OpenSource GPS

A Hardware/Software Platform for Learning GPS: Part I, Hardware

Clifford Kelley, OpenSource GPS**, and Douglas Baker,** GPS Creations

I BUILT MY FIRST CRYSTAL RADIO KIT WHEN I WAS 9 YEARS OLD. I BECAME HOOKED ON RADIO technology and later went on to build other radios, even one using a razor blade and pencil lead! I learned how radios worked by building them and tinkering with them. Radio technology has advanced considerably since my youth and although it is still possible to buy a crystal radio kit, most modern radios are digital black boxes. It's not impossible for the radio enthusiast to learn how these devices work, just rather more difficult. And the difficulty comes, in part, from the inability to tinker with these radios — to learn by doing. The GPS receiver is no exception. A simple handheld GPS receiver costs less than $100 — within the budget of many students. But users of these digital wonders have very little control over their operation. After all, they were designed to be operated by virtually anyone with next to no training.

So how can interested individuals on a budget learn how a GPS receiver works – not just superficially but at the level of the tracking loops and correlators? Can one actually tinker with the operation of a GPS receiver? Until recently this was difficult to do. Sure, you could try to build a GPS receiver from scratch using a chipset and program the controlling microprocessor but that is beyond the capabilities of many individuals who simply want to experiment with a receiver's operation. But thanks to a recent collaborative effort involving GPS technology enthusiasts from around the globe, it is now possible to learn GPS by doing. The OpenSource GPS project has developed an economical hardware and software platform specifically for educational purposes. In this month's column, two of the project's leaders, Clifford Kelley and Douglas Baker, present part one of a two-part article about the project. This month, they will discuss the hardware component, built around the Zarlink GP2015/GP2021 chipset. Next month, they will describe the platform software and the results of static and kinematic tests of this innovative tool for GPS education. — R.B.L.

Teaching the next generation of engineers about the inner workings of GPS receivers is difficult because of the expense of acquiring the appropriate hardware and software. In the past few years, several excellent books have been written about GPS (see the "Further Reading" sidebar), but, in the end, students learn best by doing. Even with available hardware, trying to squeeze software development of into a university quarter or semester is asking a lot of both professors and students. Together with collaborators from around the globe, we developed a low-cost set of hardware and free open-source software that allows students access to the inner workings of a GPS receiver without breaking the bank in terms of time or money.

The hardware and software of the OpenSource GPS project initially was based on the venerable GP1010/GP1020 chipset first introduced by GEC Plessey Semiconductor in Swindon, United Kingdom, in the early 1990s. This company was acquired by Mitel Semiconductor, Ottawa, Canada, in 1998 and reborn in 2001 as Zarlink Semiconductor. Shortly after the introduction of the chipset, GEC Plessey released a board and software called the GPS Builder, which placed the down converter and correlator chips on an Industry Standard Architecture (ISA) bus card for installa-tion in a PC. Because the software ran on PCs, it gave users complete control and visibility into the operation of the receiver. Unfortunately, GEC Plessey and Mitel found that they had to charge well in excess of $1,000 dollars for the board and software licenses, which limited its popularity among cash-strapped universities and individuals.

That pioneering educational product provided the stimulus for the Open-Source GPS project, which was initiated in 1995. Similar to the GPS Builder, OpenSource GPS provides students an opportunity to learn about GPS by experimenting with the hardware and software of a GPS receiver. In this "Innovation" article, we review the history of the project and briefly describe the current hardware and software and some of the recent testing. Additional details about the project can be found in items listed in the "Further Reading" sidebar and at its website, *www.home.earthlink.net/~cwkelley*, which describes the project and provides source code and sample input and output data.

