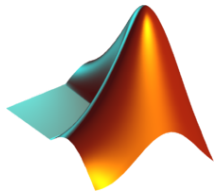
GEEDS - Group Electronics Expertise and Development Services



Live Debugging of Stateflow Charts While Running on ECU

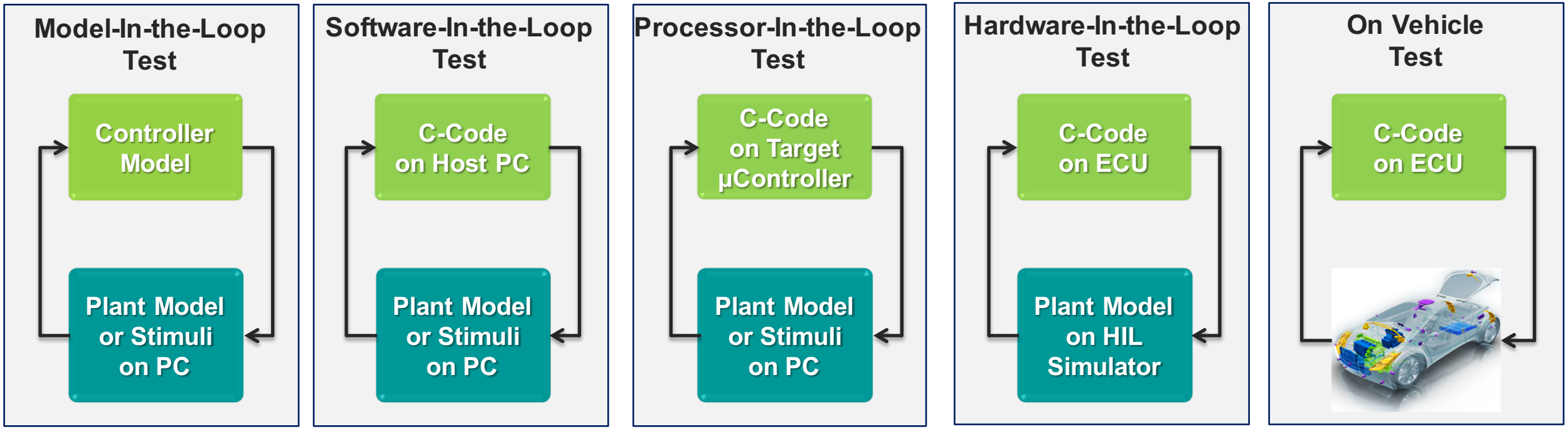
Amjad Elshenawy & Mohammad Raouf

Mathworks Automotive Conference 2015
Stuttgart, Sep., 24th 2015

September 2015

Table of contents

Testing in Model Based Design

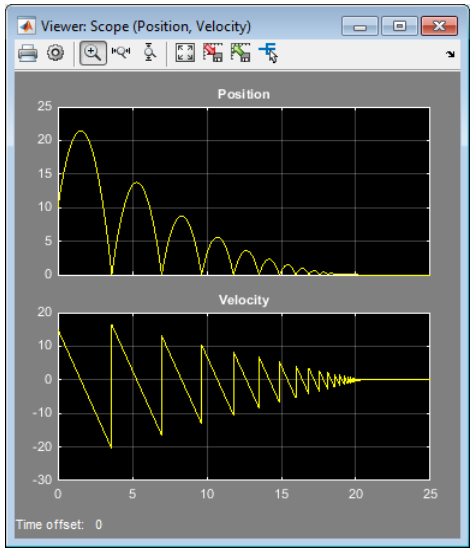


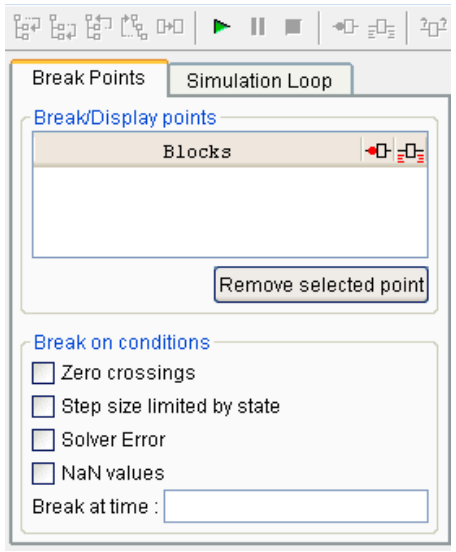
- What happens if a test case fails?
- How to debug in X-In-the-Loop testing?
- How to debug on Vehicle?

Debugging in Model-In-the-Loop (MIL) Testing

Simulink Blocks

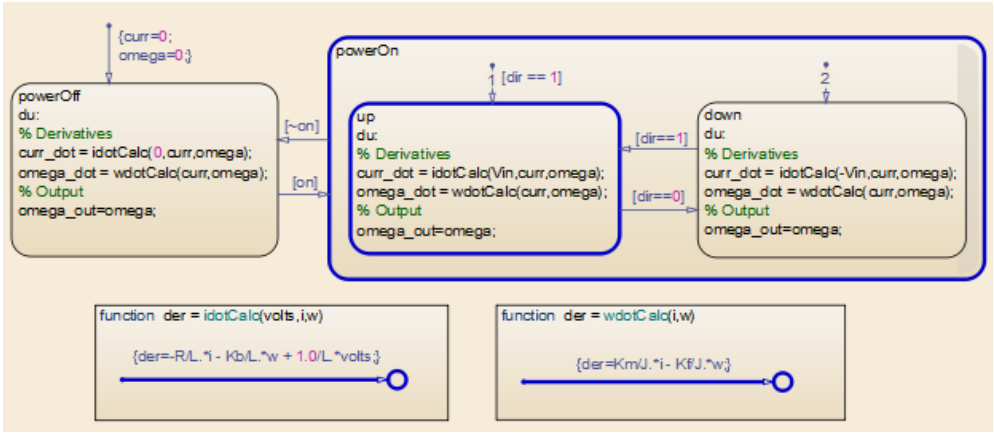
- Signal Analysis
  - Scopes and Signal Viewers
  - Signal Logging
  - etc …





