

MATLAB EXPO 2019

从Classic AUTOSAR到Adaptive AUTOSAR

- Simulink的AUTOSAR之旅

MathWorks 龚小平



内容

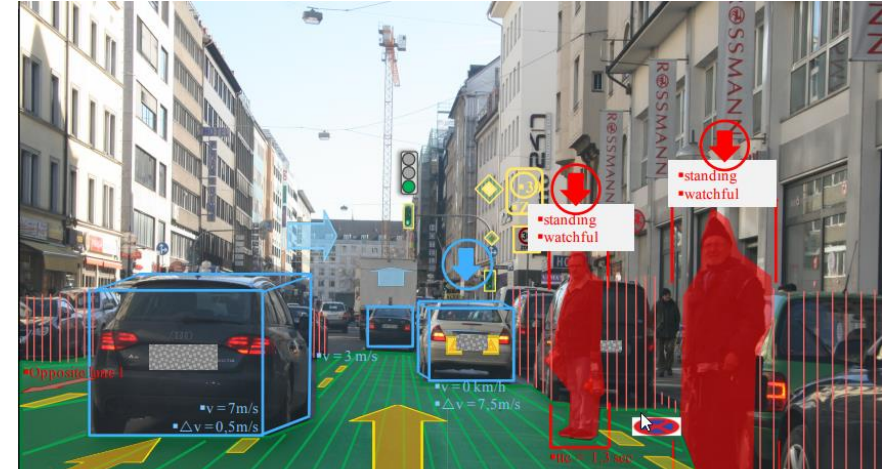
- AUTOSAR应用概述
- Simulink的AUTOSAR开发支持
- Simulink的Adaptive平台支持

内容

- AUTOSAR应用概述
- Simulink的AUTOSAR开发支持
- Simulink的Adaptive平台支持

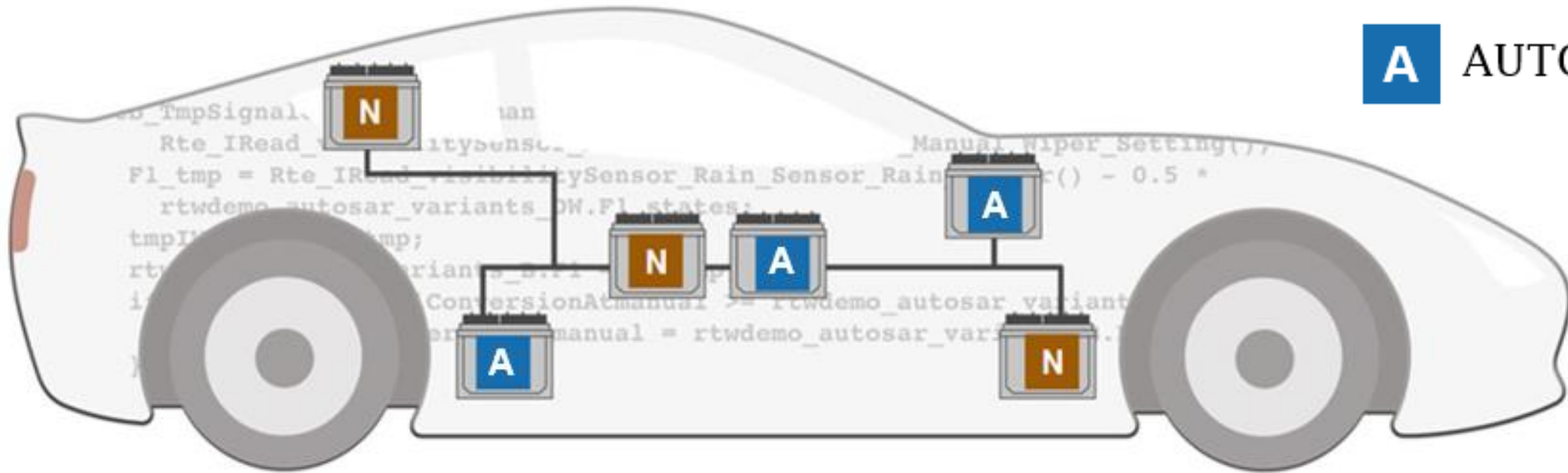
AUTOSAR Classic应用案例

- [BMW](#) - 从主机厂的角度看基于模型软件开发
- [FCA Global Powertrain Controls](#) - 利用基于模型设计、自动代码生成和AUTOSAR进行发动机控制量产应用的架构设计和实现
- [LG Chem](#) - 基于模型设计开发符合AUTOSAR和ISO26262的混动车电池管理系统软件
- [John Deere](#) - John Deere的垂直化AUTOSAR系统开发



