A Simplified GPS-Derived Frequency Standard

Here is a simple and modern approach to a 10-MHz frequency standard.

Bertrand Zauhar, VE2ZAZ

242 Robert-Martial St
Gatineau, Québec J9J 2V1
Canada
ve2zaz@amsat.org

For many reasons, an accurate frequency standard at an Amateur Radio station is desirable. For weak-signal operation such as EME (moonbounce) at microwave frequencies, you must be transmitting and listening exactly at the right frequency; otherwise the narrow receive filters used will make you miss that weak signal. With such a setup, a 10-MHz frequency reference feeds the frequency synthesizer that generates the radio's operating frequency. An accurate 10-MHz reference is also useful for test equipment adjustment. With an accurate standard, you can put frequency counters and signal generators on track.

The advent of the Global Positioning System (GPS) has allowed a simplified approach to time and frequency accuracy. Several commercially available GPS receiving units provide a 1 pulse-per-second (pps) signal. This signal typically exhibits a short-term accuracy of ±1 microsecond (1 ppm or $\pm 1 \times 10^{-6}$).

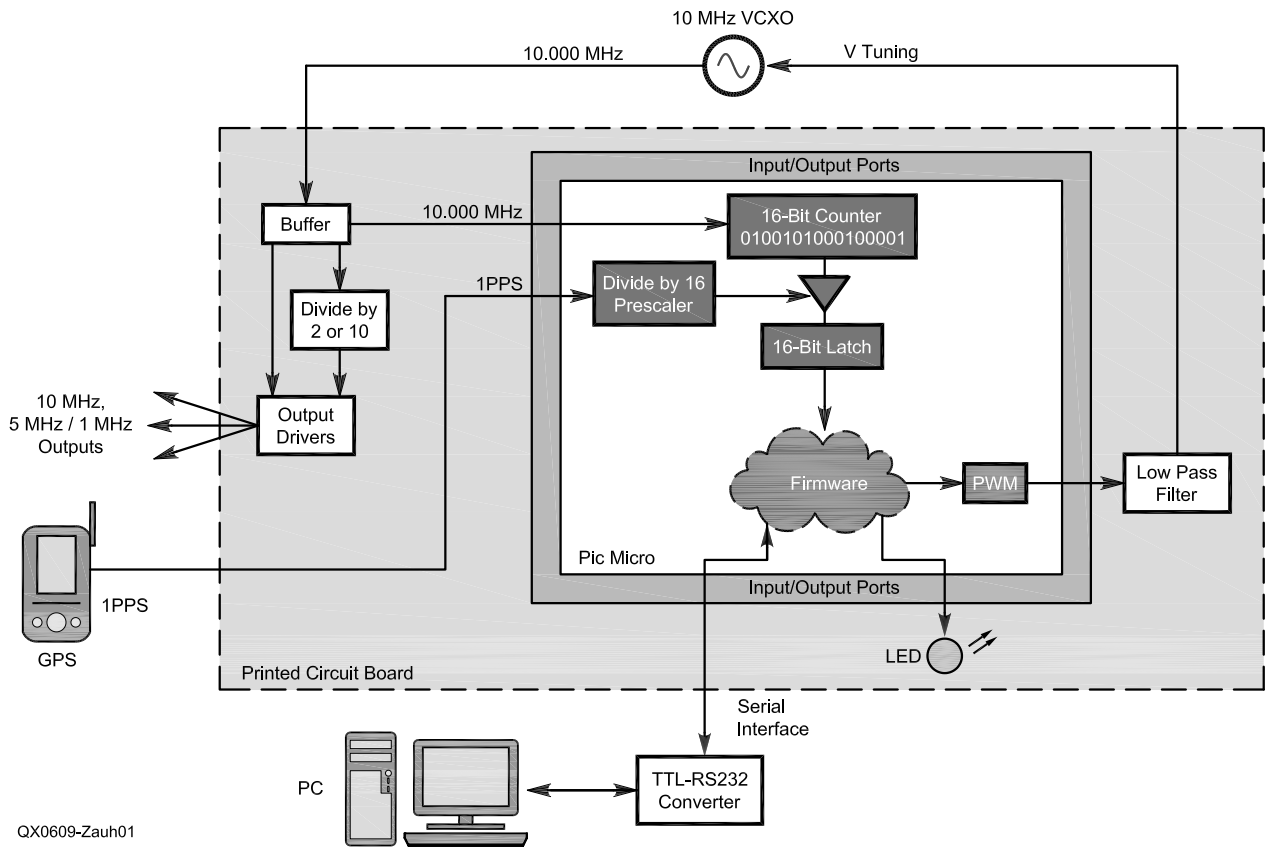

Figure 1 — Block diagram of the GPS-derived frequency standard.

By averaging it over a long period, a much better accuracy can be achieved. This is what this project does: it locks an external 10-MHz voltage-controlled signal source to the 1 pps GPS signal.

The good work of Brooks Shera[1] has generated a lot of interest within a broad community of experimenters who want to increase the level of frequency accuracy available to them at low cost. His system uses a PLL technique to lock an external oscillator to a GPS receiver and obtain an accurate frequency standard.