Stateflow Charts

- Stateflow Chart Animation
- Stateflow Breakpoints and Watch Data



Debugging in Software-In-the-Loop (SIL) & Processor-In-the-Loop (PIL) Testing

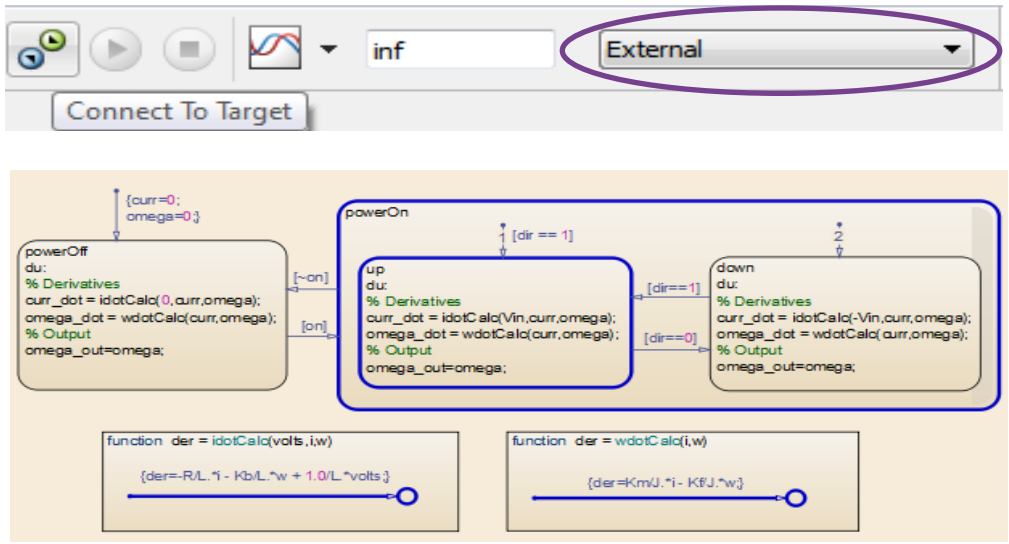
Simulink Blocks

- Signal Analysis
  - Scopes and Signal Viewers
  - Signal Logging
  - etc …

Some Limitations Exist for internal Signal Logging

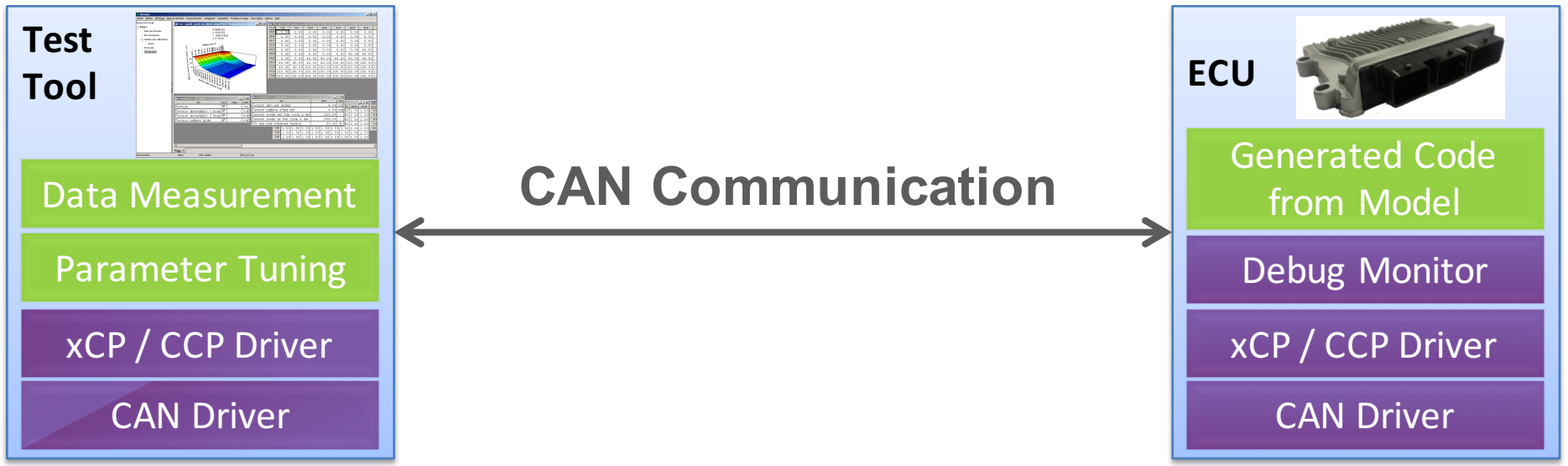
Stateflow Charts

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



Debugging in Hardware-In-the-Loop (HIL) & On Vehicle Testing

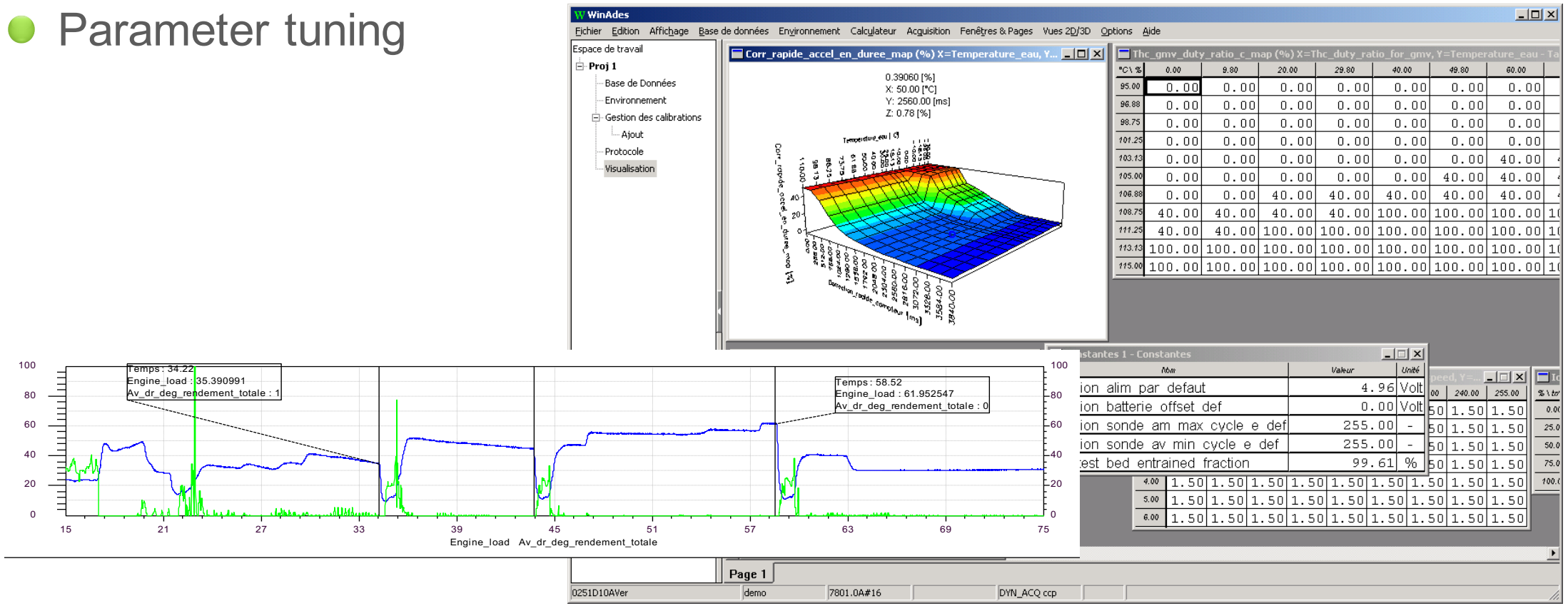
- Design phase
  - Identify debugging variables
  - Declare debugging variables as “Global Variables” (they will have Fixed Memory Address)
- Validation phase