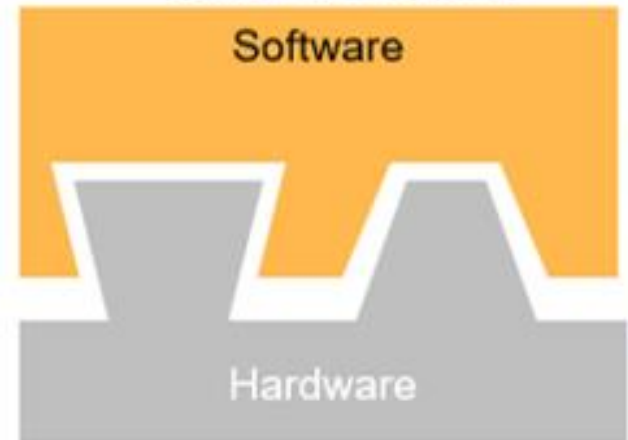
汽车软件架构平台现状

- N** Non - AUTOSAR
- A** AUTOSAR



Non-AUTOSAR

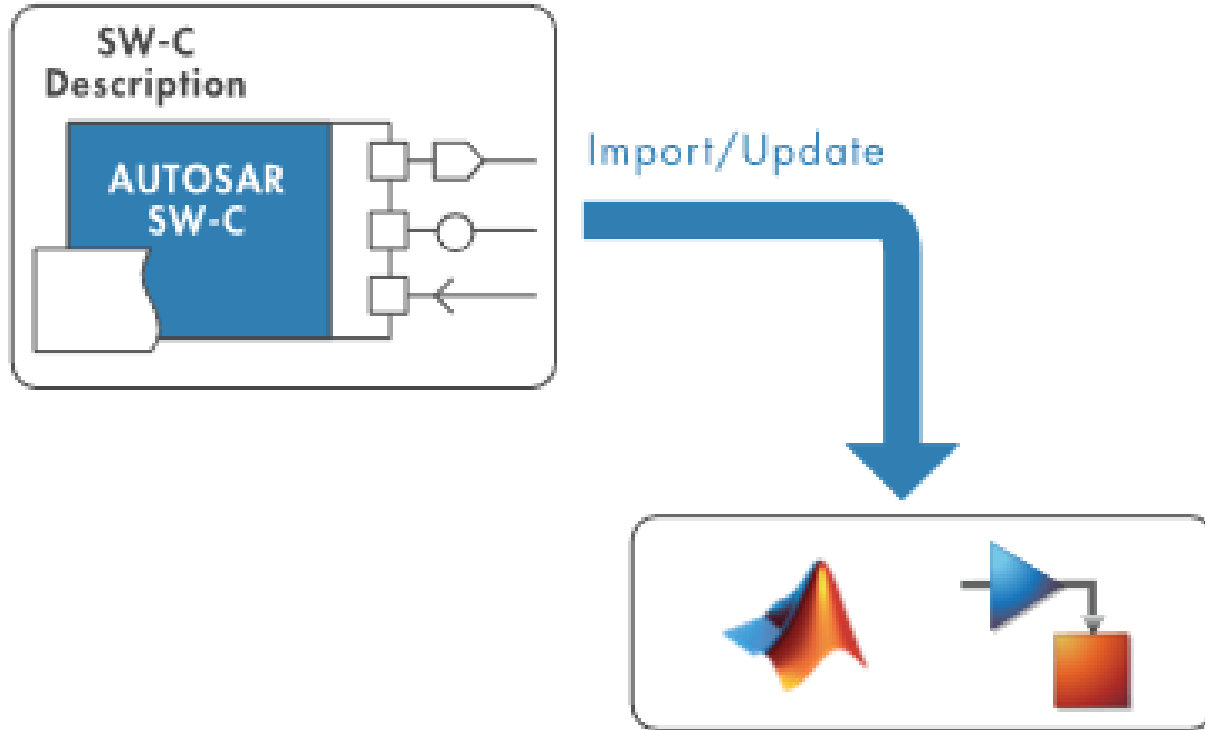
AUTOSAR



内容

- AUTOSAR应用概述
- Simulink的AUTOSAR开发支持
 - 架构导入导出
 - 属性词典
 - ECU软件仿真
 - 例程库模块
- Simulink的Adaptive平台支持

正向流程 – 从导入架构描述文件开始



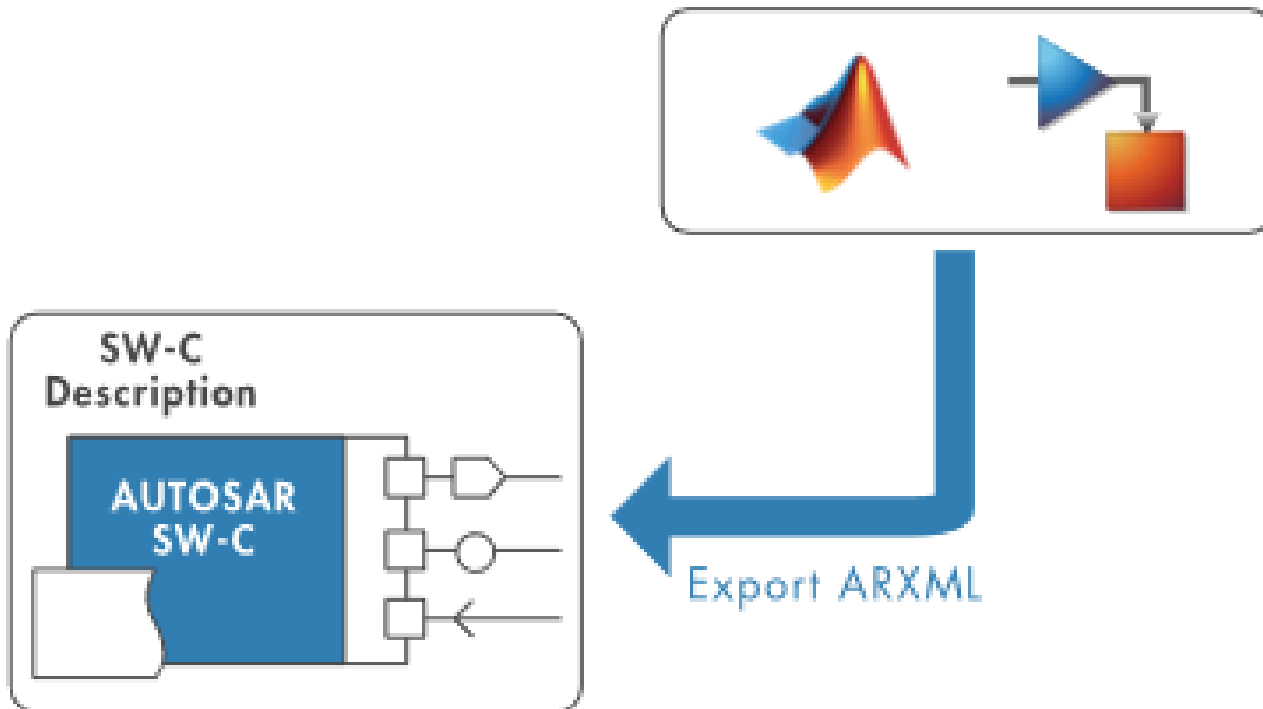
1. 导入软件组件描述文件(arxml)并自动创建Simulink模型架构

```
h = arxml.importer('mySWC.arxml')  
h.createComponentAsModel('/path/mySWC')
```