The project I present here provides a simpler and more modern approach to a GPS-derived 10-MHz frequency standard. Improvements found in today's technology offer the following benefits: solid performance, more features and a reduction in the number of components.

This design differentiates itself from other previously published designs because:

- It uses a simpler frequency measurement technique, as opposed to phase measurement.
- It provides on-board reference buffering and fan-out with 50-Ω output impedance.
- It provides the three most common reference frequencies of 10 MHz, 5 MHz and 1 MHz.
- It provides full software control of the frequency acquisition and control processes, without DIP switches.
- It has fewer components and does not require an external DAC or external input counter chips.
- It runs off only one supply voltage: +5 V dc (excluding the VCXO supplies).

Tests have shown that this system consistently produces a short-term reference accuracy in the 1×10^{-10} range. This is derived using standard automotive-grade GPS receivers. That range of accuracy does not rival cesium-based references. It is much better than most of the standard built-in, free-running oscilla-

[1]Notes appear on page 21.

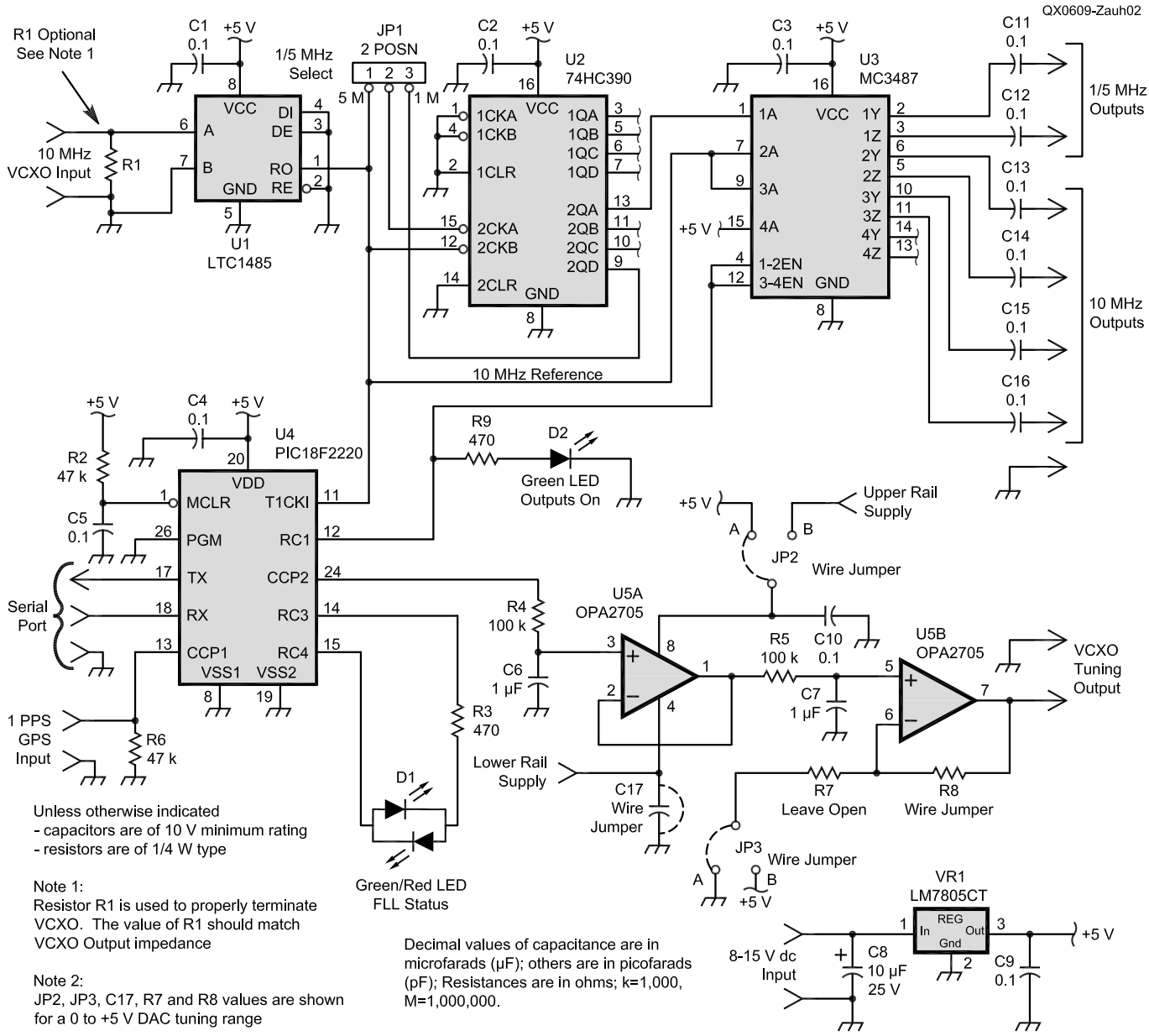

Figure 2 — Schematic diagram of the GPS-derived frequency standard circuit.

tors seen in commercial test instruments, however. Just to give you an idea of the type of accuracy, one part in 10^{10} represents an error of one hertz on a 10 GHz signal!