Debugging in Hardware-In-the-Loop (HIL) & On Vehicle Testing

Simulink Blocks

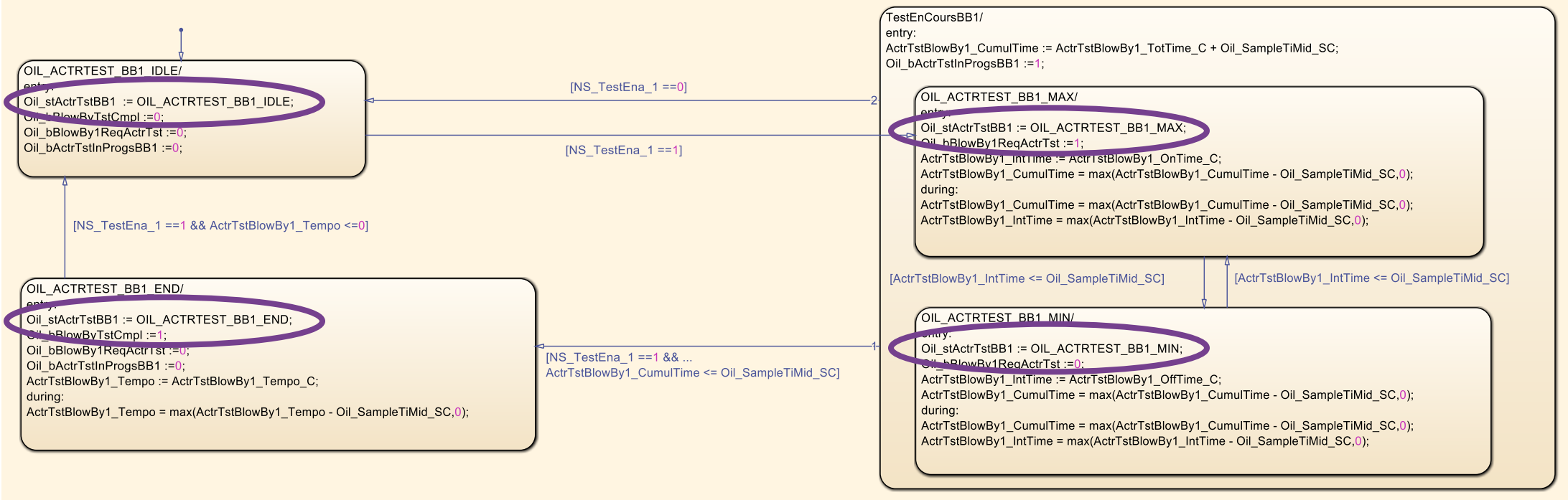
- Data acquisition and signal plotting
- Parameter tuning



Debugging in Hardware-In-the-Loop (HIL) & On Vehicle Testing

Stateflow Charts – The Classical Way

- Manually create a debugging variable representing chart states

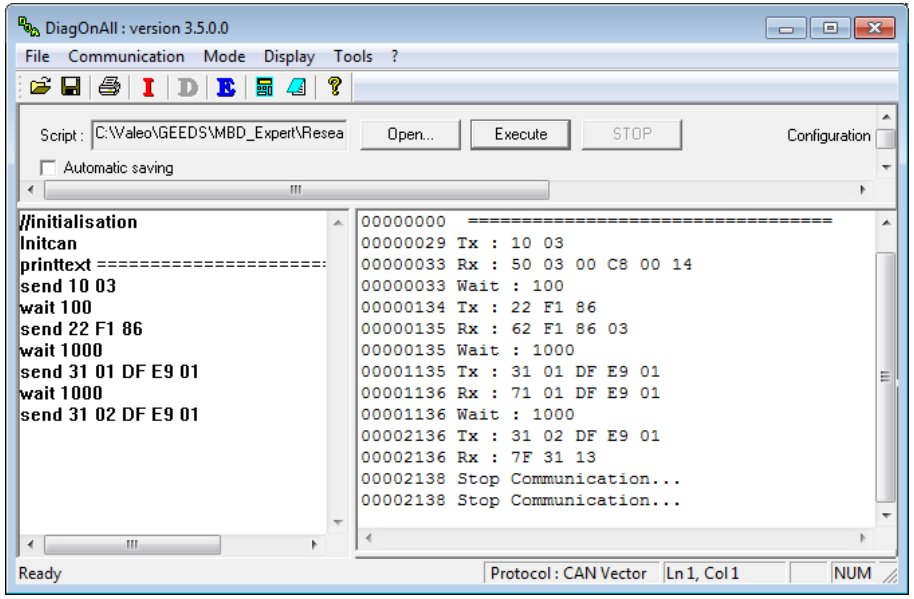


Debugging in Hardware-In-the-Loop (HIL) & On Vehicle Testing

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

Name	Acq.	Value	Unit
Oil stActrTstBB1	☑	–	-
Oil bActrTstInProgsBB1	☑	–	bool
Oil bBlowBy1ReqActrTst	☑	–	bool



Name	Acq.	Value	Unit
Oil stActrTstBB1	☑	2.00	-
Oil bActrTstInProgsBB1	☑	0	bool
Oil bBlowBy1ReqActrTst	☑	0	bool

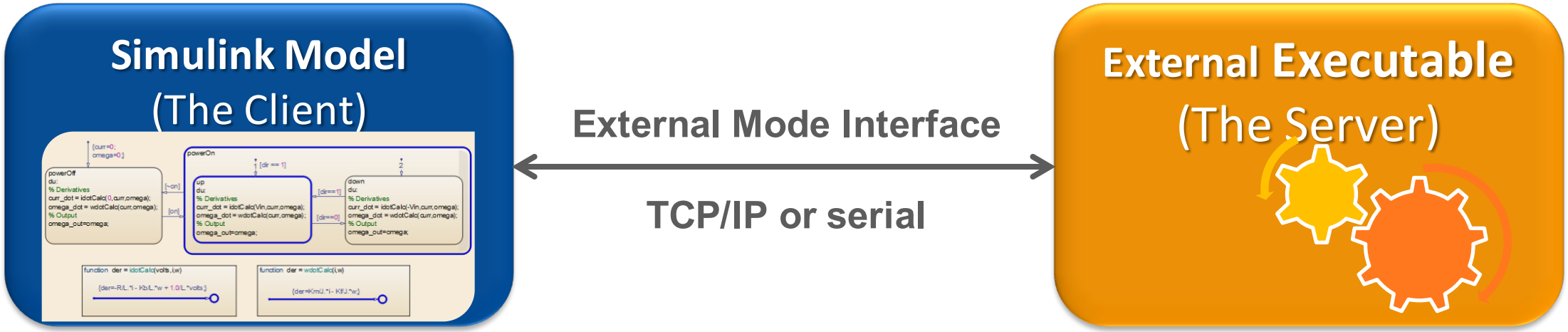
Debugging in Hardware-In-the-Loop (HIL) & On Vehicle Testing

