this concept, the transceiver's software divides the computational tasks performed by the device into small subtasks and designates these tasks to the most suitable hardware module for the task. In a sense, the SDR's software performs micromanagement of the computed tasks well below the commonly used level of task assignment.

The EVA SDR also incorporates a dynamic waveform selection algorithm that minimizes the SDR's power consumption based on the channel QoS. To do so, the SDR collects the channel characteristics on a regular basis while in the normal mode of operation and adjusts the modulation type, RF channel data rate, output power level, and some other characteristics of the transmitted signal based on the gathered QoS. This method results in, essentially, 1000s of continuously changing waveforms.

In sum, the EVA SDR bridges the gap that has historically inhibited the coexistence of flexibility, power efficiency, and miniaturization in the same mobile handset radio.

This work was done by Aleksey Pozhidaev of Lexycom Technologies for Johnson Space Center. For further information, contact the JSC Innovation Partnerships Office at (281) 483-3809.

In accordance with Public Law 96-517, the contractor has elected to retain title to this invention. Inquiries concerning rights for its commercial use should be addressed to:

Lisa Livdahl, CFO
Lexycom Technologies, Inc.
425 South Bowen Street, Unit 1
Longmont, CO 80501
Phone No.: (303) 774-7822
URL: www.lexycominc.com

Refer to MSC-25125-1, volume and number of this NASA Tech Briefs *issue, and the page number.*

Remotely Accessible Testbed for Software Defined Radio Development

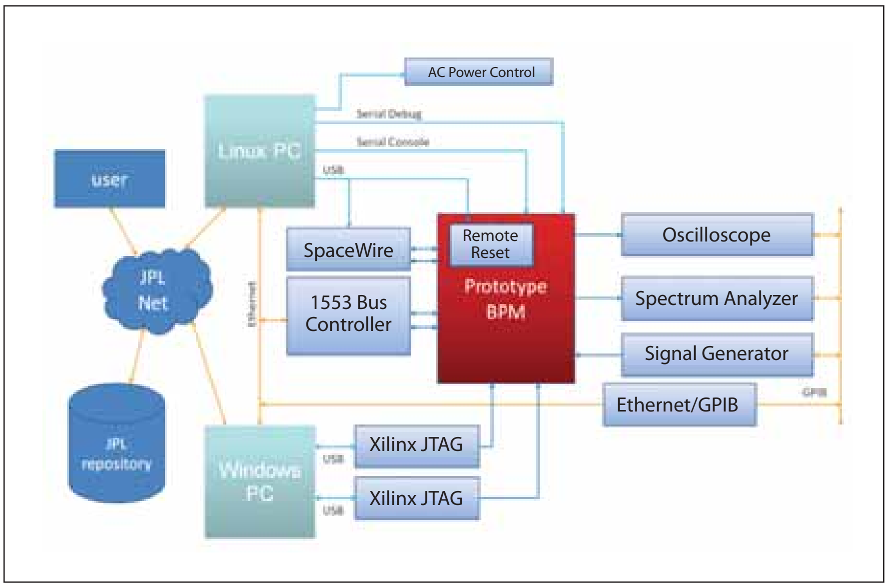
This testbed enables a geographically scattered development team to collaborate on a testbed.

NASA's Jet Propulsion Laboratory, Pasadena, California

Previous development testbeds have assumed that the developer was physically present in front of the hardware being used. No provision for remote operation of basic functions (power on/off or reset) was made, because the developer/operator was sitting in front of the hardware, and could just push the button manually. In this innovation, a completely remotely accessible testbed has been created, with all diagnostic equipment and tools set up for remote access, and using standardized interfaces so that failed equipment can be quickly replaced. In this testbed, over 95% of the operating hours were used for testing without the developer being physically present.

The testbed includes a pair of personal computers, one running Linux and one running Windows. A variety of peripherals is connected via Ethernet and USB (universal serial bus) interfaces. A private internal Ethernet is used to connect to test instruments and other devices, so that the sole connection to the "outside world" is via the two PCs.

An important design consideration was that all of the instruments and interfaces used stable, long-lived industry standards, such as Ethernet, USB, and GPIB (general purpose interface bus). There are no "plug-in" cards for the two PCs, so there are no problems with finding replacement computers with matching interfaces, device drivers, and installation. The only thing unique to the two PCs is the locally developed software, which is not specific to computer or operating system version. If a device (including one of the computers) were to fail or become unavailable (e.g., a test instrument needed to be recalibrated), replacing it is a straightforward process with a standard, off-the-shelf device.



The **Remotely Accessible Testbed** includes a pair of personal computers with a variety of peripherals connected via Ethernet and USB interfaces.

This strategy has paid off several times over the developmental effort. It made it very easy to construct a "portable" version of the testbed to take to a remote site to test the flight model radio: the two PCs were rented laptops, and copies of the required interface boxes were rented or borrowed. Everything was plugged together, the software was loaded, and a few hours later, testing could commence. Compared to the traditional approach of a rack full of customized interface drawers and customized PCs, it was much simpler and less expensive, as well as immediately responsive to changing project needs.

In fact, the experience of creating an *ad hoc* test capability at a remote site has resulted in some minor changes to the overall design: for example, rather than individual serial ports, the system now uses USB to serial adapters, all plugged into a USB hub, so there is a single USB connection to the Linux PC. Likewise, in the laptop configuration, the private internal network is implemented with USB-Ethernet adapters, because most laptops come with only one Ethernet interface.

This work was done by James P. Lux, Minh Lang, Kenneth J. Peters, and Gregory H. Taylor of Caltech for NASA's Jet Propulsion Laboratory. Further information is contained in a TSP (see page 1). NPO-48013