System Description

Figure 1 shows a block diagram of my GPS-derived frequency standard. The system operates a hardware/firmware frequency-locked loop (FLL). In essence, the system compares a local frequency source (an external oscillator) to a GPS-derived reference. It will adjust the local 10-MHz variable source to match the GPS-derived 1 pps reference. The 10-MHz source is kept aligned with respect the 1-Hz GPS reference on a real-time basis by the firmware. The resulting 10-MHz reference is fanned out, frequency-divided and provided to the user for high-accuracy applications. System control and monitoring are achieved using a bicolor LED and a serial port connected to a terminal (PC).

Hardware Description

Figure 2 shows the system circuit schematic. The main operation consists of counting the number of rising edges produced by the 10-MHz voltage-controlled crystal oscillator (VCXO) signal over a 16 s period (16 GPS pulses). If the GPS and the VCXO are at the same frequency, exactly 160,000,000 pulses will be counted (±1 pulse, inherent to counter technology).

Prior to entering the microcontroller, the 10-MHz VCXO signal is buffered and amplified by U1, a receiver chip. An optional input termination resistor R1 can be added if the VCXO's output circuit calls for one. The buffered 10-MHz reference signal is fanned out to several locations on the board.

U4, the Microchip PIC18F2220 microcontroller, has a built-in 16-bit counter incremented by an external source, the VCXO. The counter value is latched by another external signal rising edge, the GPS 1 pps signal in our application. This process is totally autonomous and independent from firmware. The microcontroller's task in this process is to analyze the results and adjust the VCXO frequency accordingly.

VCXO Frequency Control

The Microchip PIC18F2220 microcontroller does not have an integrated digital-to-analog converter (DAC). To produce an adjustable voltage source to vary the VCXO frequency, the built-in 10-bit pulse-width modulator (PWM) is used instead. A continuous rectangular-wave output is produced by the PWM. A downstream external 1-Hz, two-stage low-pass filter (U5A, U5B and discrete components) is used to recover the average dc value of the PWM output. By varying the duty cycle of the PWM, it is possible to produce an accurate analog dc voltage with 2^{10} or 1024 steps over the range.

A DAC of 14-bit resolution is achieved by precisely controlling the duty cycle of the PWM output. This translates to a tuning granularity of 6×10^{-5} Hz for a VCXO that has a 1-Hz tuning range. Achieving a 14-bit DAC using a 10-bit PWM requires additional firmware processing. The idea is to "dither" the pulse width within a 16-cycle window. Those 16 cycles translate into an additional 4-bit resolution. For example, increasing the 14-bit DAC output by one step involves increasing the 10-bit PWM output width by one increment on one of the 16 pulses. Increasing the DAC by two steps means increasing the 10-bit PWM output width by one increment on two of the 16 pulses, and so on.

To add flexibility for interfacing with various VCXOs, the filtering stages have a supply bypass feature that allows you to feed the operational amplifiers with different upper and lower rail voltages. This is done by reconfiguring JP2 and C17. Remember, though, that the maximum voltage difference between upper and lower rails must be kept to 12 V or less. Another feature, the second stage of low-pass filtering, allows for additional gain using R7/R8 if the VCXO operates on a larger tuning voltage range than the more standard 5-V range. A 5-V offset can also be added to the second stage using jumper JP3. This provides support for VCXOs that have a –5 V to +5 V tuning range.

Table 1 lists some of the possible configurations on the filtering stages for various VCXO tuning ranges. Finally, the tuning slope sign can be set in firmware to accommodate both types of VCXOs.

Output References

The system provides up to four 10-MHz reference signals. In addition, it provides up to two references with a selectable frequency of either 5 MHz or 1 MHz. These sub-rates are produced by U2, a synchronous counter. The active sub-rate is selected with an onboard jumper, JP1. All references are of 50-Ω output impedance and provide an amplitude of greater than 1 V pk-pk with a square wave shape. These signals are provided by U3, a line driver chip.

When the firmware feature is enabled, the reference outputs are inhibited if the FLL goes into its unlocked state. That protection ensures that the user does not use a reference of unknown quality. LED D2 provides an indication of the reference output state.

FLL Status LED

The system provides basic FLL status and alarm conditions with a single bicolor LED (D1). The LED control is designed to allow the user to learn about the current and past FLL status in a lapse of a single second. Being a combined green/red LED pair in a single package, it can produce three colors: green, red and amber. The latter is produced when both green and red LEDs are simultaneously on. Additionally, the LED unit will flash at a 1-Hz rate to provide a 1 pps signal sanity check.

Serial Port

Since the FLL status LED only provides basic FLL status and alarm, a serial port is also implemented. The serial port provides comprehensive control and monitoring of the FLL and other firmware features. When interfaced through a TTL-to-RS232 bidirectional converter, the serial port will connect to a personal computer RS-232 port. Such a converter can be easily assembled using a couple of transistors or can be made using Maxim's MAX23x series of conversion chips. The Internet has several simple circuits documented. It can also be purchased "off-the-shelf."

