

A **Dearborn Electronics** White Paper
on DE Model Based Automotive
Embedded Design Solutions



DE Model Based Automotive Embedded Design Solutions

Dearborn Electronics
#687, 16th Main, 4th T Block,
Jayanagar, Bangalore - 560041
Phone: (91) 80 22440025, 22445466
Website: <http://www.deindia.com>
E-mail: sales@deindia.com

Introduction

Modeling and Simulation is a discipline for developing a level of understanding of system as a whole and the interaction of the parts of a system. A model is a simplified representation of the actual system intended to promote understanding. Simulation helps us to learn implications of the interactions in the model elements during run time based on the events and helps us to get adequate level of understanding of the system and refine the model to achieve the expected behavior.

Model-Based Design can help you to evaluate system designs, prototype products, and deploy to production targets. The increasing complexity of automotive systems, considering the safety needs and the pressure to deliver these systems to market faster is driving the need for new engineering processes.

This document provides DE's expertise in providing model based automotive design solutions.

DE Expertise

Dearborn Electronics (DE) is actively participating in providing Model based solutions with its 7+ years of experience in embedded software solutions and tools to leading automotive companies for the development of E/E systems and software.

DE provides model based design and development with compliance to auto code generation and customer guideline with the use of Matlab/Simulink/StateFlow.

DE has used model based design for development of

- CEM (Central Electronic Module) to handle the Body Control functionalities,
- Rapid Prototyping of Instrument Cluster
- Fuel Efficient Power Steering System (FEPS)
- Smart Power Distribution Junction Box
- MATLAB Interface for Gryphon.

DE Offerings

DE can provide,

- Model based system design solutions
- Models for the Legacy Code
- Model Verification
- Enhancement of Models
- AUTOSAR component development

DE has expertise team which can provide efficient solution and service to customer in appropriate cost with high quality and from last few years we are working with leading OEM to provide our solution and they appreciated our expertise and on-time delivery of the projects.

Development Life Cycle

Iterative V Model Product Development Life cycle is followed

Phases of Software Development Life Cycle:

- Requirements study and analysis. Participate in requirements review with requirements team.
- Design and Model Development
- Design and Model Reviews as per Guidelines
- Unit Testing
- Auto Code Generation using RTW (with MISRA compliance)
- Model Code Integration Testing
- Integration Testing

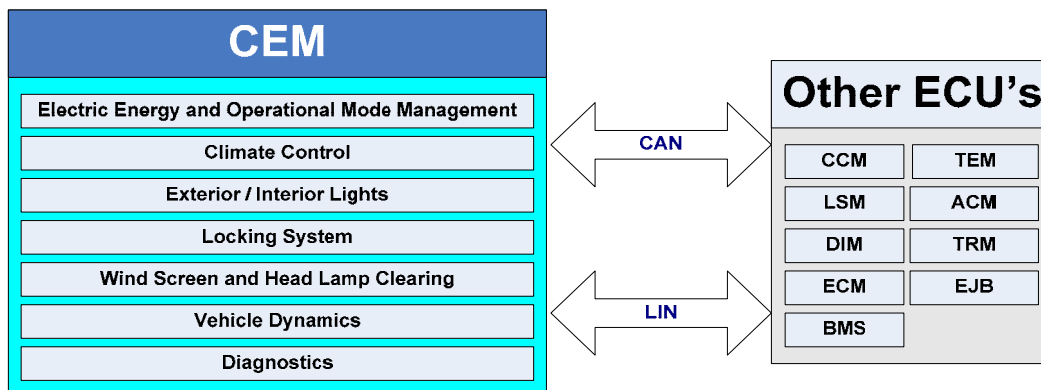
Validation Life Cycle:

- Requirements study and analysis. Participate in requirements review with requirements team.
- Validation Protocol Development
- Software Validation on Target

References / Case studies

1. Central Electronic Module (CEM)

DE has used model based approach for development of CEM to handle the Body Control functionalities as seen in the below diagram



The CEM is one of the key and smart ECUs in the vehicle to control following Body Control functionalities:

- Electrical Energy Management.

- Climate control.
- Locking System.
- Theft and Access Protection.
- Exterior Lights / Interior Lights.
- Visibility.
- Immobilizer control.
- Driver Information.
- Vehicle Dynamics.

Description:

Electrical Energy Management system shall control load and send warnings by receiving information from the ECM regarding the alternator. It also calculates maximum allowable current consumption of the climatic loads, it sends a message to DIM to display certain symbols and messages if any warnings are detected during key off functions to monitor the battery status. It communicates with BMS over LIN in all power modes, to control regenerative charging with battery status information from BMS. This system also implements diagnostics and car configurations.

Climate Control functions shall include air condition system control, heater control and blower fan control functionalities.

The visibility function area contains the functionality related to visibility aspects of the car user like windscreen, headlamp cleaning and windscreen defroster through LIN. Also implements CAN communication and part II diagnostic requirement for visibility.

Exterior Light functions include general functionality and bulb failure warning, Dipped/Main Beam Head Lights, Position Lights, Front Fog Light / Rear Fog Light, Stop Light / Emergency Brake Light, Reversing Light, Turn Indication / Hazard indication, Approach Light / Home Safe light and Auxiliary.

