

Ono Sokki Reduces Development Time for Precision Automotive Speed Measurement Device



LC-8100 GPS Speedometer System.

To precisely gauge brake performance and fuel consumption when testing a prototype vehicle, engineers must track the vehicle's speed to within 0.1 km/h. Because many commercial speed measurement devices are based solely on global positioning system (GPS) technology and are therefore susceptible to multipath fading, they cannot reliably deliver the required level of accuracy. In road tests, for example, accuracy drops where the presence of buildings degrades GPS radio signals.

Engineers at Ono Sokki used MATLAB®, Simulink®, and Embedded Coder™ to develop a precision speedometer that uses an inertial measurement unit (IMU) in locations where there is not a strong satellite signal, eliminating the issue of multipath fading.

“This is the first project for which we used Model-Based Design for embedded system development,” says Kazuhiro Ichikawa, an engineer in the General Measurement Group at Ono Sokki. “Model-Based Design was much more effective than our traditional approach. Creating an executable specification and automatically generating production code enabled us to reduce development time significantly.”

The Challenge

While Ono Sokki engineers have extensive expertise in developing automotive measurement and testing instrumentation, they had no existing design models that could be reused for the precision speedometer. To achieve the required level of accuracy, the engineers needed to develop and implement

entirely new algorithms that involved complex matrix operations on multiple state variables.

Within the deadline to complete the project, the team needed to accelerate development of embedded C code without a reference design. In Ono Sokki's conventional workflow, the team wrote C code manually. However, the algorithms required for this effort were too complicated to code by hand, particularly given the tight schedule.

The Solution

Ono Sokki used MathWorks tools for Model-Based Design to complete the precision speedometer on schedule.

The engineers captured the behavior of the system in a flow chart and then used MATLAB to develop algorithms for each component block in the behavioral flow. They conducted simulations in MATLAB to verify functionality at the component level.

In Simulink, the team created a system-level model that incorporated the MATLAB based components, including complex timing characteristics and the interactions between components. They used this executable specification to perform numerous simulations, which enabled them to identify the set of parameter values that maximized system performance.

After verifying the design via simulation, the team generated C code from their models using Embedded Coder and implemented it on a DSP.

The Challenge

Develop a high-precision speedometer for prototype vehicles within a tight schedule

The Solution

Use MathWorks tools for Model-Based Design to develop algorithms, generate production code for an embedded processor, and perform processor-in-the-loop verification

The Results

- Development time cut significantly
- Code base reduced dramatically
- System model reused as a test bench

“With MathWorks tools we have a seamless environment for development, simulation, code generation, and processor-in-the-loop verification. The advantages of Model-Based Design over hand-coding in C can’t be overestimated.” —KAZUHIRO ICHIKAWA, ONO SOKKI

The team performed processor-in-the-loop (PIL) verification of the implementation using the IDE link feature in Embedded Coder. They reused their Simulink model as a test harness to verify the execution of the code on the DSP target.

The engineers found a discrepancy in the DSP execution results caused by the compiler that could have gone undetected in their traditional workflow. PIL testing with the IDE link enabled them to identify and resolve the issue before the code went into production.

The project is currently moving into production, and Ono Sokki engineers are reusing their models as the basis for the next generation of the device.

The Results

Development time cut significantly. Ono Sokki estimates that they would not have been able to complete the project on time using their conventional development approach and writing the C code themselves. Using Model-Based Design and automatically generating the C code, the team completed the project well ahead of schedule.

Code base reduced dramatically. The engineers implemented the same algorithms in MATLAB using much less code than they estimated would be required using hand-written C code. “Because there is much less code, it is easier to maintain, and there are fewer bugs,” says Ichikawa.

System model reused as a test bench. “We captured the entire system-level design in Simulink, which enabled us to see how changes to one component affected the whole,” notes Ichikawa. “We verified the behavior of our algorithms in Simulink and later used the same model as a test bench for PIL testing, which helped us improve the quality of the embedded software.”

Industries

- Automotive
- Electronics and semiconductors

Application Areas

- Algorithm development
- System design and simulation
- Embedded code generation
- Verification, validation, and test
- Digital signal processing

Products Used

- MATLAB®
- Simulink®
- Embedded Coder™
- MATLAB Coder™
- Simulink Coder™

Learn More About Ono Sokki

www.onosokki.net