluctuations, while giving GPS full control of the VCXO frequency in the longer term. This is basically a filtering function, so we need to design software that makes the CPU act like a low-pass digital filter.

Fortunately, design help is available. The theory of analog filters in feedback control systems is described in several reference books,[5] and methods have been developed to implement these designs as digital-computer algorithms.[6] Our feedback control and filtering problem is another aspect of digital-signal processing (DSP). You might reasonably wonder why I chose a little 8-bit microcontroller instead of a DSP processor if we need to do DSP. There are several reasons. First of all, high CPU speed is not required because phase values come only once per second and the program averages 30 seconds of phase data together before computing a new DAC setting. Half a minute gives plenty of time to compute almost anything that could be needed. The 8-bit word size of the microcontroller also proved to be no limitation. The software does highly precise arithmetic simply by stacking five consecutive 8-bit words to form a 40-bit integer. Software routines do all the usual arithmetic operations on these 40-bit words. Lastly, perhaps the most important factor in my CPU choice is that the PIC micro is a lot of fun to program!

The feedback filter I have programmed is the digital equivalent of what control systems specialists call a *PI filter* (proportional integral filter—its response is proportional to the input signal, plus its time integral). You can make an analog version of a PI filter with a high-gain op amp, two resistors and a capacitor, but you might have trouble getting the long integration time we need in this application. The PI filter has several useful and interesting properties. First, because it integrates the input signal, it provides low-pass filtering. Moreover, after a period of operation, the filter learns the average drift rate of its input signal and automatically adjusts its output to correct for it! In our case, such a linear drift might result from a steady change in ambient temperature, or the aging of the VCXO crystal. The PI filter is programmed as a second-order infinite impulse response (IIR) filter, which has the
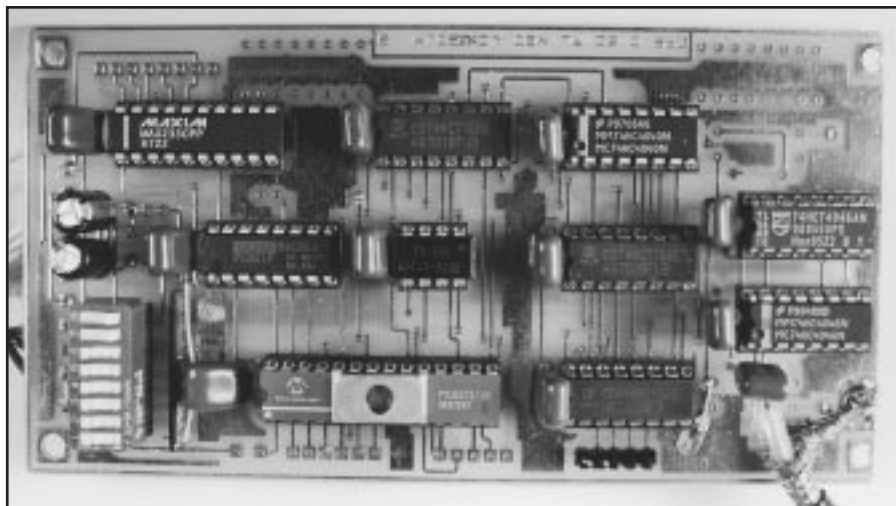
advantage that it requires only a few arithmetic operations and needs to store only three data values: the current input and the previous input and output values.

The software provides six different filter time constants, ranging from no filtering at all, to a time constant of many hours. The time constant—which is chosen by setting S1 through S3 of U11—can be changed on-the-fly while the controller is running without causing a transient that might temporarily perturb the VCXO frequency.[7] The usual procedure is to start with no filtering, which permits the controller to quickly lock the VCXO to GPS and then, over a period of time, select progressively longer constants. The ultimate selection is a balance between reduction of GPS jitter (which needs a long time constant) and elimination of medium and long-term VCXO drift (which favors a shorter time constant). The longest time constants are suitable only for very stable oscillators, such as high-quality oven-controlled crystals and Rubidium (Rb) atomic oscillators. Although Rb oscillators are very stable, even they can be improved by locking them to the GPS Cesium clocks.