Hardware

GPS receiver hardware is complicated by the fact that radio frequency (RF) signal processing occurs in close vicinity to digital signals, which can interfere with the RF signal chain. In 1995, we designed and built a two-sided board based on the GP1010/GP1020 chipset (see **Photo 1**). The GP1010 was the front-end or down-converter chip, and the GP1020 was the baseband or correlator chip with six correlator channels. The board used two GP1020 chips to provide a total of 12 channels. Although it did work, it was obvious that it had problems resetting and offered limited functionality. Build-
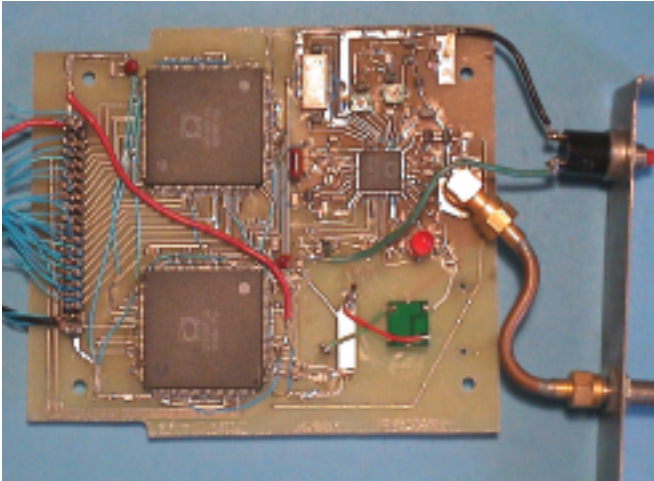
▲ **PHOTO 1** Original OpenSource receiver (1995)
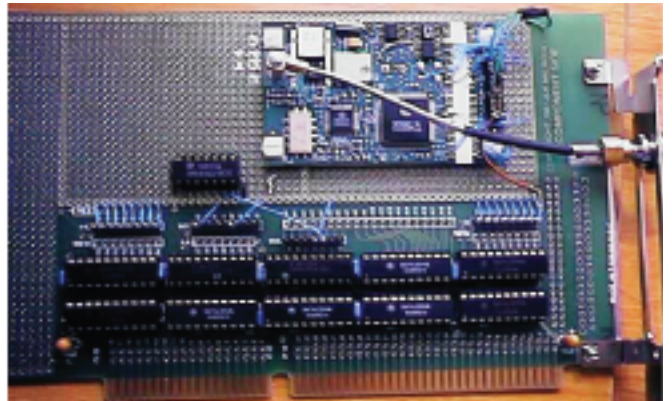
▲ **PHOTO 2** Hacked SuperStar receiver (2001)

▲ **PHOTO 3** GPS 1000 ISA interface with daughter board

▲ **PHOTO 4** GPS 1005 PCI interface with daughter board

ing high-quality, four-layer surface mount component boards is expensive and requires specialized test equipment.

Another alternative was needed. While working with the Canadian Marconi (now, NovAtel) AllStar and SuperStar receivers, we noticed that they used the same chipset family (updated to the GP2015/GP2021 chipset). Because the SuperStar OEM board is relatively inexpensive and well built, it suggested the idea of hacking into the hardware to bypass the digital processing on the board and to connect directly into a PC. **Photo 2** shows the result. The OpenSource GPS website provides a description of how to do it. Because the SuperStar GP2021 interface is config-

ured like that of the Intel 80186 microprocessor, the hardware was easy to set up. Although the modification destroys a perfectly good receiver, it seems to work well. Versions of the hacked receiver are working at both the University of California, Riverside, and the University of New South Wales, Australia. With hardware now available for less than $200, adapting the software to the GP2021 was the only missing part.

Although hacking a receiver is a workable solution, we discovered that directly connecting the GP2021 data and address lines over long distances to the PC data bus created its own problems. In response, one of us (Baker) started a company, GPS Creations, to

produce and sell hardware designed to run the OpenSource GPS software. The product line consists of an RF board containing the GP2015 and GP2021 chips without any digital processing. The address and data lines are brought out to a connector. ISA and Peripheral Component Interconnect (PCI) interface cards into which the RF board plugs are also available. **Photo 3** illustrates the ISA version, and **Photo 4** shows the PCI version. The software has been configured to search for a PCI card. If the card is not found, it defaults to the ISA interface.