The serial port is ASCII-character-based

Table 1
Tuning Range Configuration

VCXO Tuning Range	*R7*	*R8*	*C17*	*JP2 Upper Rail*	*JP3 Offset*
0 to +5 V	Leave Open	Jumper wire	Jumper wire	Position A	Leave Open
0 to +8 V *	100 k	62 k	Jumper wire	Position B	Position A
0 to +10 V *	100 k	100 k	Jumper wire	Position B	Position A
–5 V to +5 V **	100 k	100 k	0.1 μF	Position A	Position B

*External positive supply required.
**This configuration is used with the HP 10544/10811 series OCVCXO's. External –5 V supply required.

and provides standard status text strings that can be captured and analyzed by the user. The serial port also interprets a series of user commands to control the FLL and other firmware features. Additional information on the text strings and user commands can be found on my Web site.[2]

+5 V dc Voltage Regulator

To guarantee a stable and clean voltage supply to the board, a separate +5 V dc fixed voltage regulator, VR1, is used. A heatsink is required on the regulator since the board could potentially draw close to 200 mA when all reference outputs are terminated. This would make the voltage regulator excessively hot without a heatsink.

External VCXO

System performance is set to a large degree by the external 10-MHz VCXO selected for this design. It is recommended that an oven-controlled VCXO (OCVCXO) be used to provide better short term stability. Good, second-hand Hewlett-Packard OCVCXOs are readily available on eBay.[3] These should be your primary targets.

This project has been tested and proven with OCVCXOs that have a 1-Hz or a 10-Hz tuning range over their entire control voltage. Different tuning slopes may require different firmware parameter settings to achieve optimum performance. The user will want to experiment with these.

GPS Receiving Unit

The GPS receiver used on this system must have a 1 pps TTL-compatible output signal. Accuracy of this signal will influence overall accuracy of the system. Typical accuracy seen on automotive-grade units is in the order of 1 microsecond (1×10^{-6} s) and this is satisfactory for our application.[4]

This design has been tested with the Garmin GPS-35 and the Motorola Oncore GT+ GPS receivers. Both meet the above-mentioned accuracy. The Motorola Oncore GT+ unit is actually specified as having a somewhat better accuracy on the 1 pps output. This did not materialize in better overall system accuracy during evaluation. In fact, the second-to-second "jitter" on the 1 pps signal happened to be greater than on the Garmin GPS-35, while still meeting the above specification. This jitter gets averaged out, though, by the FLL since many frequency samples are taken before making a VCXO frequency correction.

For better accuracy, I recommend setting the GPS unit to fixed-position (position-pinning) mode. In this mode, the GPS firmware assumes a fixed location. This translates into a more accurate 1 pps signal. Consult the GPS unit documentation for more details on how to set the GPS unit to this mode.

Printed Circuit Board

I designed a double-sided circuit board to integrate the hardware.[5] Figure 3 shows a top view of the circuit board assembly. Its dimensions are 2.8 × 3.5 inches. IC sockets are optional, but recommended. This is especially applicable to the microcontroller. A fine-tip soldering iron should be used. I am making the circuit board design files available to the public.[5]

The user may elect to build the circuit using other techniques such as breadboard and point-to-point wiring. The layout is not that critical. Care should be taken in proper dc supply decoupling near the integrated circuits.

Firmware Description

The firmware running on the PIC microcontroller was written in assembly language. It is well documented in the source file. Both the firmware source code and the assembled hex code can be downloaded from my Web site.[6] (Also see Note 2.) Once assembled, the hex code takes about 3 kbytes of flash memory space.

FLL Acquisition and Control Cycle

A simplified FLL acquisition and control cycle flow chart is shown in Figure 4. The process starts from the left hand side and repeats indefinitely as long as the FLL is enabled. The firmware samples the 10-MHz VCXO signal for a duration of 16 seconds. Based on how close the sample is to the nominal frequency,

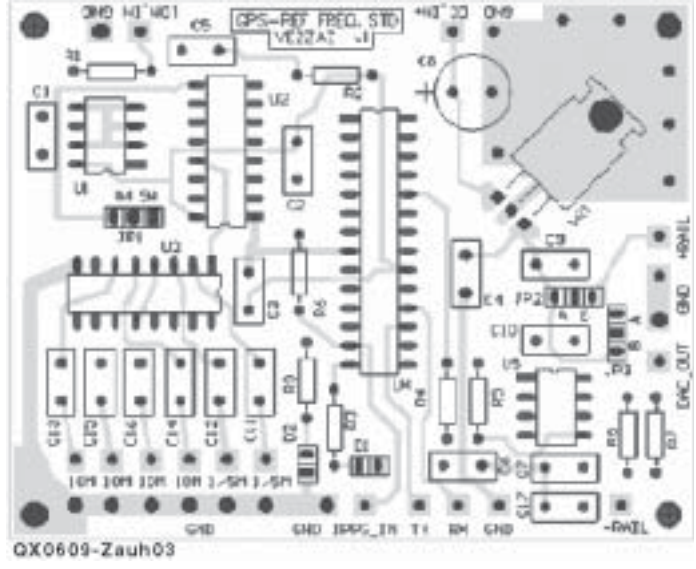

Figure 3 — Circuit board pattern for the GPS-derived frequency standard.

