Stateflow autocode generation and deployment in embedded application

Manoj E S
08/04/2014
Agenda

- About Visteon
- Automotive HMI trends & challenges
- Need for visual authoring tools
- Stateflow in automotive HMI
- Use case study results
Product Portfolio

HVCC

- Heating, ventilation and air conditioning (HVAC)
- Compressors
- Powertrain cooling
- Fluid transport
- Electric vehicle and hybrid thermal systems

Visteon®

- Electronics
  - Audio & infotainment
    - Audio head units
    - Infotainment
    - Audio components
  - Information and controls
    - Instrument clusters
    - Displays
    - Climate controls
    - Decorative control panels

Visteon®

- Interiors
  - Cockpit modules
  - Instrument panels
  - Door panels and trim
  - Floor consoles
A Strong and Growing Customer Base
Driving Electronics Technology

Advanced Instrument Clusters

• Third generation reconfigurable clusters
• Leading 3-D graphics
• Superior display resolution

Next Generation Technologies

• Consumer electronics-driven user interaction
• OEM styling freedom
• Optimized viewability

Open Architecture Infotainment

• Contextual, intuitive HMI
• Car-to-Cloud connectivity
• “SmartScreen” smartphone integration
HMI Content in automotive

- Automobile OEM desire to bring Consumer Electronics features has resulted in more HMI content in automotive entertainment and driver information systems
  - Exponential growth of HMI content
  - Development time has almost halved
In order to contain the growing HMI content and reduced development timeline, it is necessary to adopt tool chains into the development cycle.
Software Architecture

- HMI Architecture adapted for
  - Use of tool chains
  - PC based development
  - Seamless integration in to target hardware
Significance of visual authoring tools

- With the increased use of TFT displays in the automobile, HMI has become a brand identity and a differentiating factor for OEMs

- OEMs want to closely monitor the development of HMI
  - Through periodic reviews with PC executables
  - Continuously make changes to optimize user interface and to keep up with market trends

- Conventional development methods cannot support this quick turn around

- Hence it is important to utilize visual authoring tools in HMI design
  - Remove hardware dependency / Develop in PC
  - Quickly incorporate changes, provide executable to OEMs
HMI and State Machine

- HMI follows a state machine principle
  - In the current state, information is either presented to user or a user response is expected
  - The next state is determined in response to a user action or a system event
- Simulink/Stateflow provides an easy to use interface to,
  - Model the behavior aspect of HMI software (using Stateflow)
  - Connecting behavior model with third party graphics display generation blocks and test input panels (using Simulink)
  - Simulate HMI functioning in PC with in Simulink
  - Generate PC executable with help of Simulink Coder
  - Generate code for embedded target with Embedded coder
Modeling the HMI with Simulink/Stateflow

- Simulink/Stateflow allows to model the HMI and simulate and test in a PC environment

Graphical Test Input Panel

HMI behavior model

Custom Code Framework

2012 FORD C520 System Modeler Simulation

Model, Test, Debug and Simulate

Visteon®
PC Simulation / Autocode Generation

Stateflow®
State Diagram Design

Simulink Coder
PC Simulation

Auto code Generation
Cross Compile & Download

Graphic Resource
Source Code
Custom Code
Ui Framework
Target Hardware
Practical Use case Study

- Implement a system check menu for an instrument cluster
- Each system check item follows the same state behavior
- Rather than modelling each item individually, create state machine model for generic system check item
- The properties of the individual system check items has been formed as a C data structure array and included as custom C code
- Resulted in ~4KBytes ROM requirement which is comparable with manual implementation ~3.8Kbytes
Simulink vs Manual Coding

- It is beyond doubt that Simulink modelling can bring down cost and time w.r.t software engineering.

- The biggest question – “is it a viable option when it comes to feasibility in terms embedded target resource (ROM/RAM) requirement?”

- Our experience is that the design strategy has a great dependency on the embedded target resource requirement, be it Manual coding or Simulink modelling.

- Once the right approach is chosen for a design problem, we can get comparable results with Simulink modelling.

- Our own experience is that using Stateflow along with custom C code is the most efficient way to extract the benefits of both worlds.

---

Visual Authoring & Autocode: Development time can be reduced by 75%