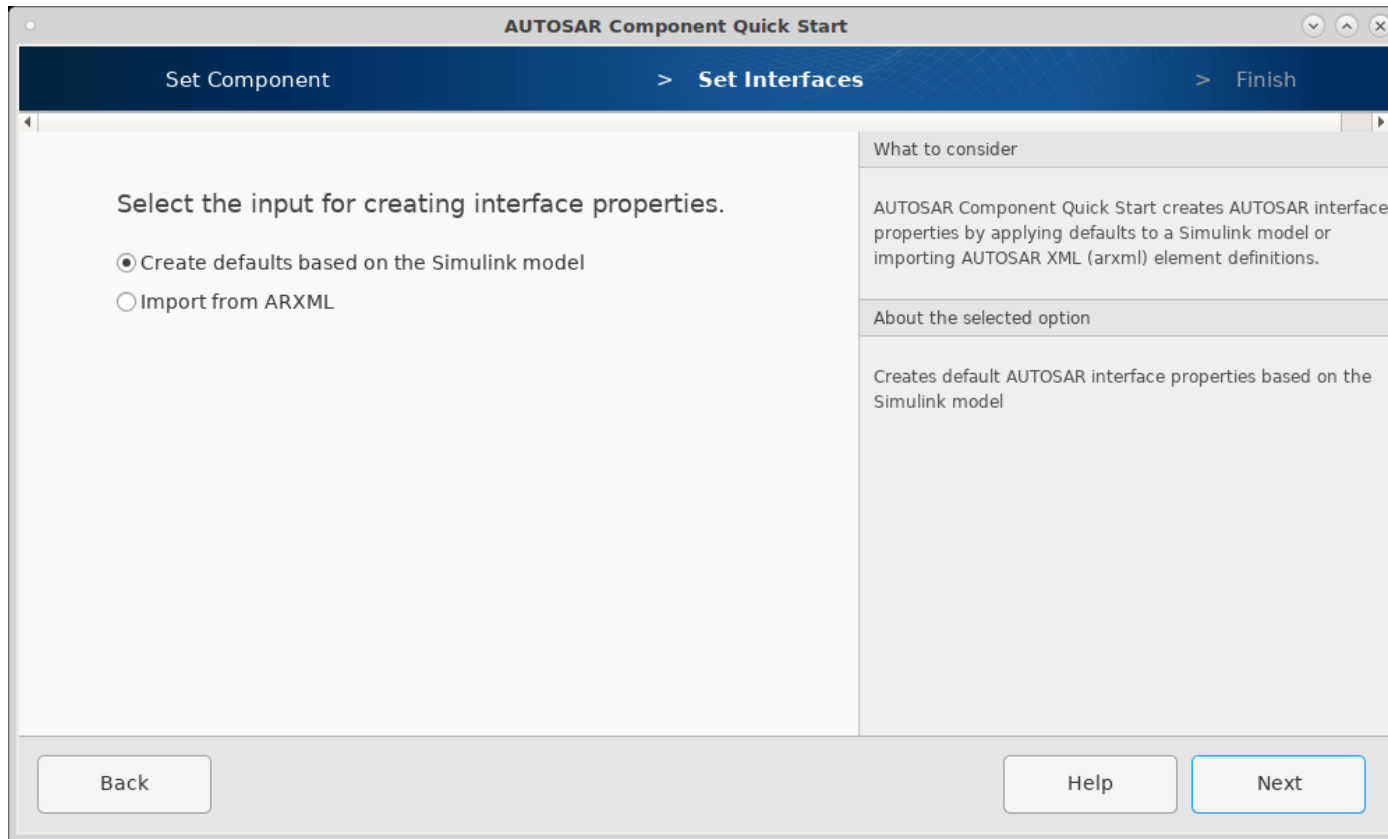
2. 软件组件细化设计、添加算法并生成代码

逆向流程 – 从已有Simulink模型开始 1/2

1. 从Simulink模型开始

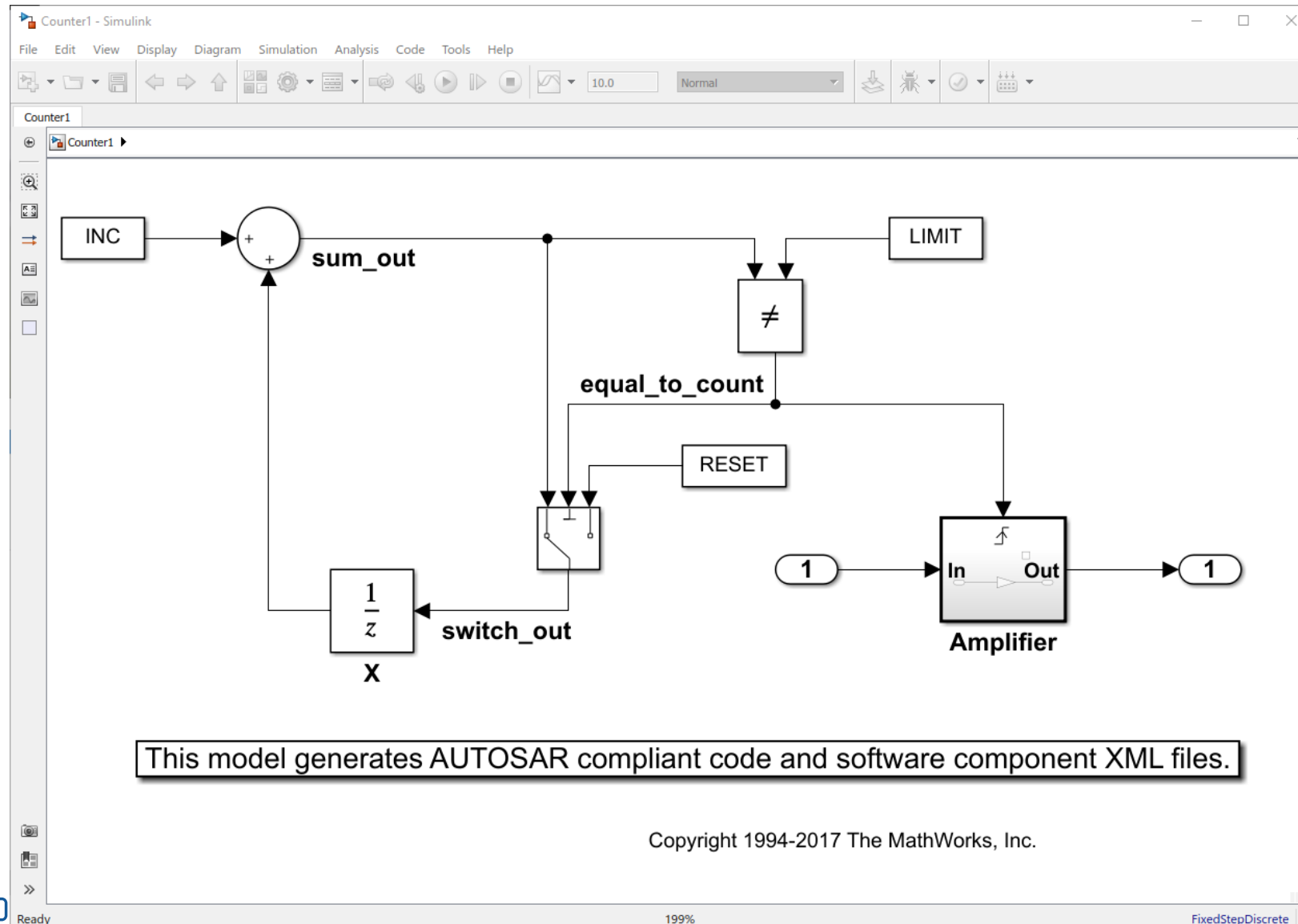


逆向流程 – 从已有Simulink模型开始 2/2



1. 从Simulink模型开始
2. 利用AUTOSAR组件快速配置功能
3. 软件组件细化设计、添加算法并生成代码

案例展示 – 从Simulink模型生成AUTOSAR代码 1/8



案例展示 – 启动AUTOSAR快速配置 2/8

Counter1 - Simulink

File Edit View Display Diagram Simulation Analysis Code Tools Help

Counter1

AUTOSAR Component Quick Start

Set Component > Set Interfaces > Finish

Configure AUTOSAR software component properties

Component details:

Map model to AUTOSAR software component

Component name: Counter1

Component package: /Company/Powertrain/Components

Component type: Application

What to consider

AUTOSAR Component Quick Start maps a Simulink model to an AUTOSAR software component. For the component, specify an AUTOSAR short name, package path, and component type, or accept default values. Package paths can use an organizational naming pattern, such as /Company/Powertrain/Components. Component type determines the APIs available to the component in the run-time environment.