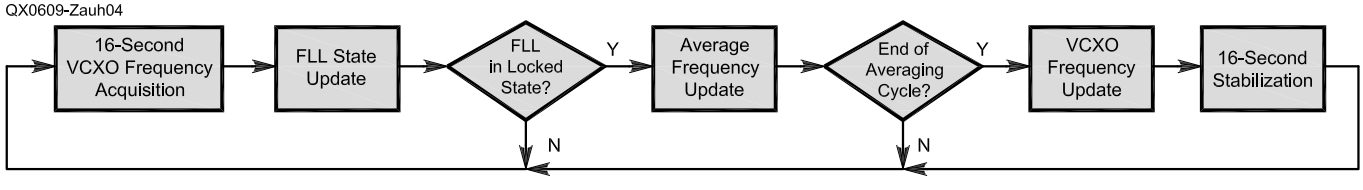

Figure 4 — FLL acquisition and control cycle.

it will update the FLL state. If the sampled frequency is within the specified limits, the firmware will add the sample to the average frequency calculation. Otherwise, it will drop the sample and start another sample acquisition.

After updating the average frequency, the firmware verifies if the frequency averaging cycle has reached the specified number of samples. If it has, the DAC output is updated to reflect the required VCXO frequency change based on the calculated average frequency. Otherwise, the firmware simply starts another sample acquisition.

Whenever the DAC output value is changed, a 16-second pause is inserted to allow the VCXO to stabilize before the next frequency averaging cycle begins.

FLL States and Transitions

Figure 5 illustrates the various states and transitions seen during FLL operation. Under normal "stabilized" conditions, the FLL will be in locked state, with occasional transitions to holdover state to reject GPS receiver timing impairments. The unlocked state is seen at system startup or if the holdover state extends for too long. Finally, the disabled state is initiated by a user command and essentially disables the FLL. In this mode, the FLL merely maintains the last valid DAC voltage.

The current FLL state can be clearly identified in the FLL text string sent over the serial port every 16 seconds. It can also be found by assessing the LED color: green for locked, amber for holdover, red for unlocked and disabled. These colors indicate the relative reliability level of the 10-MHz reference output.

Frequency Averaging Modes

Two averaging modes are available to the user. I named them summing mode and voting mode. The summing mode is more effective when the frequency averaging cycle is shorter, for example five minutes. This mode will be helpful in acquiring a lock at a faster pace. The voting mode allows the detection of a trend and is more effective when the frequency averaging cycle duration is large, for example greater than 30 minutes. This mode will be helpful with GPS units that put out a 1 pps signal with significant second-to-second jitter, as the FLL does not care about the size of the frequency difference, but merely the sign. Tests have also shown that this mode yields the best accuracy. The voting mode should be used in conjunction with long frequency averaging cycles. There are more details on how these modes operate in the user manual.

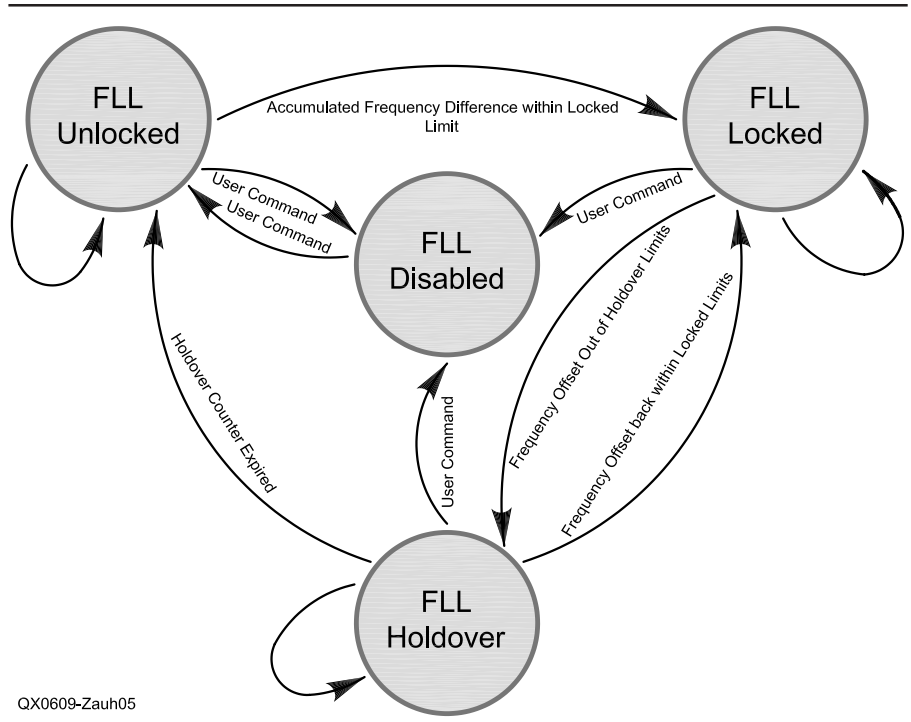

Figure 5 — FLL states and transitions diagram.