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

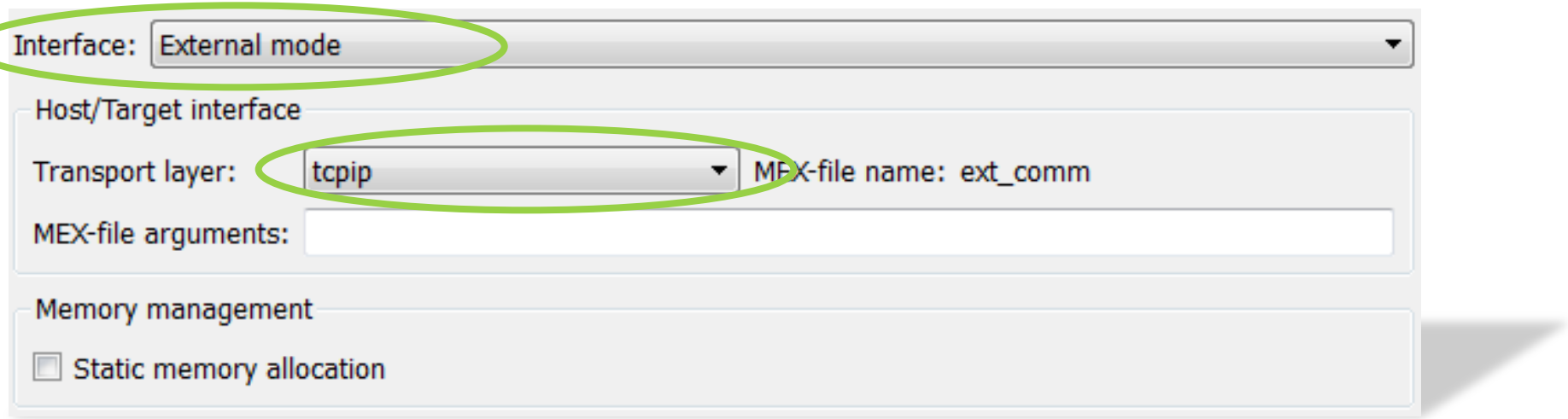
Overview of Simulink External Mode

- 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)



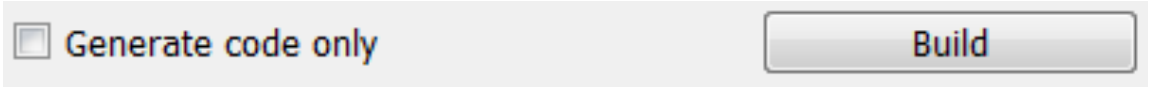
Configure Code Generation in External Mode