About the selected option

Creates application software component

Out

1

This model generates AUTOSAR compliant code and software component XML files.

Copyright 1994-2017 The MathWorks, Inc.

Ready 199% FixedStepDiscrete

案例展示 – 设置组件类型 3/8

The screenshot shows the 'AUTOSAR Component Quick Start' wizard window. The title bar includes the window name and standard minimize, maximize, and close buttons. The main area is divided into two panes. The left pane, titled 'Set Component', contains the following sections:

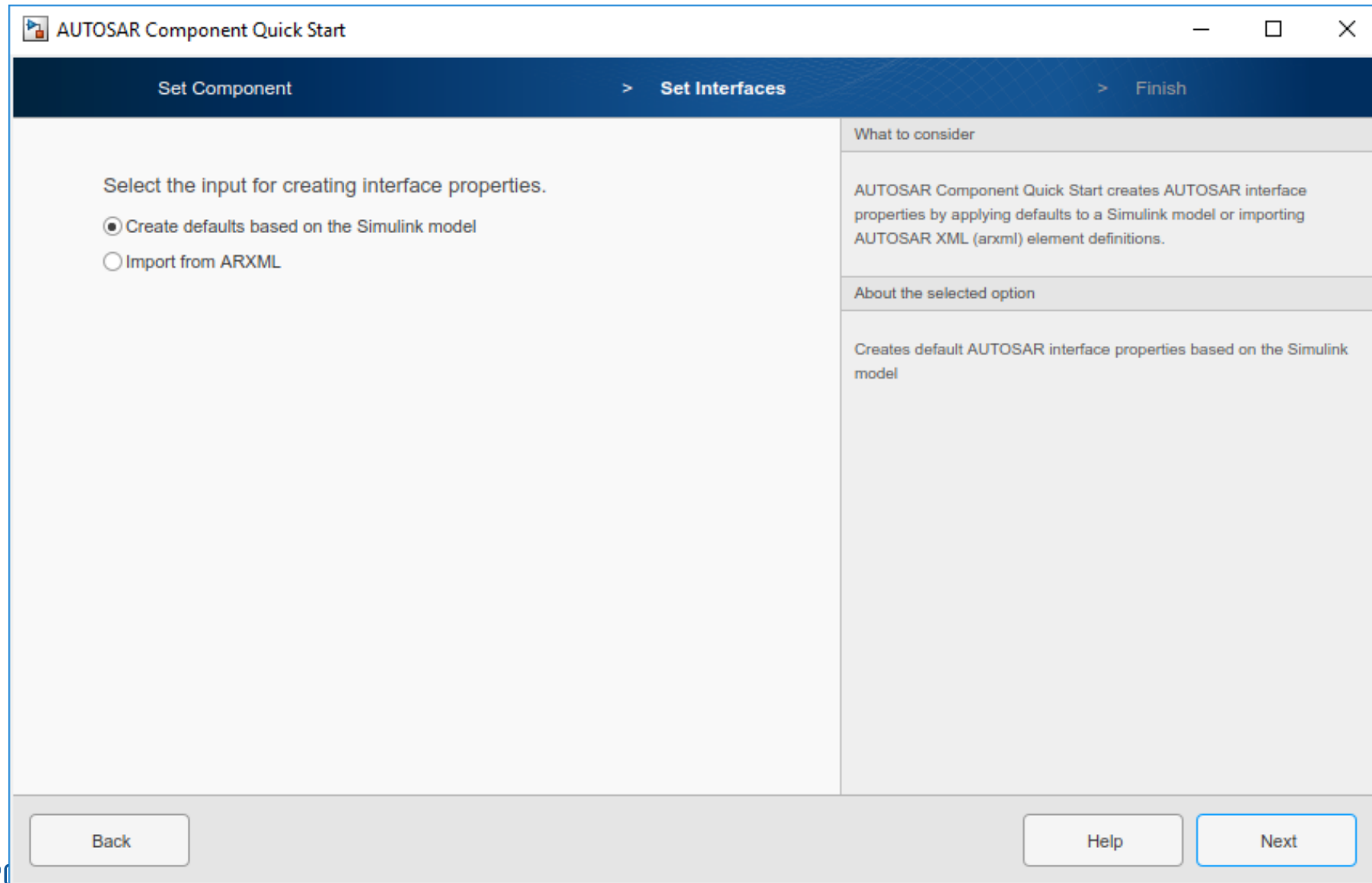
- Configure AUTOSAR software component properties**
Component details:
- Map model to AUTOSAR software component**
 - Component name:
 - Component package:
 - Component type:
 - Application (selected)
 - ComplexDeviceDriver
 - EcuAbstraction
 - SensorActuator
 - ServiceProxy