In addition to implementing a versatile digital filter, the software provides other useful functions. Each new phase value is checked for consistency against the immediately preceding values by a deglitching algorithm. Momentary phase jumps are discarded before they can affect the VCXO frequency. The CPU also continually monitors the status of the PLL and reports potential error conditions via the front-panel LEDs and the ASCII port, as described in the following sections.
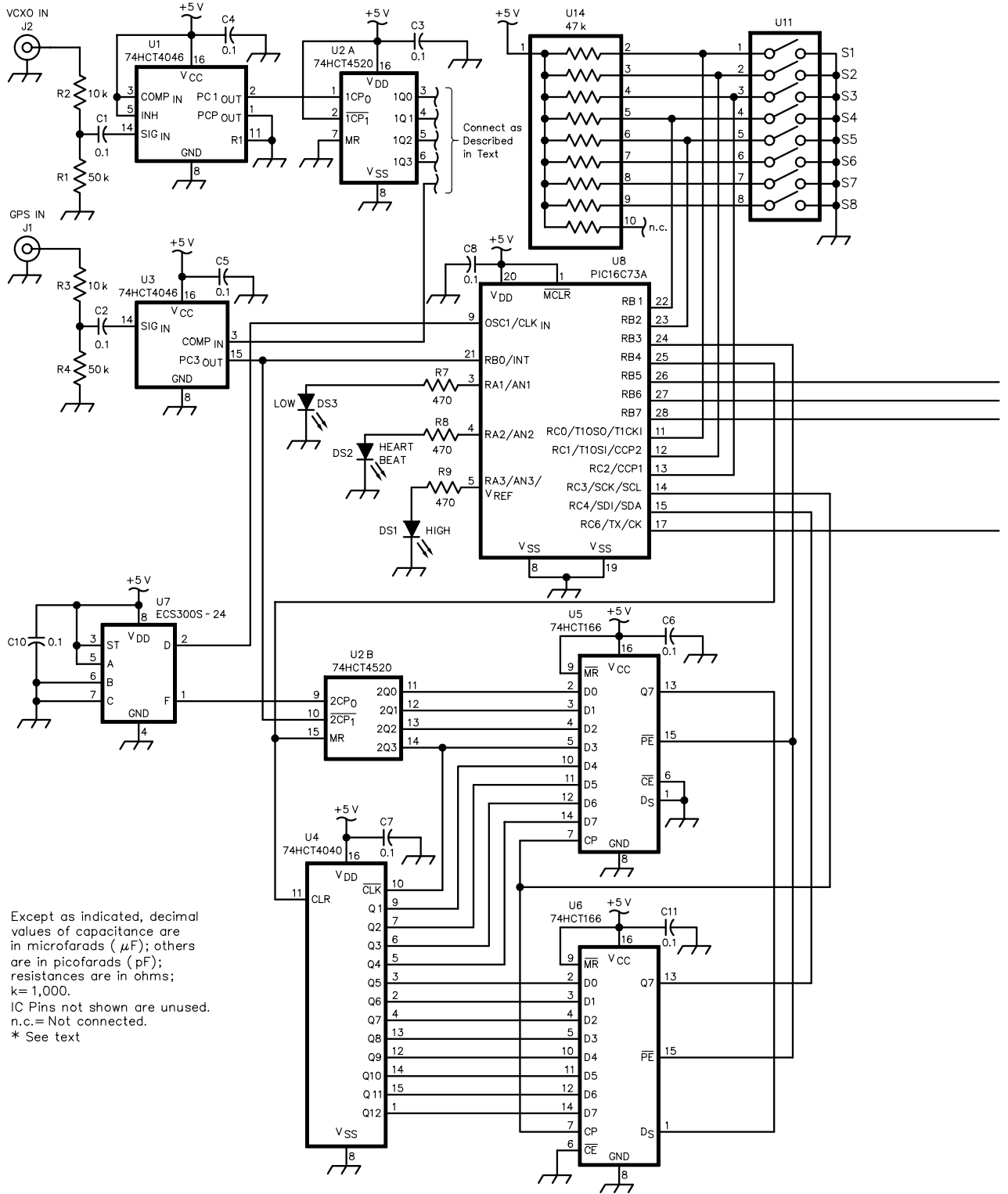
Construction

PC boards are available from A&A Engineering (see Figure 3 caption). In many applications, the controller will run essentially unattended for days, weeks, or months, so it is important to use reliable and safe construction practices and ad-

equate power-supply fusing. A metal enclosure and shielded I/O leads protect the unit from nearby RF fields and reduce EMI from the digital signals generated within. *Use IC sockets.* I installed a digital voltmeter module on the front panel of the prototype to monitor the DAC output voltage. The meter provides a convenient operational check and indicates when VCXO aging may require a manual adjustment of the frequency pot to keep the DAC within its ±3 V operating range. (A test point for use of an external voltmeter can be substituted.)

Setup

VCXO

The circuit can accommodate most any stable VCXO that you want to use. The price of this flexibility is that a few setup steps are required. First, set S4 (U11, pin 4) to correspond to the polarity of the VCXO control voltage. If the VCXO frequency increases when the control voltage increases, open S4. If the frequency decreases with increasing voltage, close S4. Install a jumper to connect U3 pin 3 to one of the four output pins of U2A. U2A divides the input VCXO frequency in binary steps from 2 to 16. Select a pin that provides a VCXO output frequency in the 150 to 700 kHz range. Solder pads are provided at the edge of the PC board to make jumper installation easy. Third, the values of R6 and R5 must be selected[8] to match the sensitivity of the control voltage input of your VCXO. The goal is to obtain a relative frequency change, $\Delta F/F = 7.5 \times 10^{-9}$, when the voltage applied by the DAC to R6 changes by 1 V. Lastly, you should check the values of the input resistors R1 and R2. The values given in Figure 2 should work fine with most VCXOs because they were selected to provide a fairly high input impedance to avoid loading the VCXO output. If you change the value of these resistors, keep in mind that the peak-to-peak voltage at U1 pin 14

Figure 2—The double-sided controller board measures $2^1/_2 \times 5^1/_2$ inches.