- 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

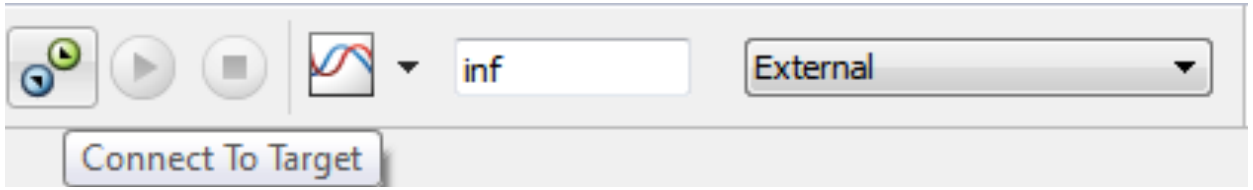
1. Build the Target Executable



   - 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

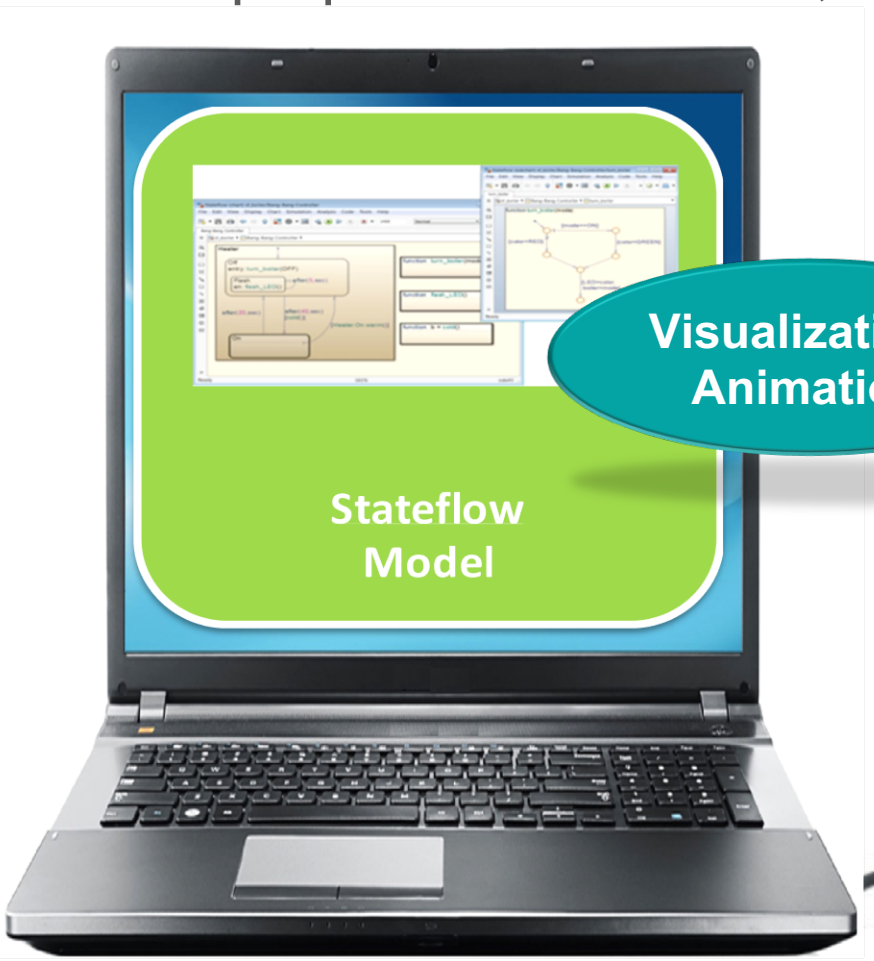
External Mode Remarks

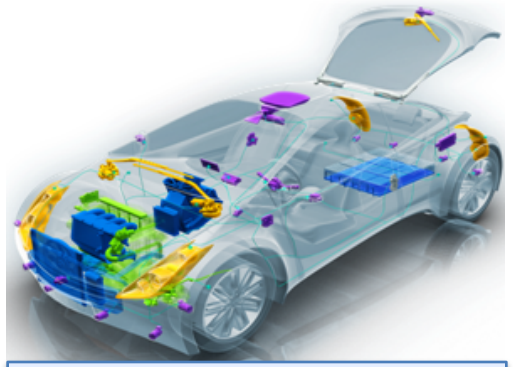
- External mode uses "**Code Instrumentation**"
  - Includes extra header files
  - Adds code for data exchange and for transport layer
  - Adds extra variables
- ERT supports only two transport layers, namely, TCP/IP and RS-232 (serial). **No direct support for automotive communication protocols.**

Valeo

Valeo Solution in Details

- To prepare environment, 3 steps are needed





ECU *(running in the car)*

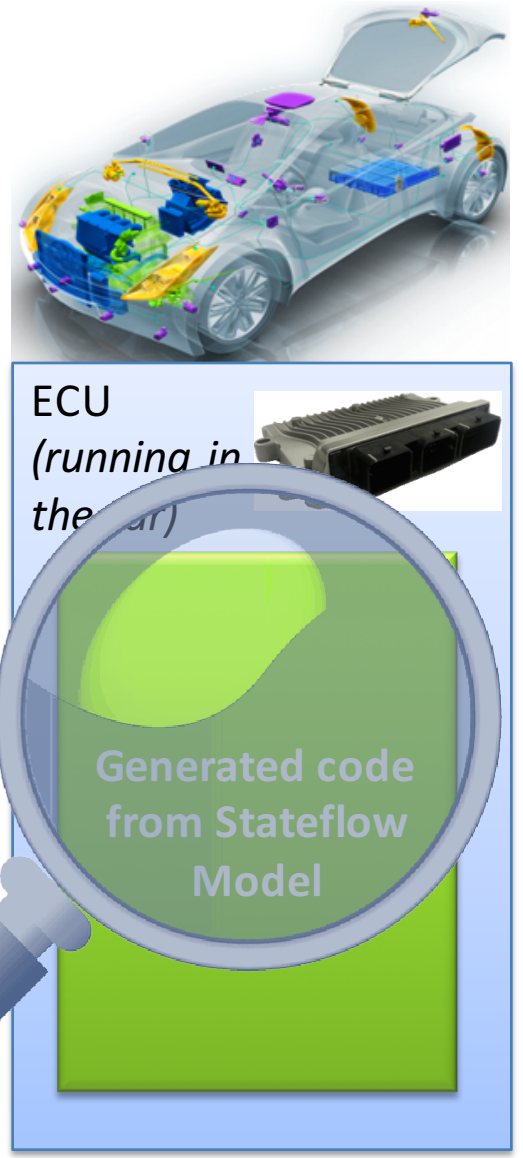
Generated code from Stateflow Model

CAN Bus Interface

Model State info on ECU

- Use Embedded Coder (ert.tlc) to generate a Global structure "DW_<modelName>", containing active state of each state machine.

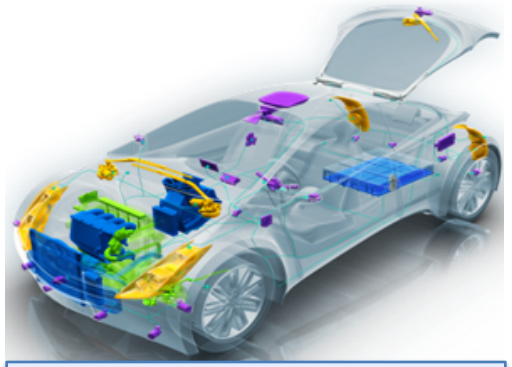
```
typedef struct {
...
uint8_T is_StateChart_1;                /* ... */
uint8_T is_StateChart_2;                /* ... */
uint8_T is_StateChart_3;                /* ... */
uint8_T is_StateChart_4;                /* ... */
...
} DW_<modelName>;
```



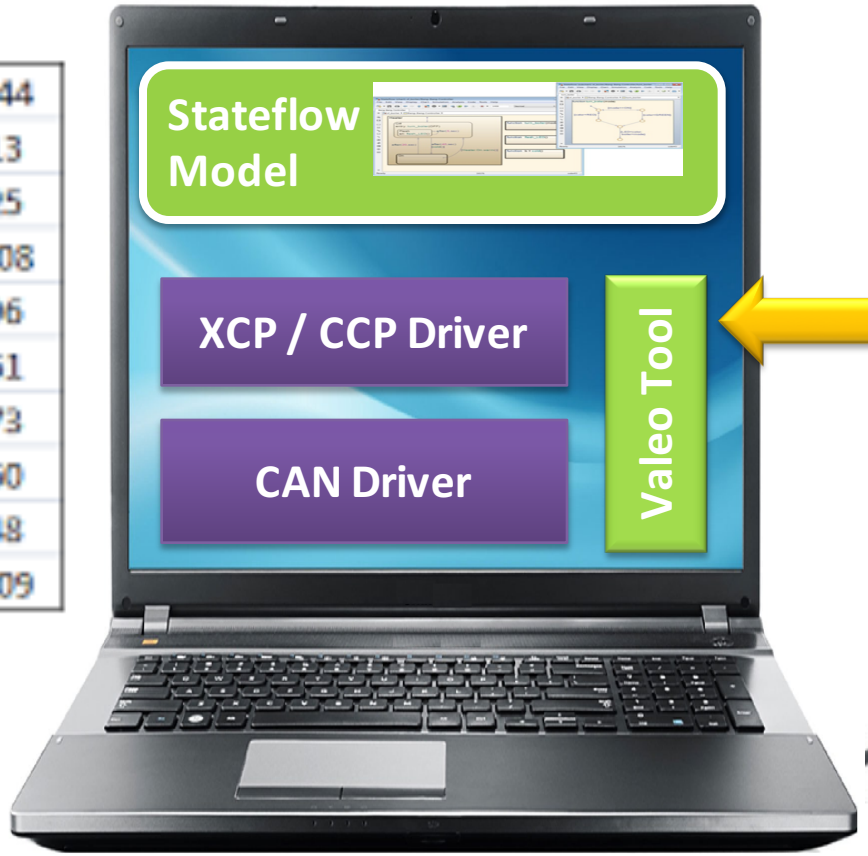
No need for code instrumentation on Target !