The right pane contains two sections:

- What to consider**
AUTOSAR Component Quick Start maps a Simulink model to an AUTOSAR software component. For the component, specify an AUTOSAR short name, package path, and component type, or accept default values. Package paths can use an organizational naming pattern, such as /Company/Powertrain/Components. Component type determines the APIs available to the component in the run-time environment.
- About the selected option**
Creates application software component

At the bottom right, there are two buttons: 'Help' and 'Next'.

案例展示 – 接口设置方式 4/8



案例展示 – 进入代码映射视图 5/8

This model generates AUTOSAR compliant code and software component XML files.

Copyright 1994-2017 The MathWorks, Inc.

Code Mappings - AUTOSAR SW Component

Source	DataAccessMode	Port	Element
Input	ImplicitReceive	Input	Input

案例展示 – 代码映射和检查 6/8



同步模型和代码映射



验证映射和AUTOSAR属性

Code Mappings - AUTOSAR SW Component

Inports

Outports

Entry-Point Functions

Data Transfers

Function Callers

Parameters

Signals/States

Data Stores



Source	DataAccessMode	Port
Input	ImplicitReceive	Input



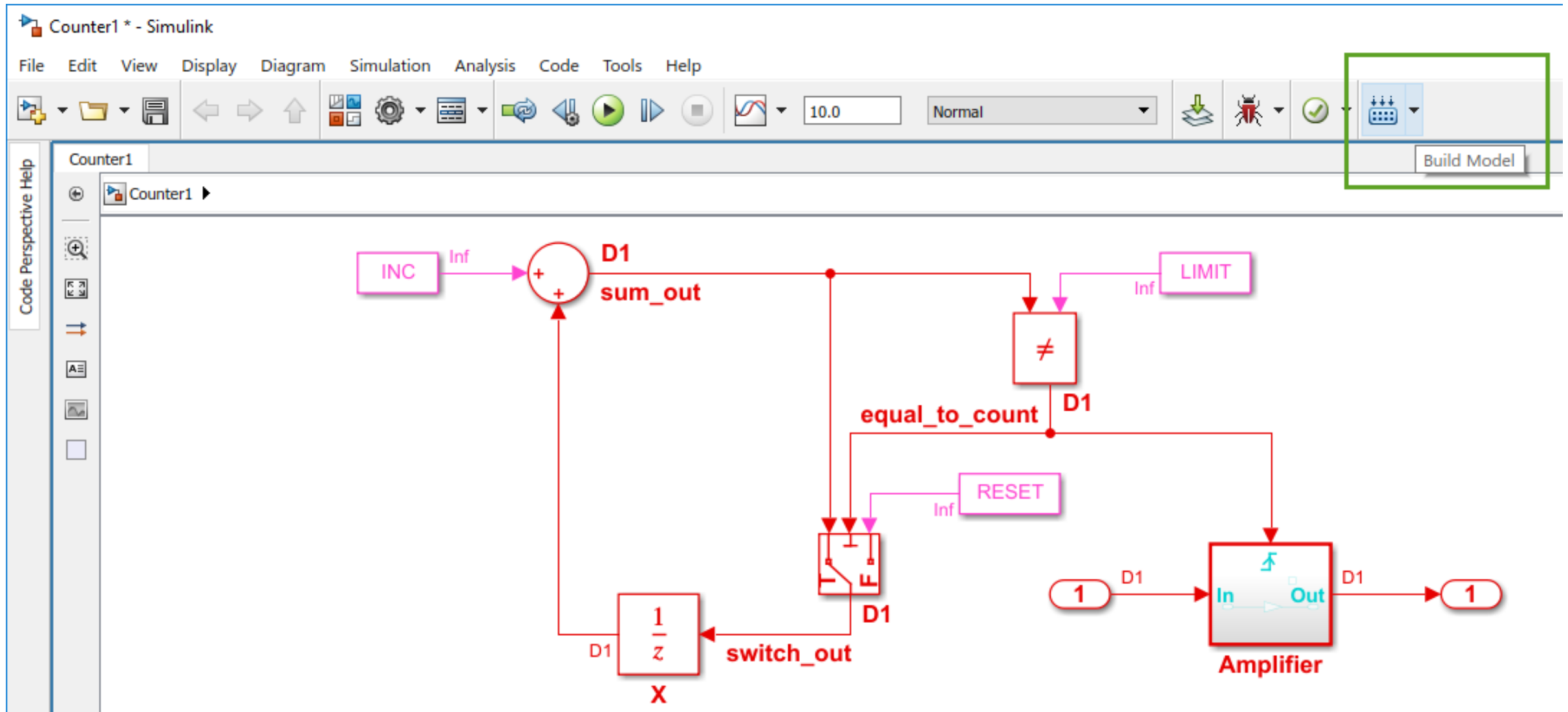
启动AUTOSAR属性词典

案例展示 – AUTOSAR属性词典检查 7/8

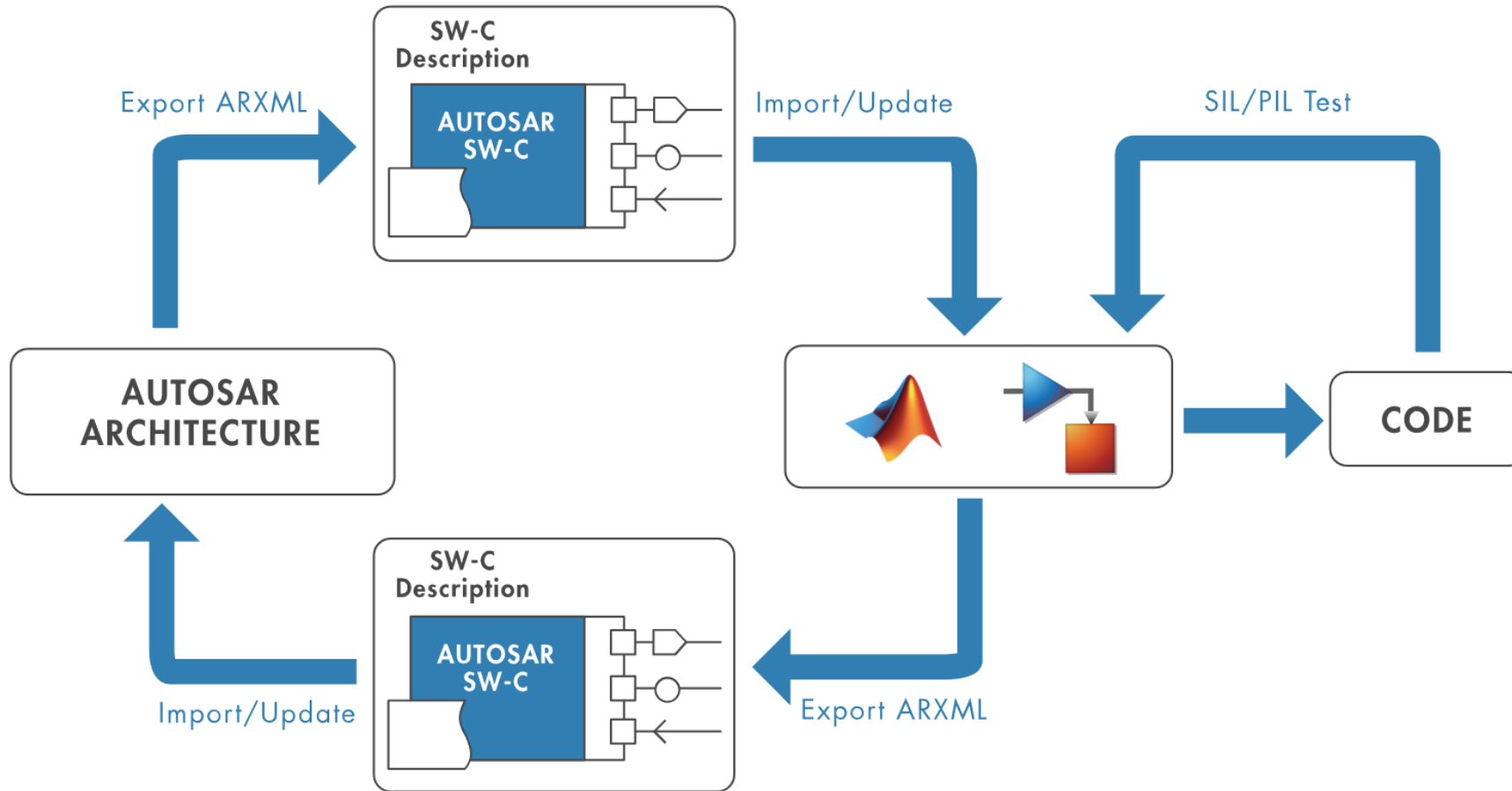
The screenshot displays the 'AUTOSAR Dictionary: Counter1' interface. The left pane shows a tree view of the dictionary structure. The right pane shows a table of data elements.