VCXO IN
J2

+5 V
C4
0.1

U1
74HCT4046
V CC

R2 10 k
C1
0.1
R1 50 k

3 COMP IN   PC1 OUT 2
5 INH   PCP OUT 1
14 SIG IN   R1 11
GND
8

+5 V
C3
0.1

U2 A
74HCT4520
V DD 16

1 1CP0   1Q0 3
2 1CP1   1Q1 4
7 MR   1Q2 5
         1Q3 6
V SS
8

Connect as
Described
in Text

+5 V
U14
47 k
1   2
3
4
5
6
7
8
9
10 n.c.

U11
1
2
3
4
5
6
7
8

S1
S2
S3
S4
S5
S6
S7
S8

GPS IN
J1

+5 V
C5
0.1

U3
74HCT4046
16
V CC

R3 10 k
C2
0.1
R4 50 k

14 SIG IN
   COMP IN 3
   PC3 OUT 15
GND
8

C8
0.1

+5 V
20

U8
PIC16C73A
V DD   MCLR 1

9 OSC1/CLK IN
21 RB0/INT
4 RA1/AN1
   RA2/AN2
5 RA3/AN3/
   V REF

RB1 22
RB2 23
RB3 24
RB4 25
RB5 26
RB6 27
RB7 28
RC0/T1OSO/T1CKI 11
RC1/T1OSI/CCP2   12
RC2/CCP1   13
RC3/SCK/SCL   14
RC4/SDI/SDA   15
RC6/TX/CK   17
V SS   V SS
8   19

LOW DS3
R7
470

DS2 HEART
     BEAT
R8
470

R9
470

DS1 HIGH

+5 V
U7
ECS300S-24

C10 0.1
3 ST   V DD D 2
5 A
6 B
7 C   GND F 1
8
4

U2 B
74HCT4520

9   2CP0   2Q0 11
10 2CP1   2Q1 12
15 MR   2Q2 13
          2Q3 14

U5
74HCT166 +5 V
9 MR   16 V CC
C6
0.1
2 D0   Q7 13
3 D1
4 D2   PE 15
5 D3
10 D4   CE 6
11 D5   DS 1
12 D6
14 D7
7 CP   GND
        8

U4 +5 V
74HCT4040 16 C7
         V DD 0.1
11 CLR
10 CLK
9 Q1
7 Q2
6 Q3
5 Q4
3 Q5
2 Q6
4 Q7
13 Q8
12 Q9
14 Q10
15 Q11
1 Q12
V SS
8

U6
74HCT166 +5 V
9 MR   16 V CC
C11
0.1
2 D0   Q7 13
3 D1
4 D2   PE 15
5 D3
10 D4
11 D5   DS 1
12 D6
14 D7
7 CP   GND
6 CE   8

Except as indicated, decimal
values of capacitance are
in microfarads (μF); others
are in picofarads (pF);
resistances are in ohms;
k= 1,000.
IC Pins not shown are unused.
n.c.= Not connected.
* See text
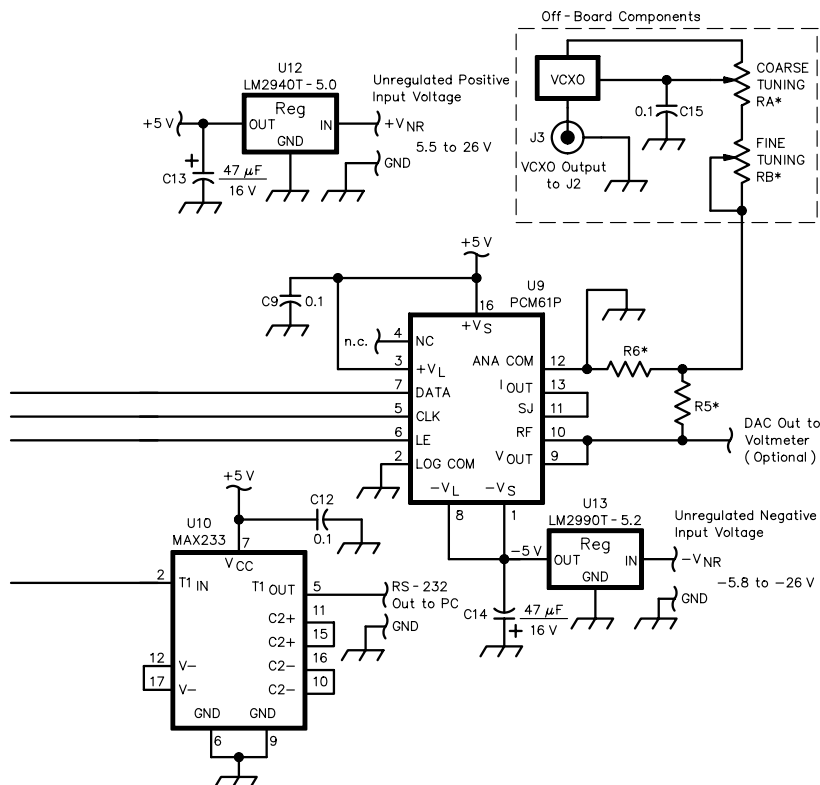
Figure 3— Schematic of the controller. Unless otherwise specified, resistors are ¼ W, 5% tolerance carbon-composition or film units. Equivalent parts can be substituted. The controller is constructed on a double-sided PC board measuring approximately 2.75×5.25 inches. A controller PC board is available from A&A Engineering, 2521 W La Palma Ave Unit K, Anaheim, CA 92801; tel 714-952-2114, fax 714-952-3280; stock no. 217-PCB, $19.95 plus $1.50 shipping and handling per order; foreign orders, shipping and handling is 15% of the total order price for surface mail. Programmed PICs are available from the author for $22 each plus $3 shipping in the US and Canada, $7 elsewhere. The source code for the controller software is available from the author, see Note 14. All the controller parts are available from Digi-Key Corp, 701 Brooks Ave S, Thief River Falls, MN 56701-0677; tel 800-344-4539, 218-681-6674, fax 218-681-3380 **http://www.digikey.com** and Radio Shack. For part numbers in parentheses, DK = Digi-Key; RS = Radio Shack.