Current GPS Hardware. At the present time, we continue to use the Zarlink GP2015/GP2021 chipset. The GP2015 is a small-format, triple-
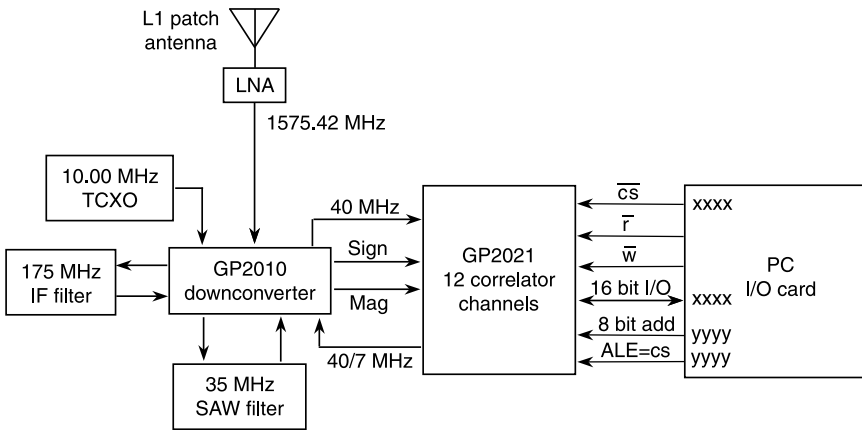
▲ **FIGURE 1** OpenSource GPS receiver hardware
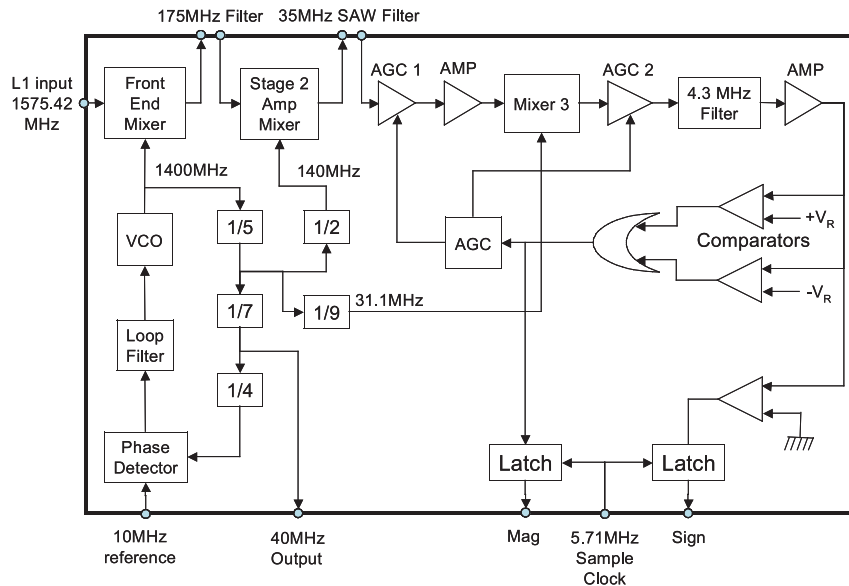
▲ **FIGURE 2** GP2015 block diagram

conversion, front-end chip (similar in performance to the GP2010 but in a smaller package). The chip features an on-board analog-to-digital converter or sampler that provides a two-bit quantized output for subsequent signal processing in the digital domain. The GP2021 is a 12-channel correlator chip. These chips constitute an old design (~1995 time frame); however, they are the only commercially available chipsets we have found that permit open access to the correlators. **Figure 1** is a block diagram of the OpenSource GPS receiver hardware. To date, we have not been successful at

obtaining any open-source assistance from any GPS semiconductor manufacturers.

Frequency Plan. The GPS receivers implemented with the GP2015/GP2021 chipset contain five signal-down-conversion stages. The first three stages are performed by mixers in the GP2015 front-end chip and are described generally in the GP2015 data sheet (see "Further Reading"). This triple-conversion design (see **Figure 2**) improves image rejection and permits user-designed filters to be installed between the conversion stages. Our current design uses a simple band-

pass filter between the first and second stages and a surface acoustic wave (SAW) filter between the second and third stages. The fourth down-conversion is digital and occurs in the sampler, which is driven by the 40/7-MHz (or 5.714-MHz) GP2021 sample clock (SAMP-CLK) output. The final down conversion occurs in the GP2021 correlator chip, where quadrature outputs of the carrier digitally controlled oscillator (DCO) are mixed with the sampled GPS intermediate frequency (IF) signal to produce the in-phase (I) and quadrature (Q) outputs at baseband.