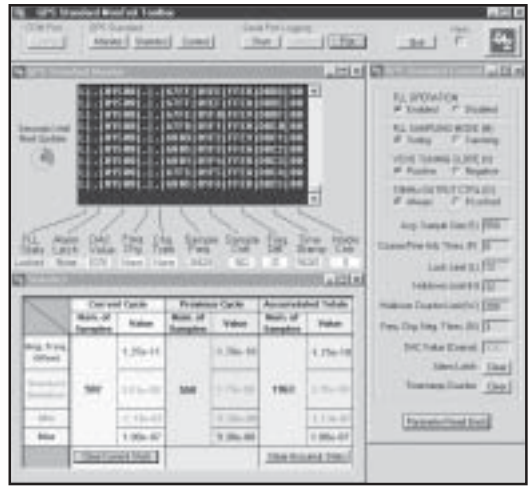

Figure 6 — The *Windows GPS Standard MonTrol* software provided by the author.

Operation

At power up, the system will display a start-up prompt at the serial port, including firmware version. It will then reload the previously saved FLL parameter settings from data flash memory. If the microcontroller is powered up for the first time after programming, default parameter values will load.

A few parameters must be set by the user via user commands, because different GPS receivers and different VCXOs require different parameter values. Parameters such as the number of samples taken before updating the VCXO frequency, the number of samples allowed while in holdover state and the tuning slope sign of the VCXO must be set. A detailed description of each of these parameters is included in the systems user manual available on my Web site (see Note 2).

The system FLL will start from the unlocked state and will try to acquire frequency samples automatically. VCXO frequency adjustments will be made at the end of each averaging cycle. After a period that may span from minutes to hours, the system will transition to locked state. Once the system is set up and in locked state, it should require little maintenance.

Every 16 seconds, and under any circumstances, the firmware sends a status string that provides detailed information on the FLL acquisition process and alarm condition. An example of such status string is shown here:

L|U|01FF6|+|F|67FC|0120|FFFD|007D|03

The definition of each field in the status string is available in the system user manual.

The user will want to accelerate the initial acquisition process by manually tuning the FLL closer to the target DAC value that yields a GPS-to-oscillator lock. This is achieved by zero-beating the 10-MHz oscillator to another known-good reference. The NIST WWV/WWVH or NRC CHU shortwave radio stations are good sources to achieve this.

Once the system is stabilized, the user can increase the averaging cycle duration. The longer the sampling cycle, the more accurate the frequency measurement will be. A typical longer sampling cycle will last from one to four hours. This means that the system will accumulate frequency errors for this period before making a VCXO frequency change. With a good and stable VCXO, the averaging cycle can be made even longer. This will improve system accuracy even further.

To make the man/machine interface even more friendly, I have created a Windows program to monitor and control the system. A snapshot of the software screen is shown in Figure 6. The compiled Windows software is available for download on my Web site.

Results

Figure 7 shows an FLL acquisition phase. The graph illustrates the VCXO DAC value as a function of time. Initially, the DAC value was intentionally set 0.1 Hz below the nominal value to highlight the acquisition process. The averaging cycle was set relatively short at five minutes between DAC value updates. The horizontal portion of the curve indicates that the system reached an equilibrium. Under these conditions, the system is considered to be stabilized. At this stage, though, ultimate accuracy may still not be achieved, since long av-

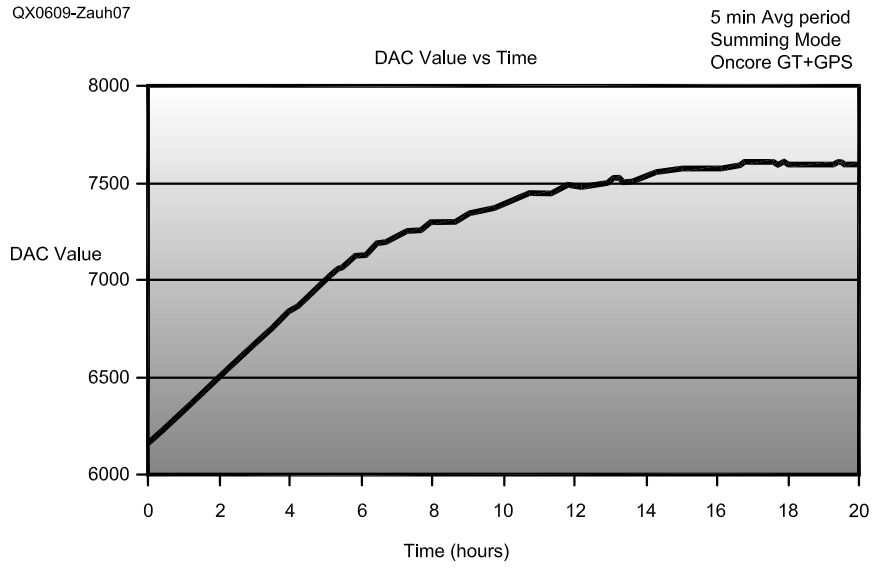

Figure 7 — DAC behavior during FLL frequency acquisition.