C1-C12, C15—0.1 µF, 50 V polystyrene
   (DK P4593)
C13, C14—47 µF, 16 V electrolytic
   (DK P1210)
DS1-DS3—LED (RS 276-069)
U1, U3—74HCT4046 (RS 276-2913)
U2—74HCT4520 (DK CD74HCT4520E)
U4—74HC4040 (DK MM74HC4040)
U5, U6—74HCT166 (DK 74HCT166E)
U7—24 MHz oscillator module ECS300S-24 (DK XC-313)
U8—*Programmed* PIC 16C73A; *unprogrammed* devices are available from Digi-Key
   (DK PIC16C73A-20/SP)
U9—PCM61P (DK PCM61P)
U10—MAX233 (DK MAX233CPP)
U11—8-pole DIP switch (DK A5208)
U12—LM2940 (DK LM2940T-5.0)
U13—LM2990 (DK LM2990T-5.2)
U14—47 kΩ, 9-resistor SIP (DK Q9473)
Misc: PC board (see Note 14), enclosure (Elma type 33 [DK 260-1001] used here), optional voltmeter (DK JDPM601; less expensive units will suffice), IC sockets, connectors and hardware.

should be in the range 50 mV to 2 V. If your VCXO provides a low-impedance TTL output, bypass the input network (including C1) and connect directly to U1 pin 14.

GPS Receiver

Several GPS receiver models are available that provide a 1-pps timing pulse. This controller was developed using a Motorola PVT-6, which has been recently superseded by the Motorola Oncore VP. The Garmin GPS-25 and the Trimble SK-8 also provide 1 pps pulses.[9] The best timing results are usually obtained when the receiver is used in "position hold" mode. Basically, the idea is to let the receiver determine its location, then manually lock that location into the software so that timing is the only variable the receiver needs to consider. Consult your receiver manual on how to set up this mode. The receivers typically provide a standard TTL signal that can be connected directly to the input at U3 pin 14, however the discussion earlier regarding the VCXO input network also applies here since both inputs feed 4046s. U3 triggers on the positive rising edge of the input and the circuitry assumes that this edge provides the GPS time reference. The GPS receiver's CPU clock introduces a granularity in the timing pulse output, but the effect of this is greatly reduced—along with jitter from other sources—by the 30-second averaging and by the low-pass filter.

Controller

In addition to the six time constants mentioned earlier, the controller provides a setup mode to help you make a coarse adjustment of VCXO frequency that places the desired lock frequency near the center of the voltage control range. Depending on the VCXO used, the manual adjustment could be a trimmer capacitor or a control-voltage trim pot. The controller's LED status lights **HIGH** and **LOW** indicate when the frequency is too high or too low. Select setup mode (U11 S1, S2, S3 all closed) and carefully adjust the VCXO frequency until the two high/low lights are extinguished. Small adjustments and a little patience are needed because the status lights are updated only at the 30-second intervals when the phase-measuring circuit produces a new reading. When the lights remain off for several minutes, switch to run mode and proceed to progressively lengthen the time constant as described later. A third status light (**HEARTBEAT**) blinks once per second to indicate that the GPS pulses are being received and that the controller CPU is running. It may be helpful, although not essential, to monitor the controller's performance by connecting a PC to the controller's ASCII port as described next.

