

Motorola Creates Electric Vehicle Battery Management Controller with Real-Time Workshop® Embedded Coder

Electric cars can travel moderate distances on a single battery charge and maintain average cruise speeds of 60 mph, making them appealing options for inner-city commuters. These battery-powered vehicles require a sophisticated electronic control unit (ECU) to regulate the flow of power from the batteries to the motor.

The Motorola Automotive Group recently developed and optimized software for a battery management controller using model-based design and automatic code generation products from The MathWorks.

“We would almost certainly not have been able to meet our deadlines if we’d used paper designs and hand coding instead of Simulink® and Real-Time Workshop® Embedded Coder,” says Salam Zeidan, software manager at Motorola Automotive.



The Motorola electronic control unit.

turnaround times, so we decided to use that approach here,” says Zeidan. “We selected the complete set of MathWorks products to avoid piecing tools together and working from a disjointed process.”

The team used Simulink and Stateflow® to model the control algorithms in floating point and then converted the resulting models into fixed point using the Fixed-Point Blockset.

They encapsulated Stateflow and Simulink submodels within parent Simulink models and created separate models based on functionality and execution rate. Using Real-Time Workshop Embedded Coder, they then generated C code for the HC12 embedded target microcontroller directly from these models. “The code generation process and the integration of the controller code with the rest of the module code was very smooth because we had control over the way the code was generated and over the early planning of the interfaces,” says Zeidan.

They then performed software integration tasks, such as interfacing the hardware I/O layer’s handwritten code with the application layer’s automatically generated code.

THE CHALLENGE

To develop battery management controller software within a tight deadline

THE SOLUTION

Use integrated model-based design and code generation tools from The MathWorks to design, test, and manage requirements for the controller

THE RESULTS

- Automatic generation of efficient C code
- Optimized memory resources
- Ability to detect design flaws before generating code

THE CHALLENGE

Developing the controller was difficult because the interfaces among various system components were still fluid and would change.

The engineers had to manage the memory resources for the HC12 microcontroller without exceeding 8KB of RAM. They also had to ensure that all the code would fit into 128KB of flash memory. The entire project had to be completed within a tight time frame.

THE SOLUTION

Motorola Automotive had just completed a production-intent ECU for a global body seat controller on a model year 2002 vehicle using MathWorks model-based design and code generation products. “We knew that model-based design is well suited for managing changes in requirements and fast

For the calibration phase, they applied an improved data-fit function that automatically selected the best tradeoff between the maximum tolerable error and the number of table data points and marked the selected optimal point on a graph. They then generated an optimal data table from the graph.

“Using the Model-Based Calibration Toolbox and the Curve Fitting Toolbox allowed us to fit the models optimally and greatly reduced the memory needed for calibration parameters. This was critical for our memory-constrained automotive ECU application,” says Zeidan.

To validate the design against the customer’s requirements, the engineers associated the model components to the written requirements with the Requirements Management Interface. “Internal reviews were then easy, and we could demonstrate to our customer that all the requirements had been met,” Zeidan explains.

The source control interface in MATLAB® enabled them to use Rational ClearCase to perform common project management activities directly from within the MATLAB and Simulink environment.

Motorola Automotive may implement system tests using other MathWorks products, including xPC Target and xPC TargetBox™, for future automation of system validation and software bench tests. xPC Target enables a Simulink and Stateflow model to be simulated in real time on an external PC, enabling hardware-in-the-loop testing of the actual ECU system against the model of the physical system being controlled.

APPLICATION AREAS

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

PRODUCTS USED

- MATLAB
- Simulink
- Stateflow
- Fixed-Point Blockset
- Real-Time Workshop Embedded Coder
- Model-Based Calibration Toolbox
- Curve Fitting Toolbox
- Requirements Management Interface

“We would almost certainly not have been able to meet our deadlines if we’d used paper designs and hand coding instead of Simulink and Real-Time Workshop Embedded Coder.”

Salam Zeidan, Motorola

THE RESULTS

■ Automatic generation of efficient

C code. “The automatic scaling capability of Real-Time Workshop Embedded Coder saved us much effort by helping us to systematically define the data types and binary points of signals throughout the fixed-point models. The fixed-point code generation also uses an advanced expression-folding approach and has capabilities that were not previously available to us,” says Zeidan.

■ **Optimized memory resources.** By explicitly defining global variables rather than using the default structure-based code, Motorola optimized memory resources. Real-Time Workshop Embedded Coder generated compact fixed-point code that fit within the existing 128KB of flash and 8KB of RAM.

■ **Ability to detect design flaws before generating code.** “Simulating the models early allowed us to perform most feature tests in the simulation environment before the embedded code was generated,” says Zeidan. “If defects were discovered during system testing, we could quickly recreate the conditions from the test bench using log files as inputs to the Simulink models.”

To learn more about the Motorola Automotive Group, visit www.motorola.com

www.mathworks.com