Table 2
Parts List

Component Reference	*Description*	*Part Number*
C1-C5, C9-C16	0.1 μF, 10 V or higher, radial	Digikey: 478-2472-ND or equiv.
C17	Piece of wire for a 0-5 V VCXO tuning range.	
	0.1 μF, 10 V or higher, radial for some other ranges. See text.	Digikey: 478-2472-ND or equivalent.
C6, C7	1 μF, 10 V or higher, radial	Digikey: 478-2479-ND or equiv.
C8	10 μF, electrolytic, 25 V or higher	Digikey: P1176-ND or equiv.
D1	Bicolor (Green-Red) LED, two leads.	Digikey: MV6461A-ND or equiv.
D2	Green LED	Digikey: MV5477C-ND or equiv.
JP1	1 × 3 Header, 0.1 inch spacing, One computer jumper required.	Digikey: S1012-36-ND. Cut to size, or equiv.
JP2	Piece of wire. Selects upper operational amplifier rail. See text.	
JP3	Piece of wire. Adds –5 V offset. See text.	
R1	51 Ω, ¼ W, axial. Optional. See text.	Digikey: P51BACT-ND or equiv.
R2,R6	47 kΩ, ¼ W, axial	Digikey: P47KBACT-ND or equiv.
R3, R9	470 Ω, ¼ W, axial	Digikey: P470BACT-ND or equiv.
R4, R5	100 kΩ, ¼ W, axial	Digikey: P100KBACT-ND or equiv.
R7	Optional, do not populate for a 0-5 V VCXO tuning range. See text.	
R8	Piece of wire for a 0-5 V VCXO tuning range. See text.	
U1	Linear Technology LTC1485, DIP-8 package	Digikey: LTC1485CN8-ND
U2	74HC390, DIP-16 package	Digikey: 296-9199-5-ND
U3	MC3487, DIP-16 package	Digikey: 296-1408-5-ND
U4	Microchip PIC18F2220, DIP-28/0.3 inch package, programmed part 5	Digikey: PIC18F2220-I/SP-ND
U5	Texas Instruments OPA2705, DIP-8 package	Digikey: OPA2705PA-ND
VR1	7805 Voltage Regulator, TO-220 package	Digikey: LM7805CT-ND
Sockets for:		
U1, U5	IC Socket, 8-pin, 0.3 inch spacing, low profile, optional	Digikey: ED3108-ND or equiv.
U2, U3	IC Socket, 16-pin, 0.3 inch spacing, low profile, optional	Digikey: ED3116-ND or equiv.
U4	IC Socket, 28-pin, 0.3 inch spacing, low profile, recommended	Digikey: ED3128-ND or equiv.
Heat sink for VR1	TO-220 Compact Heat sink, recommended	Digikey: HS107-ND or equiv.

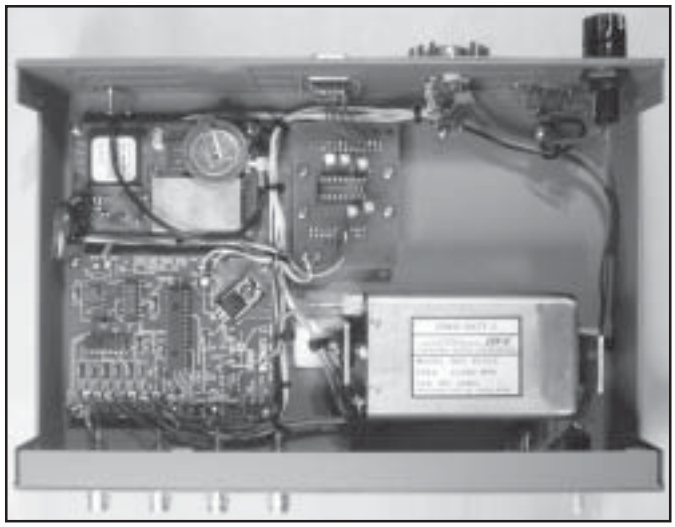

Photo A — The complete frequency standard system integrated into an enclosure. The unit at the top left corner is the GPS board. The bottom-left unit is the FLL controller board and the oven controlled variable crystal oscillator is at the bottom right. The RS232 converter chip and related circuitry is on the board shown at the top center. Also shown in the top-right corner is an 8 V regulator and dc input fuse holder.

Photo B — This photo shows the GPS derived frequency standard connected to an HP frequency counter.