Performance

The controller is programmed to send the GPS-VCXO phase-difference data (averaged over 30-second intervals) to its
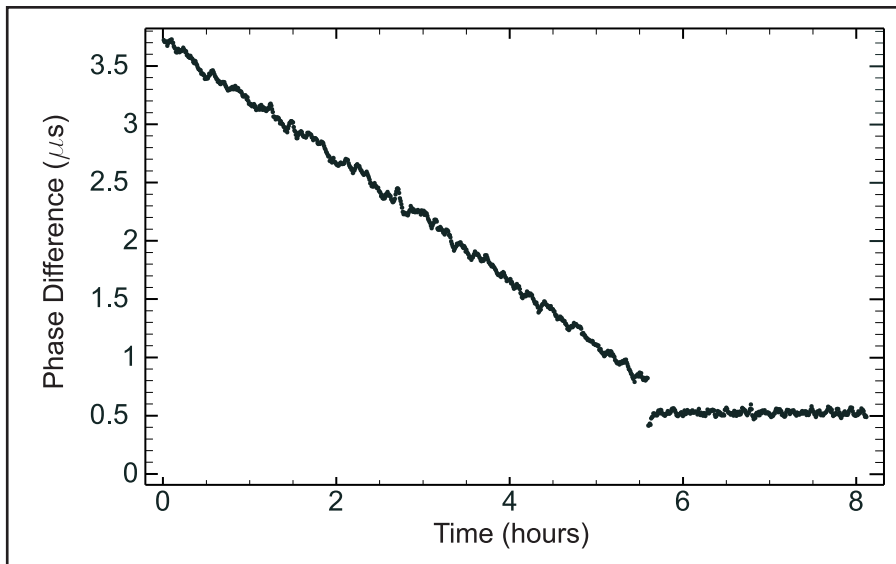
Figure 4—Locking an LO to the GPS clocks. Initially, the phase between GPS and the oscillator changes with time, indicating that the oscillator is slightly off frequency. After several hours, the controller feedback loop was closed and the oscillator becomes locked to the GPS atomic clocks!
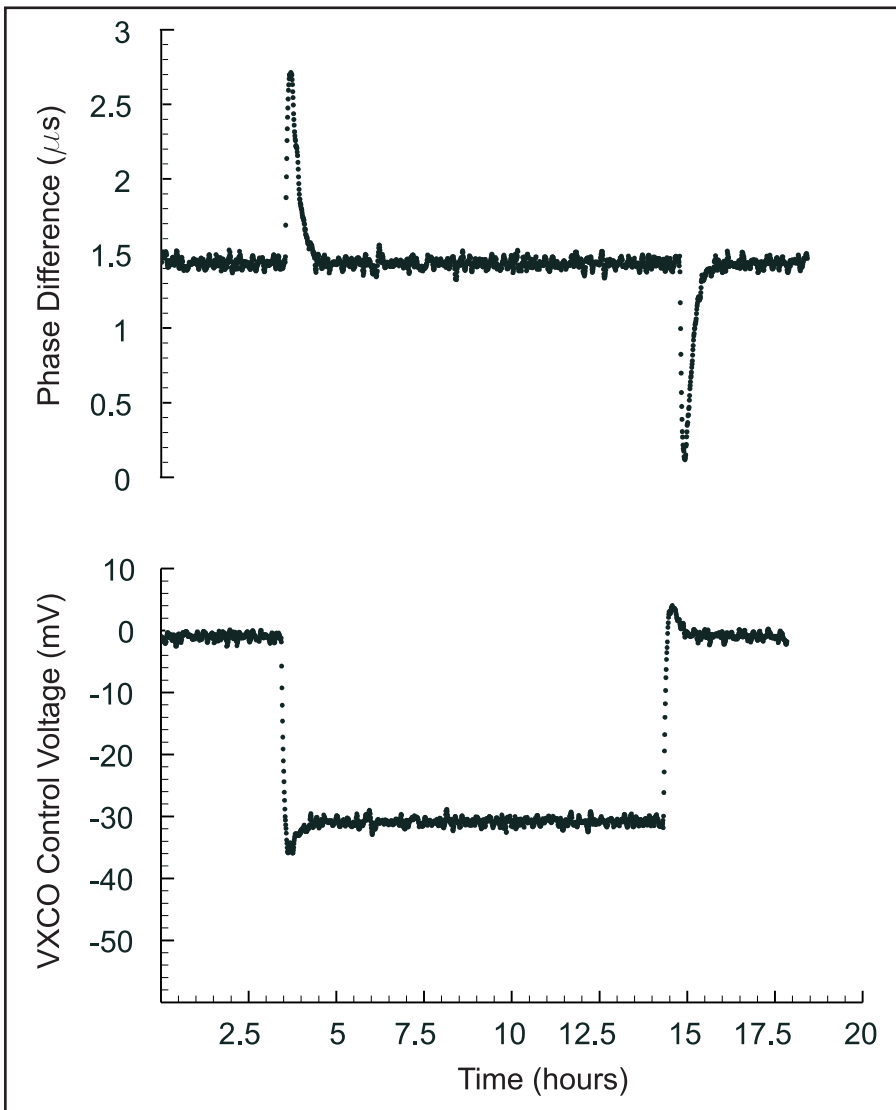

Figure 5—Dynamic behavior of the controller. Rapid response and absence of ringing are displayed when the controller is forced to correct a manual frequency offset (see text).

ASCII output where it can be captured by a PC for analysis.[10] Data from the ASCII port is quite revealing, especially when accumulated over a period of a few hours, and with the PLL feedback loop both open and closed.[11] Even when the PLL is locked, the phase data shows random noisy fluctuations from one 30-second averaging period to the next that have a standard deviation of about 35 ns. I call this "GPS jitter," and its primary cause is presumably the intentional DoD dithering of the GPS signals.[12] It is not evident when I monitor the phase difference between the VCXO and a second stable crystal oscillator that has been divided to generate 1 pps pulses. If this jitter is passed on directly to the VCXO by the PLL, it would induce short-term frequency shifts of about one part in 10^9. Although such performance isn't too bad, it can be improved by low-pass filtering. How much filtering improvement can be achieved depends on the VCXO stability, which itself can be estimated from open-loop phase observations. As programmed, the loop filter has a slope of 6 dB/octave, so each doubling of the time constant cuts the GPS jitter in half. The maximum filter attenuation is about 40 dB, but as mentioned earlier, this can only be used effectively with a highly stable oscillator. A good oven-controlled VCXO should be able to operate with a time constant that will attenuate the jitter by at least 30 dB, and should yield a frequency accuracy of a few parts in 10^{11}.

