

Penn State Develops Software Defined Radio Ground Station for Nanosatellite

Software defined radio (SDR) technology holds great promise for enabling spacecraft engineers to create reconfigurable, lightweight communications systems that can be easily upgraded to add improved functionality and support new protocols. Sponsored in part by NASA, the LionSat mission will test Internet protocol (IP) communications for uplink and downlink to a spacecraft in low Earth orbit using an SDR ground station.

At The Pennsylvania State University, students used MathWorks tools and Lyrtech hardware to design, model, and simulate a hardware prototype of an SDR ground station for LionSat.

Dr. Sven G. Bilén, an associate professor of engineering design and electrical engineering at Penn State, is leading the initiative as well as a number of other SDR projects. He reports, “There are only a handful of satellites that have implemented IP for space-to-ground communications. With MathWorks tools and Lyrtech technology, we have contributed to increasing that number by rapidly developing an effective prototype and building up our competence in SDR technology.”



SDR ground station.

system, the group needed an integrated software and hardware system that enabled them to design, simulate, and test their systems quickly.

“Coding in VHDL takes a long time to learn,” explains Bilén. “We needed a turn-key solution that enabled us to develop systems at a high level, without getting bogged down in the details or in interfacing different products together.”

THE SOLUTION

Penn State used Simulink® to design, develop, and simulate SDR prototypes for the satellite ground station. Simulink, together with a Lyrtech SignalMaster system to digitize and process signals, provided them with a comprehensive development environment for the LionSat project. Because of the computational requirements of the algorithms, the prototype included a combination of TI DSPs and Xilinx FPGAs.

Using MATLAB® and Simulink with related toolboxes and blocksets, Bilén and his team explored different algorithm options, optimized system parameters, and evaluated design trade-offs.

THE CHALLENGE

To develop an SDR ground station for a nanosatellite

THE SOLUTION

Use Simulink® and Lyrtech SignalMaster™ to model, simulate, and generate code for complete SDR systems

THE RESULTS

- Rapid prototype
- Accurate simulations
- Flexible development and testing environment

THE CHALLENGE

A key technical challenge was programming the field programmable gate array (FPGA) that processes incoming signals. Writing and debugging code for an FPGA in very high-speed integrated circuits description language (VHDL) can be a time-consuming and error-prone process. To allow more time to focus on system design, Bilén needed an approach that minimized low-level programming.

To complete the ground station and other SDR projects on the same Lyrtech hardware

“MathWorks tools helped us to move from floating point to fixed point to get an accurate representation of the hardware,” says Jerker Taudien, a student involved in the SDR projects.

They used Real-Time Workshop® to generate code for the embedded DSP, a TI C6000™. Using Xilinx System Generator, they generated VHDL code from their Simulink models without any manual coding.

Taudien used Embedded IDE Link™ CC for TI’s Code Composer Studio™ to verify and debug their code on the DSP. He also used Communications Blockset™ to generate simulated modulated signals to debug and verify the systems.

MATLAB enabled Taudien to determine filter coefficients. He explains, “It was very easy to change filter responses with MATLAB. It also helped me plot data and visualize frequency variations.”

“In the next few months, we will validate our designs using real data,” says Bilén. “I’m confident that everything will work as planned.”

Penn State is currently planning an end-to-end communications test. If issues arise, the development environment will enable the team to quickly update the system.

Bilén is also planning to teach a special topics course on SDR, in which students will use Simulink to learn and apply SDR techniques on different SDR platforms.

“SDR opens opportunities for aspiring communications engineers,” Bilén says. “By using a high-level environment such as Simulink, I can focus the course on SDR architecture and applications, rather than the nitty-gritty of VHDL. This will also enable students who come with an electro-

“

With MathWorks tools and Lyrtech technology, we have an integrated solution that allowed us to prototype very quickly without programming our FPGAs at a low level. This enabled us to immediately begin exploring how to implement an overall system.

”

Dr. Sven G. Bilén, Penn State

magnetics and RF background, as I do, to get as much out of the course as those with a signal processing background.”

THE RESULTS

▪ **Rapid prototype.** “Using Simulink, I completed an SDR prototype in one month without writing any VHDL code,” says Taudien. “With Simulink and Xilinx System Generator, I just create the model, and generate all the VHDL code automatically from the model. That saved us months of time.”

▪ **Accurate simulations.** “I developed a complete simulation in Simulink, from input to output, which made it very easy to debug and improve the model. Moreover, after generating the code, it worked the first time. I expected more debugging, but everything that worked in simulation worked in real time,” reports Taudien.

▪ **Flexible development and testing environment.** Students shared a single hardware test environment because they could first design, develop, and simulate entire systems in Simulink. “With Simulink, you can switch easily between hardware-in-the-loop and computer-only simulation. You can use simulated signals, and then quickly change to using input from a real antenna or digitizer. That flexibility accelerates development,” says Bilén.

To learn more about the College of Engineering at Penn State, visit www.engr.psu.edu

APPLICATION AREAS

- Aerospace and defense
- Communications
- Modeling and simulation
- Signal processing

PRODUCTS USED

- MATLAB®
- Simulink®
- Real-Time Workshop®
- Embedded IDE Link™ CC
- Communications Blockset™
- TI’s Code Composer Studio™

www.mathworks.com

 The MathWorks™

© 2008 The MathWorks, Inc. MATLAB and Simulink are registered trademarks of The MathWorks, Inc. See www.mathworks.com/trademarks for a list of additional trademarks. Other product or brand names may be trademarks or registered trademarks of their respective holders.

91335v01 07/08