Live Debugging of Stateflow Charts While Running on ECU

Amjad Elshenawy & Mohammad Raouf

Mathworks Automotive Conference 2015
Stuttgart, Sep., 24th 2015
# Table of contents

1. Testing & Debugging Techniques in MBD 3
2. Focus on « Simulink External Mode » 12
3. Valeo Solution in Details 17
4. Conclusion 25
Testing & Debugging Techniques in MBD

Focus on « Simulink External Mode »

Valeo Solution in Details

Conclusion
Testing in Model Based Design

- What happens if a test case fails?
- How to debug in X-In-the-Loop testing?
- How to debug on Vehicle?
Debugging in Model-In-the-Loop (MIL) Testing

**Simulink Blocks**
- Signal Analysis
  - Scopes and Signal Viewers
  - Signal Logging
  - etc …

**Stateflow Charts**
- Stateflow Chart Animation
- Stateflow Breakpoints and Watch Data
Debugging in Software-In-the-Loop (SIL) & Processor-In-the-Loop (PIL) Testing

**Simulink Blocks**
- Signal Analysis
  - Scopes and Signal Viewers
  - Signal Logging
  - etc …

**Stateflow Charts**
- Stateflow Chart Animation in ”External Mode”
  - Chart local data can be viewed on signal viewers by designating them to be test points

Some Limitations Exist for internal Signal Logging
Debugging in Hardware-In-the-Loop (HIL) & On Vehicle Testing

- **Design phase**
  - Identify debugging variables
  - Declare debugging variables as “Global Variables” (they will have Fixed Memory Address)

- **Validation phase**

![Diagram](image-url)
Debugging in Hardware-In-the-Loop (HIL) & On Vehicle Testing

Simulink Blocks

- Data acquisition and signal plotting
- Parameter tuning
Manually create a debugging variable representing chart states
Stateflow Charts – The Classical Way

- Visualize the value of the state variable using the test tool

- Run the test case and monitor the state variable

<table>
<thead>
<tr>
<th>Name</th>
<th>Acq</th>
<th>Value</th>
<th>Unit</th>
</tr>
</thead>
<tbody>
<tr>
<td>Oil sActrTstBB1</td>
<td>✔</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Oil bActrTstInProgsBB1</td>
<td>✔</td>
<td>-</td>
<td>bool</td>
</tr>
<tr>
<td>Oil bBlowBy1ReqActrTst</td>
<td>✔</td>
<td>-</td>
<td>bool</td>
</tr>
</tbody>
</table>

![Debugging in Hardware-In-the-Loop (HIL) & On Vehicle Testing](image)
Debugging in Hardware-In-the-Loop (HIL) & On Vehicle Testing

Stateflow Charts – The Classical Way

- Drawbacks of the Classical Way of Debugging Stateflow Charts
  - Manually Define State Variables
    - Extra development effort is required
    - Additional memory consumption
  - Poor Visualization of Statecharts; No Statechart Animation

- Isn’t there a better way to debug Stateflow charts running on ECU?
  - Valeo solution with visualization and animation of Stateflow models based on “Simulink External Mode” will be presented
1. Testing & Debugging Techniques in MBD

2. Focus on « Simulink External Mode »

3. Valeo Solution in Details

4. Conclusion
In “External Mode”, Simulink algorithm is executed outside Simulink environment. Simulink is merely a GUI for:

- Visualizing Data
- Acquiring Signals
- Tuning Parameters (Provided that parameters are not inlined)
Configure Code Generation in External Mode

- In the model «Code Generation» configuration, configure:
  - «System target file» as «ert.tlc»
  - «Interface» as «External Mode»
  - «Transport Layer» as «tcpip» or «RS-232 (serial)»
Steps to start External Mode communication

1. **Build the Target Executable**
   - Both code and external executable are generated

2. **Run the External Executable**

3. **Select Simulink simulation as “External” Mode**

4. **Connect to the Target**

5. **Start Simulation in External Mode**
External Mode Remarks

- External mode uses “Code Instrumentation”
  - Includes extra header files
  - Adds code for data exchange and for transport layer
  - Adds extra variables

- ERT supports only two transport layers, namely, TCP/IP and RS-232 (serial). No direct support for automotive communication protocols.
1 Testing & Debugging Techniques in MBD
2 Focus on « Simulink External Mode »
3 Valeo Solution in Details
4 Conclusion
To prepare environment, 3 steps are needed:

1. Stateflow Model
2. Generated code from Stateflow Model
3. CAN Bus Interface

Visualization / Animation

ECU (running in the car)
Use Embedded Coder (ert.tlc) to generate a Global structure “DW_<modelName>”, containing active state of each state machine.

```c
typedef struct {
    ...
    uint8_T is_StateChart_1; /* ... */
    uint8_T is_StateChart_2; /* ... */
    uint8_T is_StateChart_3; /* ... */
    uint8_T is_StateChart_4; /* ... */
    ...
} DW_<modelName>;
```

No need for code instrumentation on Target!
1st Step: Exchanging Model States Info from ECU to PC

- Add XCP/CCP and CAN drivers on laptop
- Configure reception of data in Valeo Tool

Configurations:
- Baud Rate
  -- XCP/CCP
  -- CAN/LIN/FlexRay
  -- Struct Memory address

ECU (running in the car)

Generated code from Stateflow Model

XCP / CCP Driver
CAN Driver
Valeo Tool
CAN Bus Interface
2nd Step: Exchange Received States Info to Simulink Model

- Modify generated External-Mode files:
2nd Step: Exchange Received States Info to Simulink Model

- Modifications in details:
  - Comments all direct assignation to the states values
  - Generates additional C files that directly update the states value with the values received from ECU

```c
/* Entry: ActrTstBlowBy/F02_BlowBy2Tst/F01_BlowBy2TstChart */
/* ValeoTool Comment: ActrTstBlowBy_AUTOCODE_DWork.is_active_c1_ActrTstBlowBy_AUTO = 1U */

/* Entry Internal: ActrTstBlowBy/F02_BlowBy2Tst/F01_BlowBy2TstChart */
/* Transition: '<S8>:10' */
/* ValeoTool Comment: ActrTstBlowBy_AUTOCODE_DWork.is_c1_ActrTstBlowBy_AUTO/OIL_ACTRTEST_BB2_IDLE */

/* Entry 'OIL_ACTRTEST_BB2_IDLE': '<S8>:1' */
Oil_stActrTstBB2 = OIL_ACTRTEST_BB2_IDLE;
```
3rd Step: Final Setup

- Generate executable “Simulink-to-ECU Communication”:
Then, launch Stateflow Debugging Activity

- User requests to start communication
- Valeo Tool starts EXE
- Simulink Model initiates communication
- ECU requests states value
- Simulink-to-ECU Communication EXE sends states value
- Exchanges value for animation
- ECU requests states value
- Simulink-to-ECU Communication EXE sends states value
- Exchanges value for animation
- ECU requests states value
- ECU sends states value
- Exchanges value for animation
- User requests to stop communication
- Valeo Tool kills EXE

Property of Valeo. Duplication prohibited
1. Testing & Debugging Techniques in MBD
2. Focus on « Simulink External Mode »
3. Valeo Solution in Details
4. Conclusion
Conclusion

- Using “Simulink External Mode”, and with our in-house tool, we were able to read Statechart information in real-time from the ECU and accordingly animate Stateflow charts on PC.

- This technique facilitates a lot debugging of Statecharts for on-vehicle tests.

- Next step is to support FlexRay and LIN communication protocols.