Locking system shall include,

- Door locks
- Central Locking
- Remote Control

Vehicle Dynamic functions include

- AirBag
- Active Yaw Control
- Cruise Control and CADS System
- Collision Mitigation by Braking (CMbB)
- Engine Management: Either Clutch Switch ClutchTOTSwitch (digital) and/or Clutch Pedal Sensor (analogue) is connected to CEM
- Engine Off Time

Driver Information system shall include Distance, Vehicle Speed, Fuel Level, Washer Fluid Level and Brake Fluid Level Indications.

Tools Used:

- MATLAB/Simulink/Stateflow, RTW (Auto Code Generation), Volcano (OSEK based Network Management), BDM (Trace 32), Freescale Code Warrior (Compiler), Vector CANoe / LIN. Proprietary RTOS
- Unit Testing and Model Code Integration Testing (MCIT) (MxVDev)
- Configuration management (DOORS, CM Synergy)

2. Instrument Cluster Prototyping

With increase in demand to bring in more information for Driver Assistance and to increase the safety, Instrument cluster implementation is becoming more complex. The Key to reduce the effort involved and time frame for the programs, DE has started with Model based approach to implement the features on instrument clusters using the MATLAB / SIMULINK / Stateflow. DE is involved in creating the models for different features like Telltales, Chimes, and Display...

Tools Used:

- MATLAB/Simulink/Stateflow, RTW (Auto Code Generation), DI Kernel Proprietary RTOS, BDM (Trace 32), CANalyzer with CAPL, LabVIEW
- Unit Testing and Model Code Integration Testing (MCIT) (MxVDev)
- Configuration management (Clear Case)

3. Smart Power Distribution Junction Box

This project is a prototype ECU developed as a proof of concept to implement a smart ECU in the vehicle. This project has following functionality to be implemented.

Power and Energy, Lighting, Locking, Alarms, Tire Pressure Monitoring, Warnings and Chimes, Global Window, Closure Interfaces, Body Features (2nd or 3rd row seat control, brake shift interlock feature, central configuration global real-time counter...), Operational Mode Management (power mode, car mode), Vehicle Starting Control, Common Input Processes (ignition input processing, door ajar input processing, driver door key switch input process, trunk key switch input process, voltage range monitor and input interface reference), Network Interfaces (MS-CAN network signal processing, HS-CAN network signal processing, lin network signal processing, gateways... etc.), diagnostics...

Tools Used:

- MATLAB/Simulink/Stateflow, RTW (Auto Code Generation), Volcano (OSEK based Network Management), BDM (Trace 32), Freescale Code Warrior (Compiler), Vector CANoe / LIN. Proprietary RTOS
- Unit Testing and Model Code Integration Testing (MCIT) (MxVDev)
- Configuration management (DOORS, CM Synergy)

4. Fuel Efficient Power Steering (FEPS)

In the Fuel Efficient Power Steering (FEPS) system, the flow of hydraulic fluid, which is responsible for supplying the assist force is provided and managed by a hydraulic pump and actuator assembly. The fluid is routed through the gear valve via the hydraulic lines. The amount of fluid supplied by the pump and actuator assembly is directly controlled by an Electronic Control Valve based upon system inputs, which include the steering wheel position, the steering rotation rate (rate of rotation of the hand-wheel by the driver), the vehicle speed, engine RPM, ignition status, and electrical system voltage.

While performing the primary function to control the current to the FEPS actuator in order to generate a desired level of pump flow, the module must also perform the following secondary functions:

Perform run time internal diagnostics to detect and mitigate severity of module failures

- Perform run time diagnostics on input signals to ensure that external input data are valid
- Store diagnostic error codes to assist service diagnostics
- Protect system from out of range vehicle signals (Reverse Battery, Double Battery...)
- Communicate with calibration tools to allow control and diagnostic parameters and software to be changed
- Communicate with Service Diagnostic tools particular to OEM customer (some functions include reporting status of internal self diagnosis stored codes, Flash Programming, and reporting operation state).

Tools Used:

MATLAB/Simulink/Stateflow

5. MATLAB Interface for Gryphon

The project involves the design and development MATLAB Interface to communicate Gryphon hardware through a Gryphon DLL. Gryphon is a hardware interface that has Ethernet interface and can communicate with vehicle protocols like CAN, J1850 and LIN.

Tools Used:

Matlab/Simulink/Stateflow

Summary

How DE Helped:

- Requirements study and analysis, Req. review with customers.
- Design and model development of feature.
- Unit Testing (using MxVDev)
- Model Code Integration Testing (MCIT using MxVDev)
- Integration Testing.

Benefit to Customer:

- Cost effectiveness solution,
- Follows UML Notations
- Guided process using the tools,
- Reduction in Human Errors,
- Target Independent development,
- Need not to wait for the hardware till Integration testing on H/W,
- On computer Simulation to correct the system pitfalls at early SDLC stages,
- High Reusability,
- Rapid Requirement changes adoption,
- Auto code generation capability in compliance to MISRA standards,