The drift rate of a high-stability oscillator is conventionally specified by computing the Allan variance (AV). The AV is a measure of the fractional difference in frequency we would expect to observe if two measurements of the oscillator frequency were made separated by a time, T. If the time between the two measurements is long, the AV of our disciplined oscillator will be essentially the same as the long-term stability of the GPS atomic clocks. A typical value for T = 24 hours would be AV = 2×10^{-13}. If the time between the measurements is shorter—in the range of a few minutes to a few hours—the AV is determined by the stability of our VCXO and the CPU filter time constant. For a good-quality VCXO, we can expect to obtain an AV of perhaps 5×10^{-12} for T = 10 min.[13]

The best way to judge your disciplined VCXO is to compare it with a hydrogen maser, but lacking that, a lot can be inferred from the ASCII data provided by the controller. To select the time constant that gives the optimum performance from a particular VCXO, attach a PC to the controller's ASCII port and record the data for awhile. Locking an oscillator to a short-term-noisy, long-term-accurate reference like GPS is a relatively unexplored field. I encourage you to experiment. My assembly-language PIC software is available as a starting point if you want to try different filtering techniques.[14] The code is highly commented and the *free* software program-

ming tools you need are available on the Internet.[15]

Figure 4 is an illustration of phase-locking to the GPS signal. Initially, the VCXO was allowed to run open-loop, ie, not under GPS control. Then, after several hours, the loop is closed. The plot shows the phase difference (expressed in time units) between the VCXO and the GPS pulses. The phase difference slowly decreases, indicating that the VCXO frequency is a little too high. From the slope of the plot, we can estimate that it is too high by about two parts in 10^9, or 0.01 Hz for our 5 MHz VCXO. Note that the plot is quite linear, indicating that although the frequency is off a bit, it is quite constant—the trademark of a stable VCXO. Note also the noise on the plot caused by GPS jitter. After about 5.5 hours, I flipped U11 S5 to allow the DAC to control the VCXO thereby closing the loop. After a short transient, the plot abruptly becomes horizontal indicating that the phase difference is now constant and that the VCXO is locked to the GPS atomic clocks. It was an exciting moment when I first watched this happen!

The closed-loop transient response of the controller is shown in Figure 5. The upper plot shows phase difference data like that of Figure 4, and the lower plot is the DAC output voltage fed to the VCXO. The two curves represent the input and output of the digital filter. About three hours into the run, I manually offset the frequency of the VCXO by 0.035 Hz, forcing the controller to make a large correction. The phase difference begins to increase and the controller responds by decreasing the DAC output. After about 45 minutes, the transient is complete and the VCXO has been disciplined back onto frequency as shown (upper plot) by the phase difference returning to the setpoint. The lower plot shows the DAC voltage output that was needed to correct the manual frequency offset. A shorter filter time constant would make the response faster, but at the cost of greater GPS jitter. After about 14 hours, I removed the manual offset and the transient is repeated, but with the opposite polarity. The clean response of the controller and the small nearly optimal overshoot is evident for both transients.

Figure 6 shows the DAC output voltage during normal operation of the controller. The data were accumulated over a period of several days by recording the output from the ASCII port. Several interesting things can be observed. There is a slow decrease in voltage, which corrects the long-term aging of the VCXO crystal. Converting the VCXO input voltage to relative frequency indicates that the aging process is about 7×10^{-12} per day; which is typical of a good quality VCXO that was manufactured a few years ago and has had a while to stabilize. Superimposed on the slow decrease is a nearly sinusoidal 24-hour variation caused by a day-night temperature shift of about
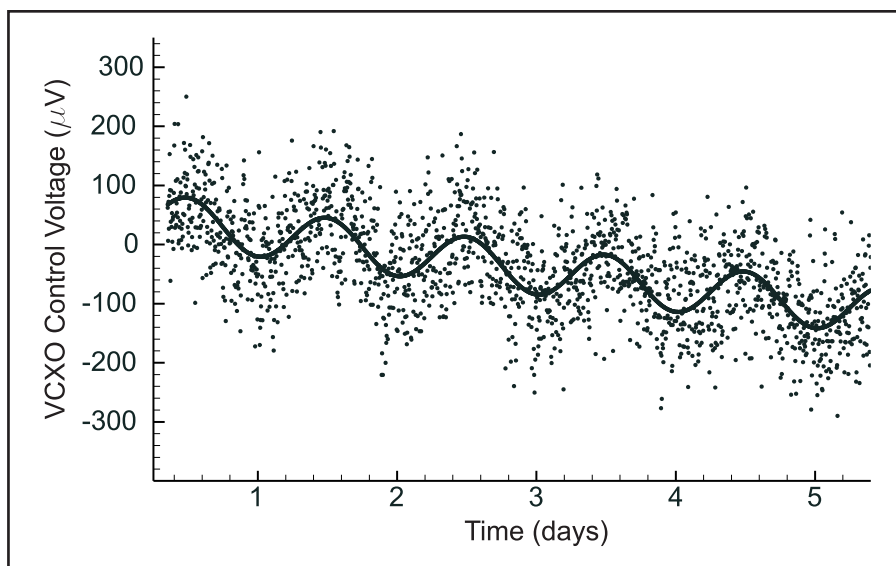