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



144
13
25
108
96
61
73
60
48
109



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

CAN Bus Interface

ECU *(running in the car)*

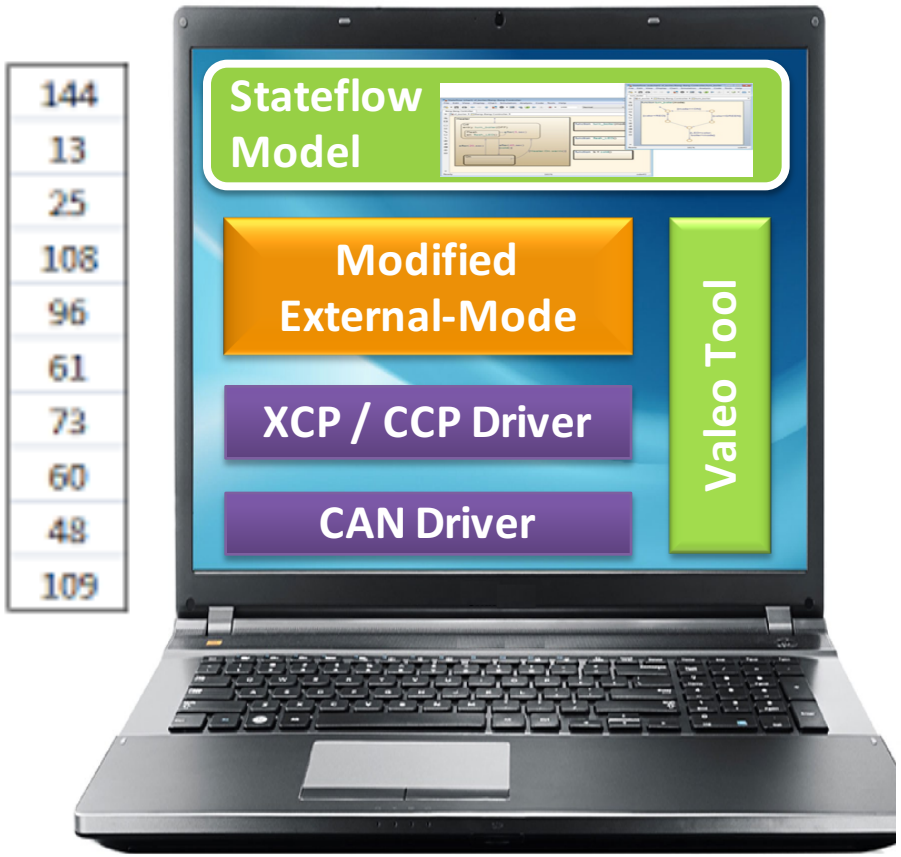
Generated code from Stateflow Model

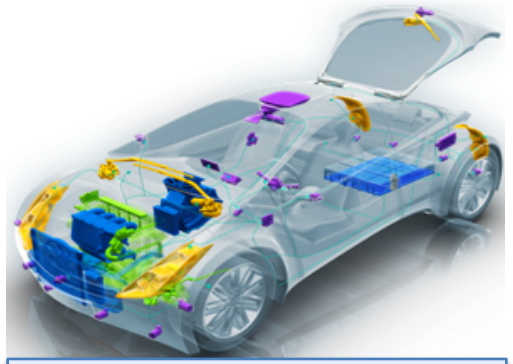
XCP / CCP Driver

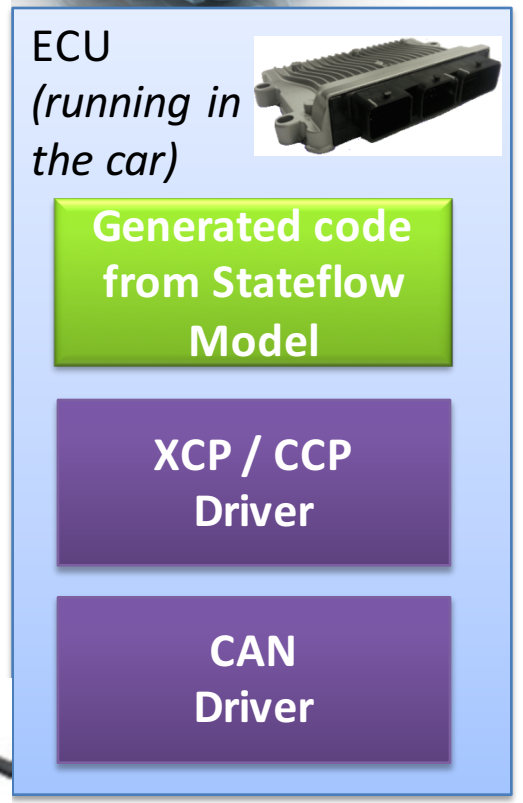
CAN Driver

2nd Step: Exchange Received States Info to Simulink Model

- Modify generated External-Mode files:

