

IAV Designs a Universal ECU for On-Target Rapid Prototyping with MathWorks Code Generation Tools

The increasing complexity of new and unique automotive features has forced engineering firms to find more efficient ways to address the design requirements in concurrent customer projects. To reduce development time and meet aggressive customer deadlines, these firms must explore innovative technology and techniques.

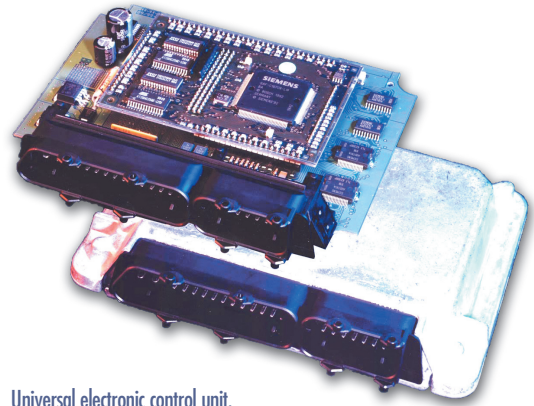
With more than 2,000 employees, automotive engineering firm Ingenieurgesellschaft Auto und Verkehr (IAV) in Berlin, Germany develops vehicles, engines/drives, and automotive electronics for leading car manufacturers and their suppliers. Using MathWorks tools, the company is addressing their customers' needs by developing and prototyping vehicle systems with a universal electronic control unit (ECU) that supports on-target rapid prototyping.

With on-target rapid prototyping, IAV can better investigate the impact of changing requirements, explore new ideas, and validate algorithms on production processors. The universal ECU enables IAV to support multiple projects and target multiple processors simultaneously.

“Using Simulink® and Real-Time Workshop® Embedded Coder for on-target rapid prototyping with our universal ECU has proven successful in many of our customer projects,” says Martin Richter, department manager of automotive electronics systems, IAV. “The integration of MathWorks tools with ECU hardware establishes a complete development environment for designing and deploying all of our embedded software.”

THE CHALLENGE

IAV set out to develop a general-purpose ECU that would let them quickly prototype



Universal electronic control unit.

new algorithms on target microprocessors. The ECU would need to serve multiple customers working on a variety of projects in areas such as powertrain controls, chassis and safety systems, and emerging by-wire controls systems.

Because customers often select different microprocessors, and only after the functional requirements have been fully defined, the ECU had to support multiple mass production microprocessors, each with a variety of target-specific input/output (I/O) modules and peripheral devices.

IAV sought a set of unified software tools that could speed up their development process and enable them to perform on-target rapid prototyping quickly using functional specification models.

THE SOLUTION

IAV used MATLAB®, Simulink, and Stateflow® to design models and create algorithms, and Real-Time Workshop Embedded Coder to automatically generate code. To meet project, application, and performance requirements, the engineering team at IAV developed a flexible, reusable architecture for the universal ECU.

THE CHALLENGE

To develop a general-purpose ECU that targets various microprocessors

THE SOLUTION

Use MathWorks production code generation tools to develop a flexible, reusable ECU in an on-target rapid prototyping environment

THE RESULTS

- Flexible system created for ECU development projects
- Productivity and team workflow improved
- Development time reduced

“ On-target rapid prototyping is an emerging technology that is being adopted by our ECU customers in Germany. The MathWorks open code-generation architecture is a key part of our universal electronic control unit development efforts. ”

— Martin Richter, IAV

With support for multiple microprocessors, the universal ECU features a common hardware abstraction layer (HAL), a configurable task scheduler, and CAN Calibration Protocol support for calibration and tuning.

Working with MathWorks consultants, IAV developed embedded targets for deploying their algorithm concepts to specific target microprocessors. The universal ECU currently supports three different CPU modules: Infineon C167, Motorola MPC555, and Infineon TriCore.

IAV created a common HAL to integrate the modules and manage the interfaces to the controller I/O. They used template files to create custom I/O blocks within Simulink. The templates also ensured that the necessary I/O device driver interfaces would be included in the code generated by Real-Time Workshop Embedded Coder. Because they used a common HAL, IAV avoided writing extra code for each microprocessor and simplified the data exchange between the simulation and the controller.

The team then created a configurable scheduler as a custom Simulink S-function block to support both simulation and code execution. The scheduler simulated Simulink and Stateflow components as individual tasks using function-call subsystems. They then used Real-Time Workshop Embedded Coder to automatically generate code for the subsystems and created the scheduler for the ECU using task configuration parameters.

IAV chose the CCP to calibrate and tune control parameters in real time using standard calibration tools, including ETAS INCA and Vector CANape. They then created custom CCP modules to obtain I/O signals and cali-

brate block parameters from the Simulink block diagram while the application ran on the target system.

THE RESULTS

- **Flexible system created for ECU development projects.** With the design of the universal ECU complete, IAV engineers can now easily deploy their Simulink and Stateflow applications on a choice of CPU modules by selecting the appropriate system target file from Real-Time Workshop targets prior to code generation.
- **Productivity and team workflow improved.** Using Simulink as the primary source for gathering system requirements, IAV improved design, coding, and calibration while improving productivity and team workflow.
- **Development time reduced.** Using Simulink to develop the I/O drivers and Real-Time Workshop Embedded Coder to generate embedded code saved IAV from hand coding and customizing the code for each microprocessor.

To find out more about IAV, visit www.iav.de

www.mathworks.com

APPLICATION AREAS

- Automotive
- Automatic code generation
- Algorithm development
- Modeling and simulation

PRODUCTS USED

- MATLAB
- Simulink
- Stateflow
- Real-Time Workshop
- Real-Time Workshop Embedded Coder