Figure 6—Normal operation of the controller. A day-night temperature effect and the long-term aging of the VCXO crystal oscillator are evident in this plot of the correction voltage applied to the VCXO by the controller during normal operation. The controller adjusts the VCXO input voltage as needed to eliminate the frequency shifts that would otherwise occur if there were no GPS stabilization. The data point scattering is primarily caused by residual GPS jitter that is present even after low-pass filtering. The solid line is a curve fitted to the data to obtain the aging and temperature parameters.

4°C in my workshop. The data indicate a temperature coefficient of about 4×10^{-12} per $^\circ$C, a respectable value for an ovenized crystal. The GPS phase lock has prevented these normal aging and temperature effects from significantly affecting the VCXO frequency. GPS jitter is also quite evident as noise on the plot. The RMS jitter amplitude has been reduced from about 35 ns at the filter input to about 0.5 ns equivalent at the output. Still, the noise dominates the plot and suggests that an even-longer time constant could be used with this VCXO to further improve the short-term stability of the output frequency.

Figure 6 also suggests that two major causes of frequency instability—temperature shift and aging—could be predicted and largely eliminated by tracking the performance of the VCXO for a while to estimate the aging parameters and by measuring the ambient temperature. The predicted corrections could be applied to the VCXO independently of the PLL, which might allow much longer loop filtering time constants to be used, further reducing GPS jitter. Although this scheme would be ultimately limited by sources of crystal frequency instability that are random and inherently unpredictable, it might be interesting to explore. Only minor additions to the hardware would be needed since the PIC microprocessor has an unused ADC input that is available for temperature monitoring.

A Demonstration

As a demonstration, I tested the controller by disciplining the master oscillator (a 10811) in an HP 5328A frequency counter (see the title photo). The 1-MHz frequency output on the rear panel of the '5328A was connected to the controller VCXO input and a pair of wires was added to carry the DAC feedback voltage to the master-oscillator board inside the HP counter. After only a few moments of operation, and a little adjustment of its master-oscillator frequency pot, the HP5328A was locked onto the GPS. The result was a counter with an accuracy and stability far surpassing anything that could have been imagined by Hewlett-Packard for the 5328A when they built this counter nearly 20 years ago.

Notes

[1] The end of the almost 50-year span of ARRL-sponsored frequency measuring tests was announced in "Operating News," *QST*, Dec 1981, p 99—*Ed.*
[2] For an interesting discussion of ultra-weak signal detection and the importance of high frequency accuracy, see the article by Darrel Emerson, AA7FV "The Mars Global Survey Relay Test," *AMSAT Journal*, Volume 20, No 1, Jan/Feb 1997, or visit Darrel's Web site at **http://www.tuc.nrao.edu/~demerson/marsspec/marsspec.htm**. SETI information can be found at **http://seti1.setileague.org** and interesting EME work has been reported by Michael Cook, AF9Y (**http://www.webcom.com/af9y**).
[3] A possible source of HP10811s is Gary Glassmeyer, N6ZD, at Certified Used Test Equipment, Lockwood CA; tel 408-385-0301; also Douglas Dwyer e-mail: **ddwyer@ddwyer.demon.co.uk** and Joakim Langlet, e-mail **joakim@seaview.se**, have VCXOs available. For used Rb oscillators, try Wade Lehman, **http://www.lehman.scientific.com**.
[4] For more information about the 4046, see Neil Heckt, "A PIC-Based Digital Frequency Display," *QST*, May 1997, pp 36-38.
[5] A good introduction to PLLs and digital filters appears in Chapters 14 and 18 of *The ARRL Handbook*, 75th ed, 1997. The excellent and classic work in the field is F. M. Gardner, *Phaselock Techniques*, (New York: John Wiley & Sons, second ed, 1979).

[6]Digital PLLs are discussed by C. L. Phillips and H. T. Nagle, *Digital Control System Analysis and Design*, (Englewood Cliffs: Prentice-Hall, Inc third ed, 1995), and by R. E. Best, *Phase-Locked Loops*, (New York: McGraw-Hill, 3rd ed, 1997); watch out for typos in the equations). The *MatLab* software package (MathWorks, Inc, Natick, MA; tel 508-653-1415, has a digital filter package that is useful for checking designs. A student edition of the software is available.

[7]Switches S1 through S3 of U11 can be considered as controlling a three-bit number, N, in the range of 0 to 7, where *closed* is 0 and *open* is 1. Then, N = 0 is setup mode, N = 1 implements a first-order PLL with no filtering beyond the 30-second integration of the phase measuring circuit, and N = 2 through 7 implement second-order PLLs with time constants, τ, starting at 1500 seconds and increasing by a factor of two at each step to approximately 13 hours. Here, $\tau = 2\pi \div \omega_n$ where ω_n is the natural loop frequency. It is approximately the time required for the PLL to fully recover from a transient (see Notes 5 and 6).