Name	SwCalibrationAccess	DisplayFormat	SwAddrMethod
Output	ReadOnly		<None>

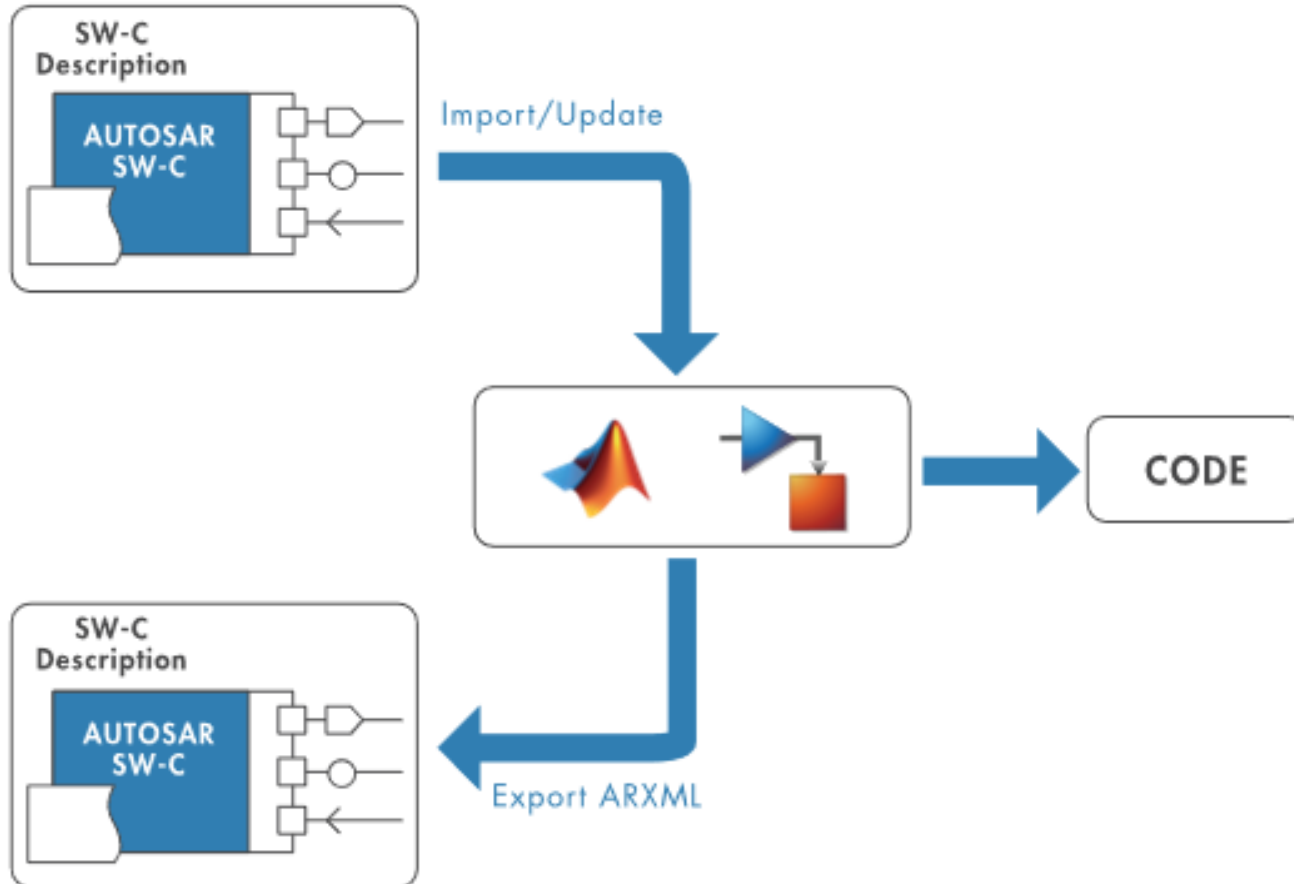
案例展示 – 启动代码生成 8/8



AUTOSAR完整工作流程一览

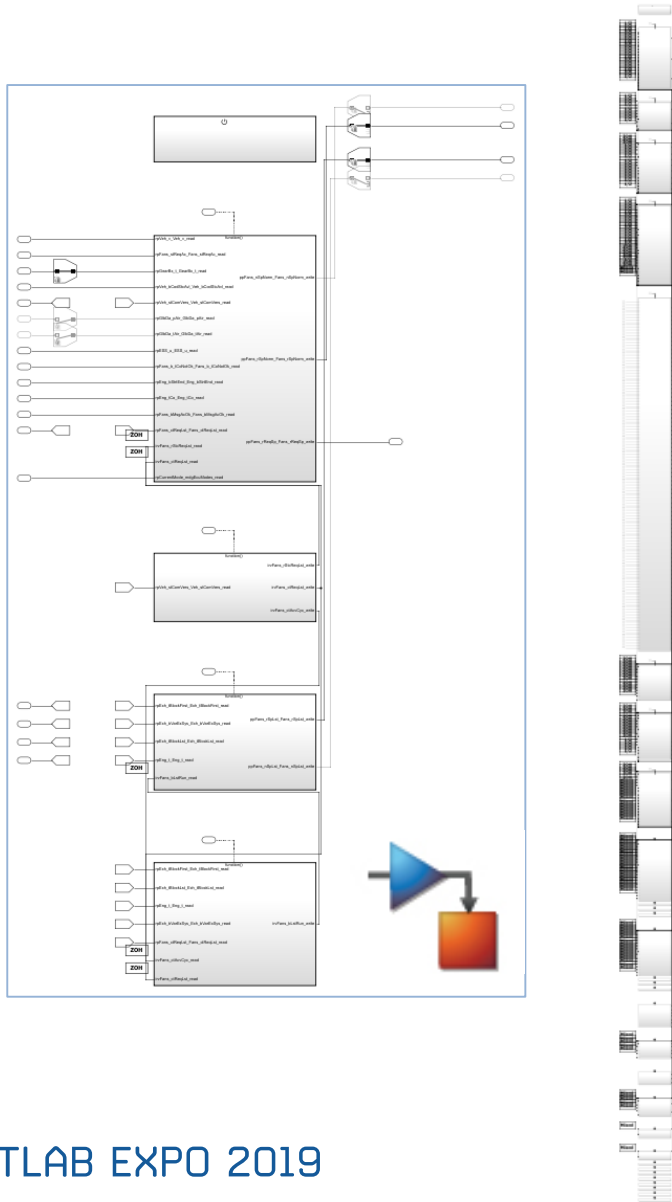


软件组件设计与实现



1. 从Simulink模型开始或导入架构描述文件
2. 软件组件细化设计、添加算法并生成代码

在Simulink中设计AUTOSAR软件组件



- 1) 模型中的哪些模块需要进行AUTOSAR配置?
- 2) 如何修改模型中的AUTOSAR属性?
- 3) 如何获取更多的信息和帮助?

AUTOSAR配置界面

快速帮助

帮助进行
AUTOSAR
模型配置

The screenshot displays the AUTOSAR configuration environment. On the left, a sidebar provides quick help and configuration options. The main workspace shows a block diagram with an 'Initialize' block and a 'Runnable_1s' block containing a 'SubVal' and an 'Override' block. A 'Code Mappings - AUTOSAR' window is open, showing the following table:

Inports	Outputs	Entry-Point Functions	Data Transfers	Function
Source	.DataAccessMode	Port	Element	
SubVal	ImplicitReceive	RPort	SubVal	
Override	ImplicitReceive	RPort	Override	

On the right, the 'Property Inspector' window shows the following table for the 'SubVal' component:

NAME	VALUE
Source	SubVal
Code	
DataAccessMode	ImplicitReceive
Port	RPort
Element	SubVal
Communication attributes	
AliveTimeout	60
HandleNeverRec...	false
InitValue	0