The proper implementation of a GPS receiver requires a careful study of this frequency plan. Designers must pay particular attention to the exact IFs and to whether high-side or low-side mixing occurs at each down-conversion stage. These issues determine several phase-frequency relationships, which must be processed accurately to achieve maximum positioning accuracy.

In the usual case (low-side mixing), an RF signal is mixed with a local oscillator (LO) signal of lower frequency to produce a down-converted IF signal. For low-side mixing, a positive-phase change of the RF signal results in a positive-phase change of the IF.

In less-common cases (high-side mixing), an RF signal is mixed with a higher frequency LO signal. This result often occurs as a result of digitization of an IF signal at a rate higher than the IF, for which the sampling clock serves as an LO to produce a down conversion. The down-converted IF still is the frequency difference of the RF and LO, but positive-phase changes of the RF result in negative-phase changes at the IF, and vice versa. This phase reversal must be accounted for if it occurs. High-side mixing occurs in the fourth down-conversion stage of GP2015/GP2021 GPS receivers.

First stage: The 1575.42-MHz GPS signal is low-side mixed with a 1400-MHz (40 MHz $\times$ 35) LO signal to produce a 175.42-MHz first IF.

Second stage: The 175.42-MHz first IF
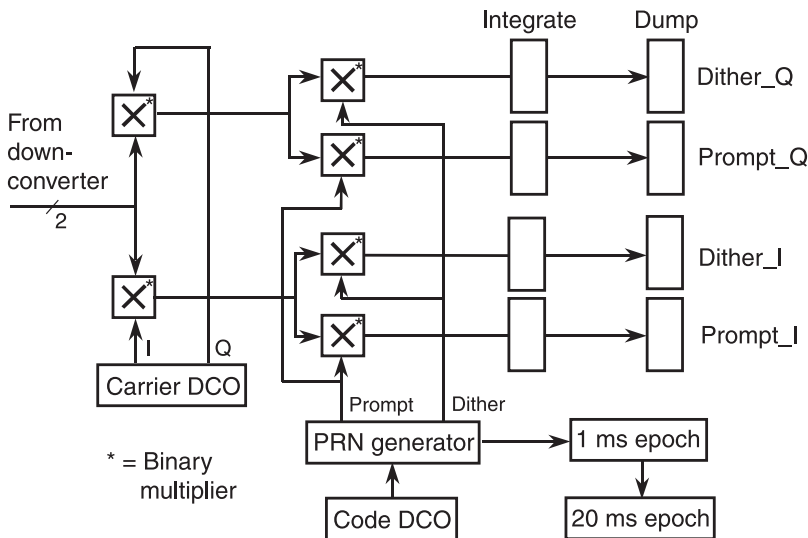
▲ FIGURE 3 Channel digital processing

signal is low-side mixed with a 140-MHz (1400 MHz/10) LO signal to produce a 35.42-MHz second IF.

Third stage: The 35.42-MHz second IF signal is low-side mixed with a 31.111-MHz (1400 MHz/45) LO to produce a 4.309-MHz (35.42 MHz – 31.111 MHz) third IF. The third IF frequency has a nominal value of approximately 4.31 MHz.

Fourth stage: The third IF signal is sampled by a 40/7-MHz clock. Over-sampling is a high-side mixing process, so a phase reversal occurs in this down-conversion step. The output of this process is a fourth IF signal with a nominal (no Doppler shift) frequency of exactly –1.405396825 MHz (35.42 MHz – 280/9 MHz – 40/7 MHz = –88540000/63 Hz = –1.405396825 MHz). The negative frequency indicates that a phase reversal has occurred.

Final stage: The fourth IF signal is sampled by the software-controllable carrier DCO clock. The output of this process is a final IF signal with a nominal frequency of f_{DCO} – 88540000/63 Hz, where f_{DCO} is the carrier DCO frequency set by the software. An additional phase reversal may or may not occur, depending on whether the DCO is set high or low, relative to the fourth-stage IF.