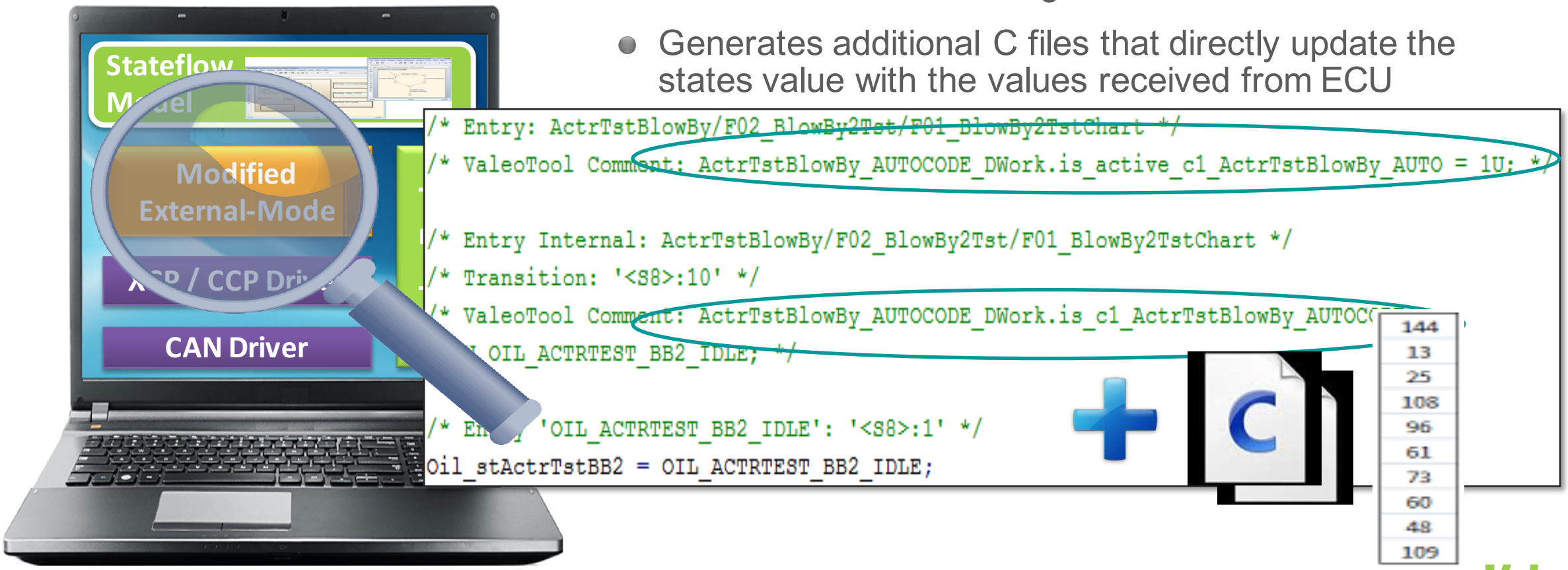






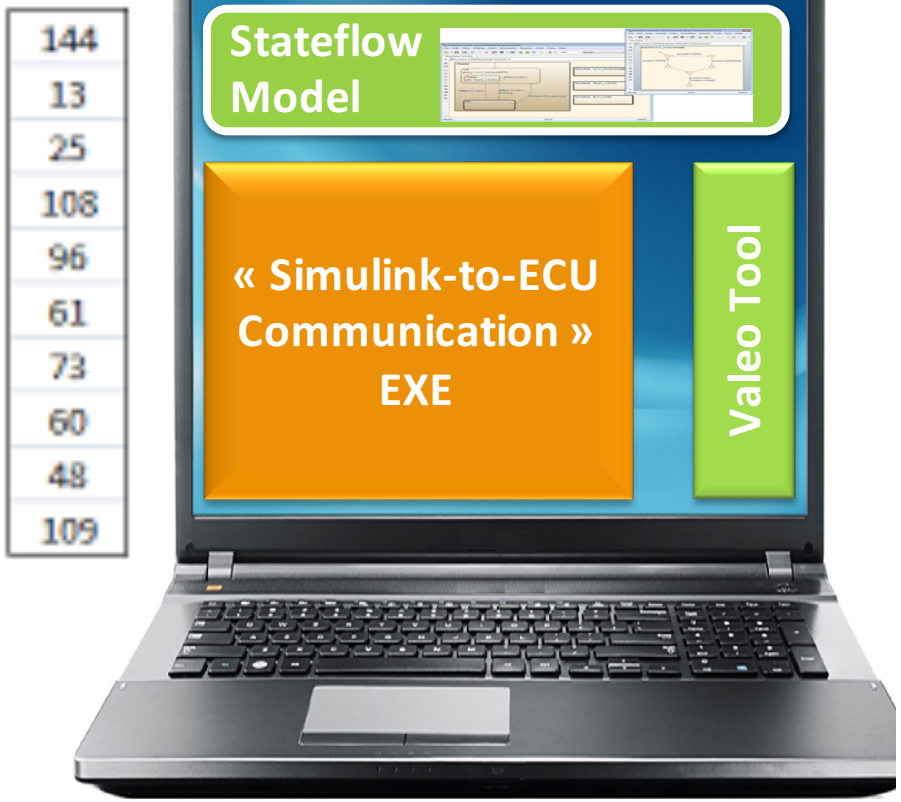
2nd Step: Exchange Received States Info to Simulink Model

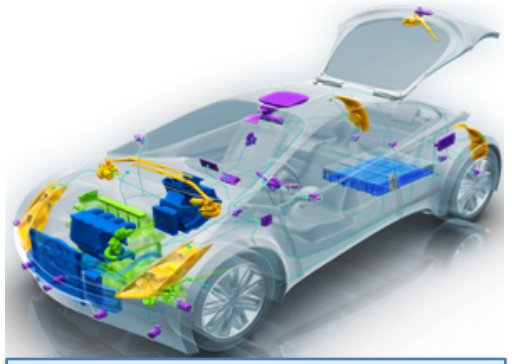
- Modifications in details:
  - Comments all direct assignation to the states values
  - Generates additional C files that directly update the states value with the values received from ECU