[8]A procedure for selecting R6 and R5 is as follows. (1) Construct a VCXO control input network that is suitable for your VCXO. Use high-stability resistors and/or wire-wound multiturn pots. Typical values are in the 10 to 20 kΩ range for the **COARSE** adjustment and 1 kΩ for the **FINE** adjustment pot. (2) Ground the bottom end of the **FINE** adjustment pot, set it to the center of its range and adjust the coarse pot to bring the VCXO to approximately the correct frequency. (3) Now, measure the resistance values you have for **RA** and **RB**. (4) Determine the *relative* frequency change (Δ F/F) that occurs when you adjust the pots to change the voltage by 1 V at the control input. This value is the control sensitivity, S. A triggerable 'scope and a second stable oscillator may be useful in determining S. Trigger the 'scope from the second oscillator and observe the VCXO signal. The VCXO signal will probably drift slowly across the screen. The change in the *rate* of drift when the VCXO control voltage is changed by 1 V is S. (5) Select a value for R5 around 100 Ω (the value should be much less than **RA** and **RB**). (6) Finally, compute the value of R6 you need from the equation

$$R6 = R5[(RA / (RA+RB) \times (S / 7.5\times10^{-9})) - 1]$$
(Eq 1)

[9]The Oncore VP and the GPS-25 receiver boards are presently available via a bulk purchase from TAPR, Tucson, AZ; tel 940-383-0000. The Oncore is more expensive, but offers a more-stable timing output. The recently introduced Oncore UT+ model is a less-expensive version of the VP that is said to retain its high quality timing. It is available from Synergy Systems, Inc; tel 619-566-0666; http://www.synergy-gps.com.

[10]At each 30-second DAC update, the controller prints three five-digit (16-bit) numbers, each separated by CR and LF characters. The first of the three numbers is the total count from the phase detector counter (U2A/U4) for the previous 30 seconds. When multiplied by the constant, 41.7 ns per count divided by 30 counting intervals = 1.39, this number is the phase difference in nanoseconds between GPS and the VCXO. The controller attempts to keep this count constant at the value 1024. By using a phase difference offset from zero the controller can easily track both positive and negative phase changes. The second of the three ASCII numbers is the value the digital filter is currently sending to the DAC to control the VCXO frequency. This number can be either positive or negative (depending on whether a positive or negative VCXO disci-plining voltage is needed) and is in two's complement binary notation (values larger than 32768 are interpreted as negative and equal to the value minus 65536). Only the most-significant 16 bits of the 18-bit DAC input are printed. The third ASCII value indicates the status of the controller. This number is a combination of three values arranged so

that it is easy to see the status at a glance. The current filter switch setting determines the lowest decimal place (0-7). If the phase difference is far from the set point, suggesting that the PLL is not locked, the value 100 is added to the number. If the phase has just changed abruptly, which invokes a "deglitching" algorithm in the software, the value 10 is added. For example, the value 105 indicates that filter time constant 5 is in use and that the phase lock may be questionable.

[11]Opening S5 will hold the DAC voltage at its current setting, thereby preventing the controller from further changing the VCXO frequency. However, the ASCII output continues to provide data as before so that the "open loop" GPS-VCXO phase drift can be monitored. In normal operation, S5 should be closed, but opening it for short periods is an effective method of eliminating "GPS jitter" that can be useful when ultra-stable short-term performance is needed.

[12]Variations in the electron density of the ionosphere can also cause the arrival time of the GPS signals to vary; however this is partially corrected in the GPS receiver using data broadcast by the satellites. The residual uncorrected variation is only a few nanoseconds.

[13]These AV performance numbers are based on my estimates and the unpublished GPS stability measurements of W3IWI in connection with the Totally Accurate Clock project (see www.tapr.org).

[14]Contact me at RR 2 Box 423 Santa Fe, NM 87505, or by e-mail at **shera@rt66.com**. The binary software is free for noncommercial uses. The software is in the file *GPSCNTRL .ZIP* and can be found on the Internet (ftp to **oak.oakland.edu/pub/hamradio/arrl/qst-binaries** and on the ARRL BBS 860-594-0306.

[15]Start with the Microchip Web site at **http://www.microchip.com**.

Brooks Shera, W5OJM, was first licensed in 1961 as K8WGA. He received his MS in Physics from the University of Chicago and a PhD in Physics from Case Western Reserve University. Brooks has been employed for most of his professional career at Los Alamos National Laboratory in New Mexico, where he has conducted basic research on the structure and properties of atomic nuclei. More recently, Brooks has developed laser-based techniques for detecting single atoms in liquid solutions and is applying these methods to biological and medical research. He is presently an independent consultant focusing primarily on developing instrumentation for research. Brooks holds several patents, is the author of more than 100 research papers, and is a Fellow of the American Physical Society. He can be reached at RR 2 Box 423 Santa Fe, NM 87505; e-mail shera@rt66.com.

New Products