In-phase and quadrature samples are accumulated during 1-millisecond periods (I and Q correlator outputs, respec-tively). I and Q are quadrature samples of a residual frequency difference between the original GPS signal and the sum of all the LO frequencies described above. The GPS receiver software adjusts the carrier DCO to first drive this residual frequency difference to zero, and eventu-ally to achieve a near-perfect frequency lock to the signal. The nominal carrier DCO setting is 88540000/63 Hz, which would convert a 1575.42-MHz GPS sig-nal to zero frequency. **Figure 3** illustrates the final stage sampling and integration.

Software

OpenSource GPS is a C program written in C/C++ and compiles under Borland versions 4.5 and later, Microsoft Visual C++ 1.5, and DJ's GNU Programming Platform (DJGPP), a free C++ compiler. The software functionality extends from satellite acquisition and tracking to the decoding of the navigation message and to producing raw pseudorange and delta-pseudorange (carrier-phase) measure-ments.

We will continue with our discussion of the OpenSource GPS software in the February issue of *GPS World.* We will overview the operation of the software, including a description of the four states in which a tracking channel can find itself: the acquisition state, the confirma-tion state, the pull-in state, and the nor-mal tracking state. For a preview, consult the well-documented source code on

the project website.

We will also discuss testing of the OpenSource GPS platform in static and kinematic modes and outline some future plans for the project.

Acknowledgments

CLIFFORD KELLEY holds a B.S. in physics, a B.S. in electrical engineering, an M.S. in systems engineer-ing, and a Ph.D. in industrial and systems engineering, all from the University of Southern California, Los Angeles. He has been working on GPS software since 1995 and began offering it as OpenSource GPS in 2001.

DOUGLAS BAKER has a B.S. in electrical engineering from Vanderbilt University, Nashville, Tennessee, and has been actively working with GPS hardware design since 1987.

"Innovation" is a regular column featuring discussions about recent advances in GPS technology and its applications as well as the fundamentals of GPS positioning. The column is coordinated by **RICHARD LANGLEY** of the Department of Geodesy and Geomatics Engineering at the University of New Brunswick, who appreciates receiving your comments and topic sug-gestions. To contact him, see the "Columnists" section on page 6 of this issue.

Further Reading

For details on the operation of GPS and GPS receivers, see
Understanding GPS: Principles and Applications, second edition, edited by E.D. Kaplan and C.J. Hegarty, Artech House Publishers, Boston, 2005.

Global Positioning System Signals: Measurements, and Performance by P. Misra and P. Enge, Ganga-Jamuna Press, Lincoln, Massachusetts, 2001.

GPS: Theory and Practice, fifth edition, by B. Hoffmann-Wellenhof, H. Lichtenegger, and J. Collins, Springer-Verlag, New York, 2001

Global Positioning System: Theory and Practice, vols. I and II, edited by B.W. Parkinson, and J.J. Spilker Jr., American Institute of Aeronautics and Astronautics, Inc., Washington, D.C., 1996

Fundamentals of Global Positioning System Receivers: A Software Approach, second edition, by J. B.-Y. Tsui, John Wiley & Sons, New York, 2004

For details on the operation of GPS and GPS receivers, see
"OpenSource GPS: Open Source Software for Learning about GPS," by C. Kelley, J. Barnes, and C. Dingdong, published in the *Proceedings of ION GPS 2002,* the 15th International Technical Meeting of the Satellite Division of The Institute of Navigation, Portland, Oregon, September 24–27, 2002, pp. 2524–2533.

"OpenSource GPS: Open Source Software for Learning about GPS" by C. Kelley and D. Baker, presented at ION GNSS 2005, the 18th International Technical Meeting of the Satellite Division of The Institute of Navigation, Long Beach, California, September 13–16, 2005.

For detailed information on the GP2015/GP2021 chipset, see
GP2015: GPS Receiver RF Front End and GP2012: GPS 12-Channel Correlator data sheets, Zarlink Semiconductor, Ottawa, Ontario, Canada, 2005. (Also available on-line at *http://assets. zarlink.com/DS/zarlink_GP2015_MAY_05.pdf* and *http://assets. zarlink.com/DS/zarlink_GP2021_AUG_05.pdf*.)

INDEX

2006

››› FEBRUARY 21–23
Munich Satellite Navigation Summit 2006

Munich, Germany
For more information, contact *http://munich-satellite-navigation-summit.org*

Complete calendar at www.gpsworld.com.