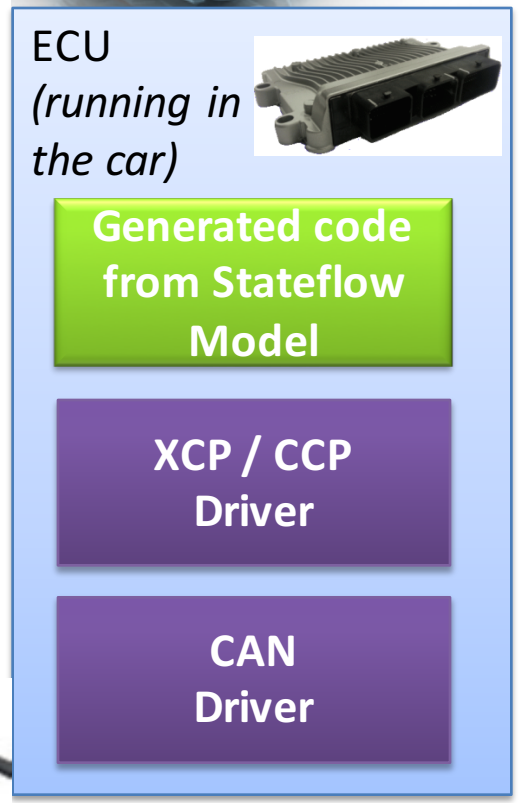


3rd Step: Final Setup

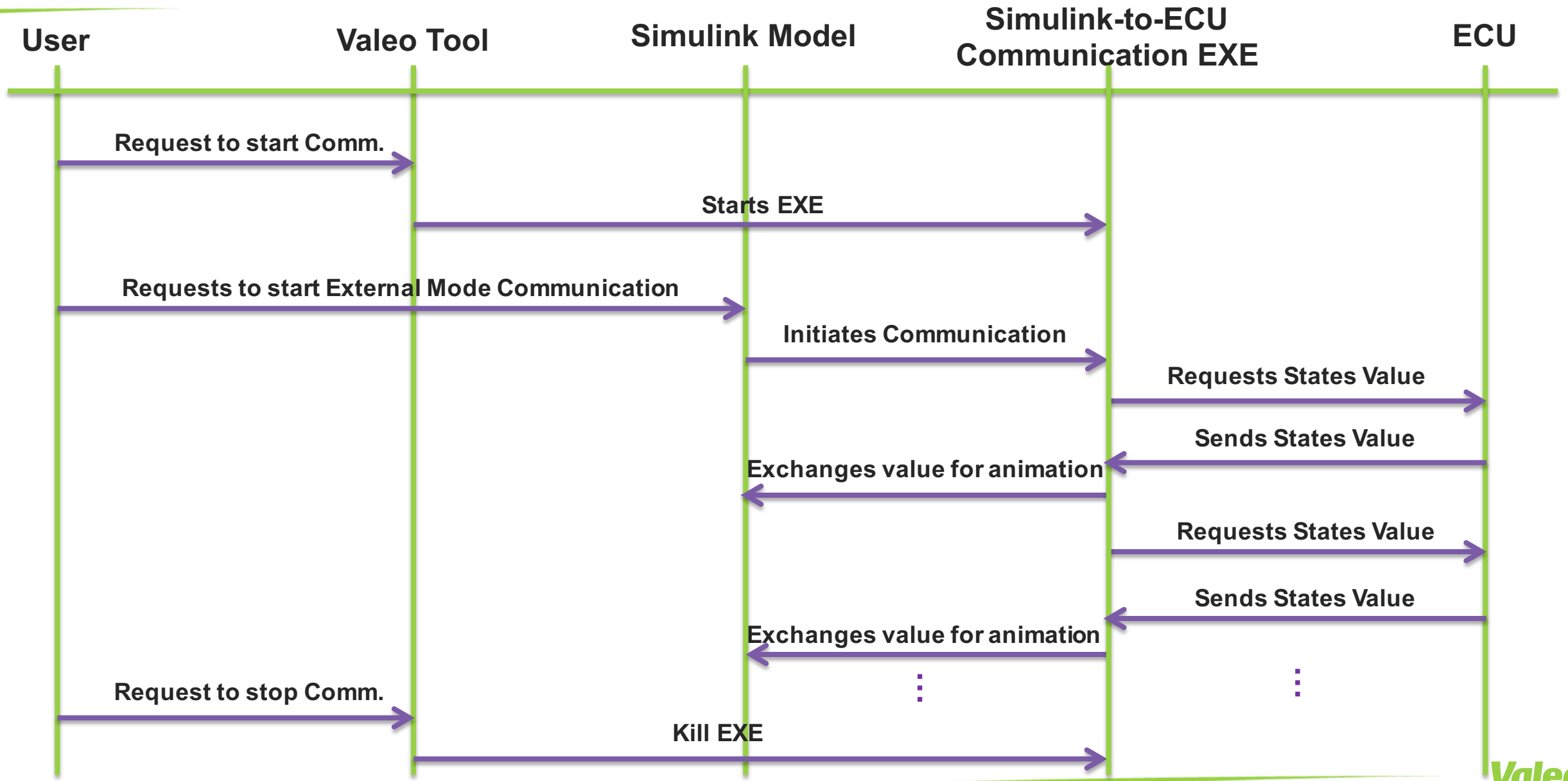
- Generate executable “Simulink-to-ECU Communication”:





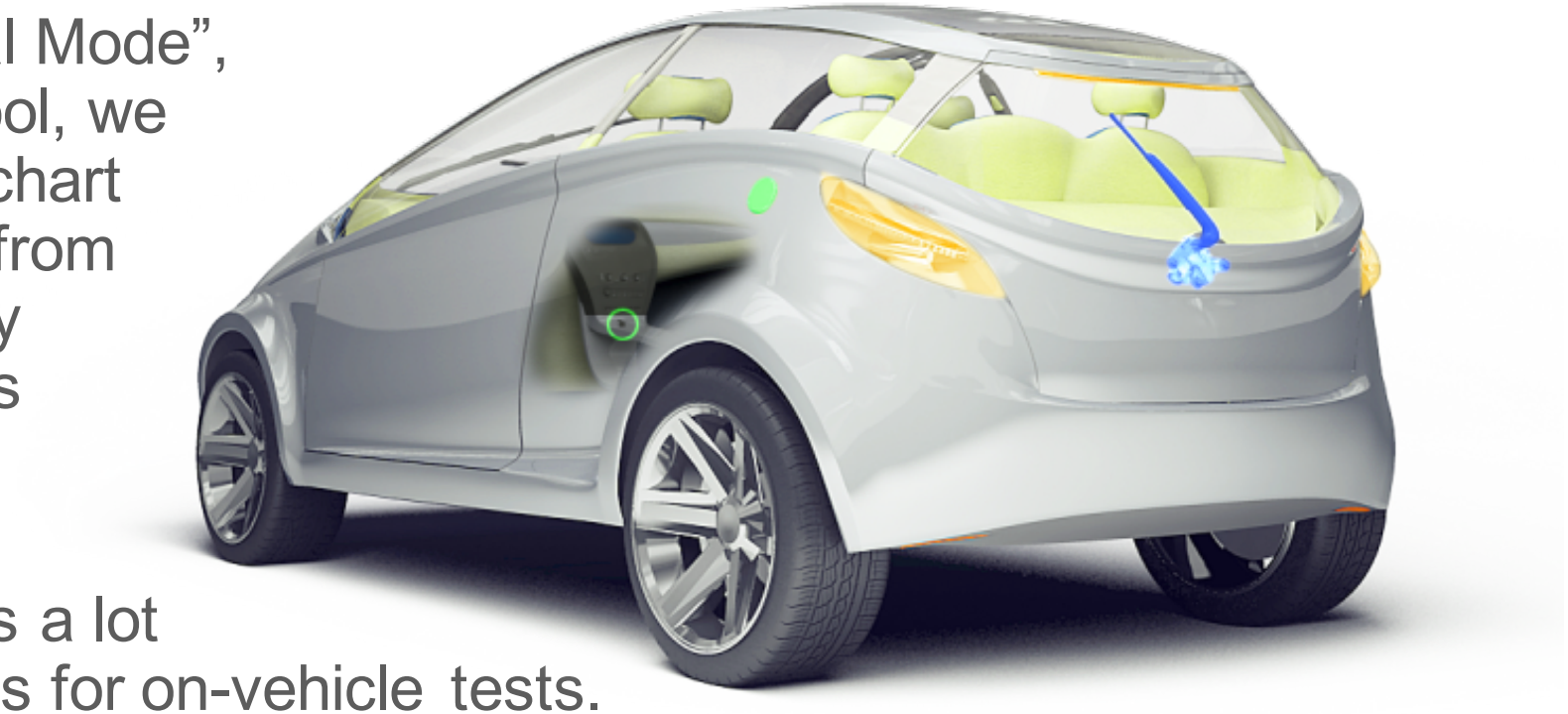


Then, launch Stateflow Debugging Activity



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.

Automotive technology, naturally