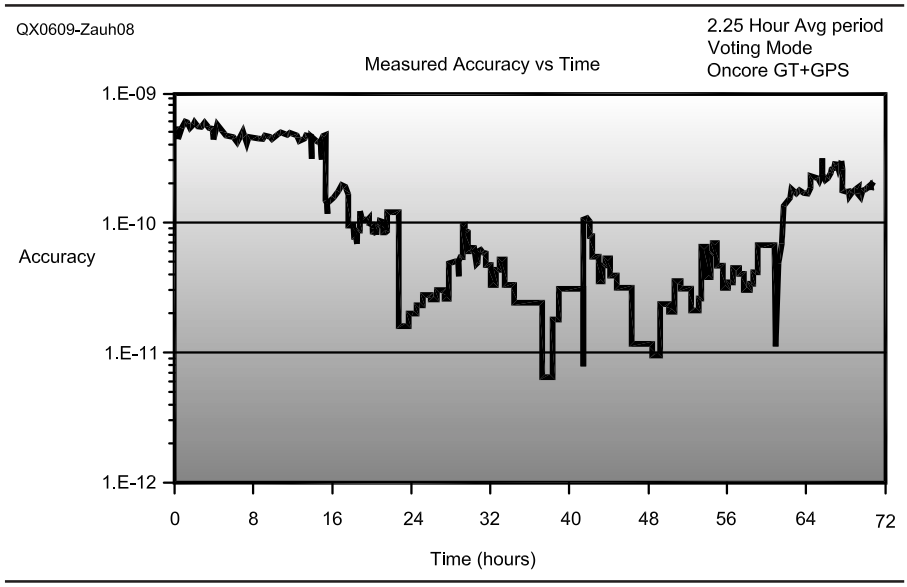

Figure 8 — Frequency accuracy for a 72-hour period.

eraging cycles and fine DAC adjustments are required to achieve higher accuracy.

Figure 8 shows the measured system accuracy as a function of time for a 72-hour period. The reference I used for this comparison is Canada's CHU atomic time radio station.[7] The graph shows frequency accuracy as a function of time. From the graph, we can conclude that rubidium-like accuracy can be reached, but cannot be guaranteed. This is due to several factors. One of them is OCVCXO stability over time, voltage, temperature and vibration. When shooting for atomic-like accuracy, the slightest disturbance on the VCXO will make it shift in frequency. This is why I believe that it is difficult to obtain an accuracy better than the 10^{-10} range on OCVCXO-based systems.

Conclusion

Assuming a reasonably stuffed junkbox, you will be able to assemble and own this high-accuracy 10-MHz standard for less than $200 US, including VCXO and GPS receiver. This represents a tiny fraction of the cost of atomic-based systems. While using it, you will also learn a lot about frequency accuracy, stability and related measurement techniques. You will no longer have doubts about your transmitted frequency or your frequency counter reading!

Notes

[1]B. Shera, W5OJM, "A GPS-Based Frequency Standard," *QST*, Jul 1998, p 37.

[2]I maintain a Web site where I provide updates to the project, a detailed user manual, source files and additional comments. Please visit **www3.sympatico.ca/b.zauhar** for more details.

[3]Look for HP10544A, HP10544-xxxxx, HP 10811A or HP 10811-xxxxx Crystal Oscillator units. These units differ one from the other in their stability and tuning range, but all of them should work in our application. Expect to pay around $80 for a working unit when purchased on eBay.

[4]GPS receivers that provide a 1 pps signal can commonly be found new on eBay for less than $50. These are OEM versions without display. On such units, settings are entered via a serial port.

[5]I distribute high quality, fully-etched bare circuit boards for this project. Please contact me via e-mail if you are interested in purchasing a circuit board. I am also making the circuit board layout files available as PDF files on my Web site. Printing them at a 1:1 scale will give accurate printouts. The layout is made in such a way that component leads pass the signals from one layer to the other. In the case of a hand-made circuit board, all component leads and wires should then be soldered on both sides of the board.

[6]For those of you who cannot program Microchip PIC18F series microcontrollers, I make pre-programmed PIC micro-

controllers available for purchase. Please contact me via e-mail for more details.

[7]I use the carrier frequency of the 14.670 MHz CHU atomic time radio station for frequency measurement purpose. I basically add (zero-beat) a signal generator synchronized off the GPS-derived 10-MHz standard with the CHU signal into a shortwave receiver. From the duration between two signal nulls, I derive the frequency difference. I compute accuracy using the following formula: Accuracy = 1 / ($T_{null\text{-}null}$ × 14.67 × 10^6). The National Research Council of Canada maintains three atomic clocks, and the CHU carrier frequency is derived from them. Carrier frequency accuracy is guaranteed at 5 × 10^{-12} or better. I am fortunate to live about 15 km from the transmitters. With such a small station-to-station distance, the propagation mode is ground wave and most certainly surface wave. Since surface wave propagation is very stable (no phase distortion), I can consider the received CHU carrier frequency (and phase) to be virtually as good as the 5×10^{-12} they guarantee at their antenna, and at least one order of magnitude better than the level of accuracy I am trying to measure.

*Bertrand Zauhar has been a radio amateur as VE2ZAZ since 1984. He holds an advanced amateur license. Bertrand has designed for the hobby, amongst other things, an L-band transmit converter (*Amsat Journal*, May/June 2003), a 1 to 12 GHz frequency counter prescaler, a microprocessor-based repeater controller, several amateur satellite antennas and a RF-sensing alarm (*73-Amateur Radio*, May 1998). Bertrand received his Electronics Engineering degree in 1989 from École Polytechnique de Montréal. Since then, his professional engineering career has been spent working for Nortel at the Montréal and Ottawa locations. In his current position, he is an electronics hardware design engineer on optical transmission equipment.* QEX