属性查看

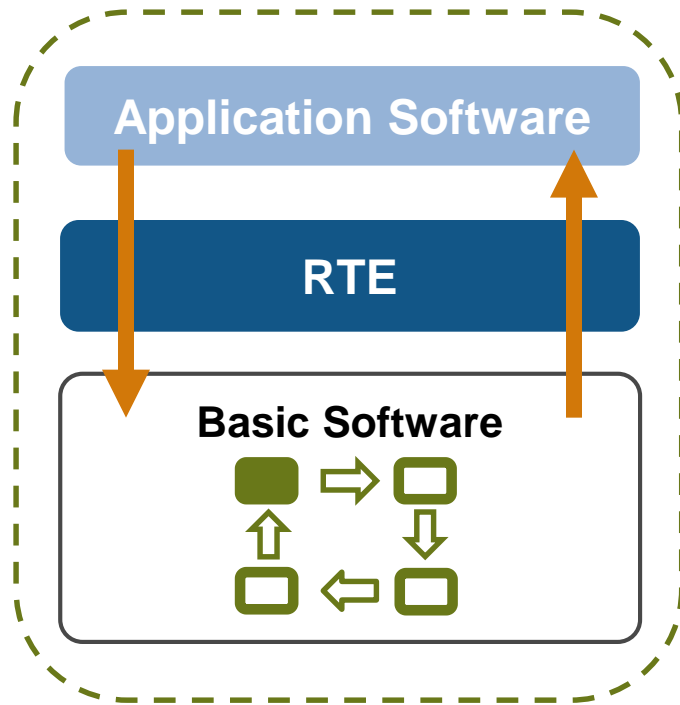
查看/编辑
AUTOSAR
软件组件属性

代码映射

查看/编辑已配置为AUTOSAR的所有的模块和元素

基础软件的功能仿真 1/2

AUTOSAR ECU 分层架构



应用软件和基础软件之间有许多调用



基础软件的功能高度动态化和复杂化



对基础软件进行仿真可以减低开发时间、提升软件质量

基础软件的功能仿真 2/2



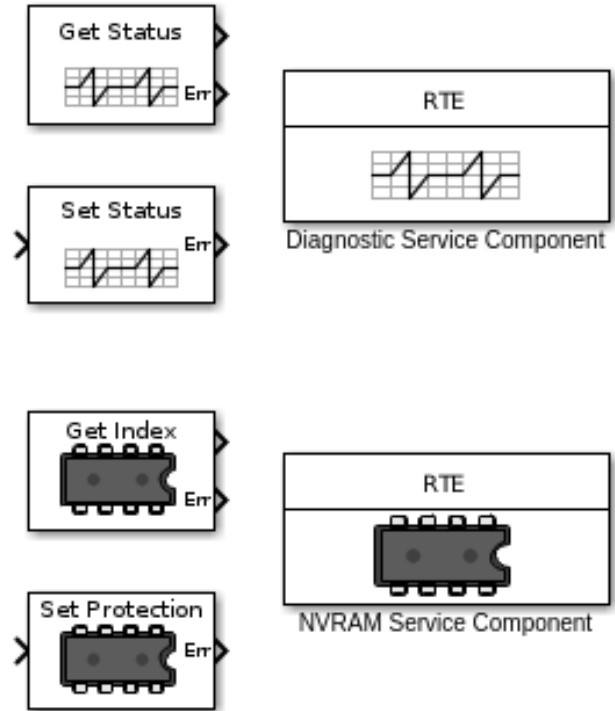
Specification of Diagnostic Event Manager
AUTOSAR Release 4.2.2

Document Title	Specification of Diagnostic Event Manager
Document Owner	AUTOSAR
Document Responsibility	AUTOSAR
Document Identification No	019
Document Classification	Standard

Document Status	Final
Part of AUTOSAR Release	4.2.2

Document Change History		
Release	Changed by	Description
4.2.2	AUTOSAR Release Management	<ul style="list-style-type: none"> New APIs <code>Dem_GetEventFreezeFrameDataEx</code> and <code>Dem_GetEventExtendedDataRecordEx</code> with buffersize as parameter and corrected return value definitions. Providing OBD FreezeFrame for UDS service 0x19 0x05 ISO 14229-1:2013[1] NRC handling for service 0x14 Refined service interfaces for DataElements minor corrections / clarifications / editorial changes; For details please refer to the ChangeDocumentation
4.2.1	AUTOSAR Release Management	<ul style="list-style-type: none"> Support of ISO 27145 (WWH-OBd / Euro VI)[2] Update to support ISO 14229-1:2013[3] Introduction of event dependencies Refined DTC/Event suppression
4.1.3	AUTOSAR Release Management	<ul style="list-style-type: none"> Further clarification of event combination Clarification of DTC groups Editorial changes
4.1.2	AUTOSAR Release Management	<ul style="list-style-type: none"> Added API table for service interfaces Clarification of event combination Editorial changes Removed chapter(s) on change documentation

封装



基础软件库

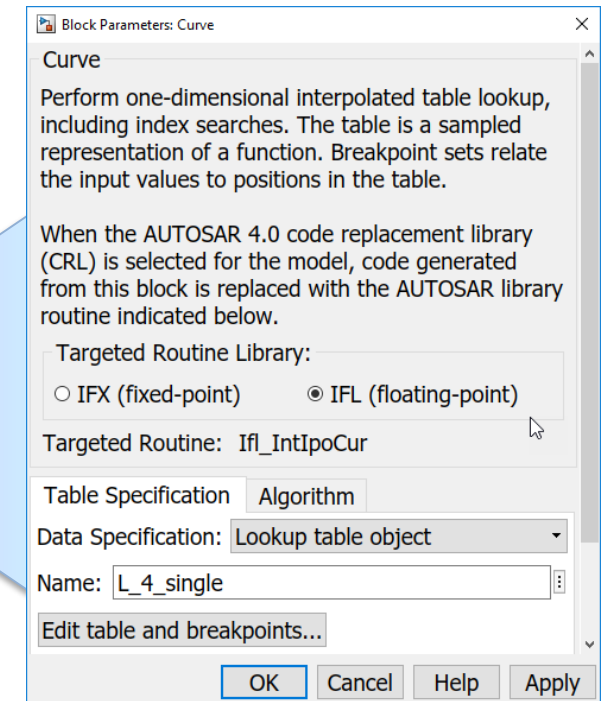
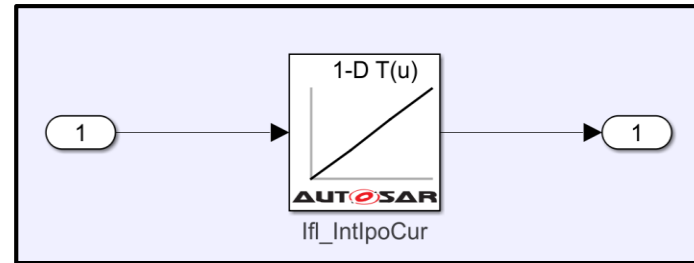
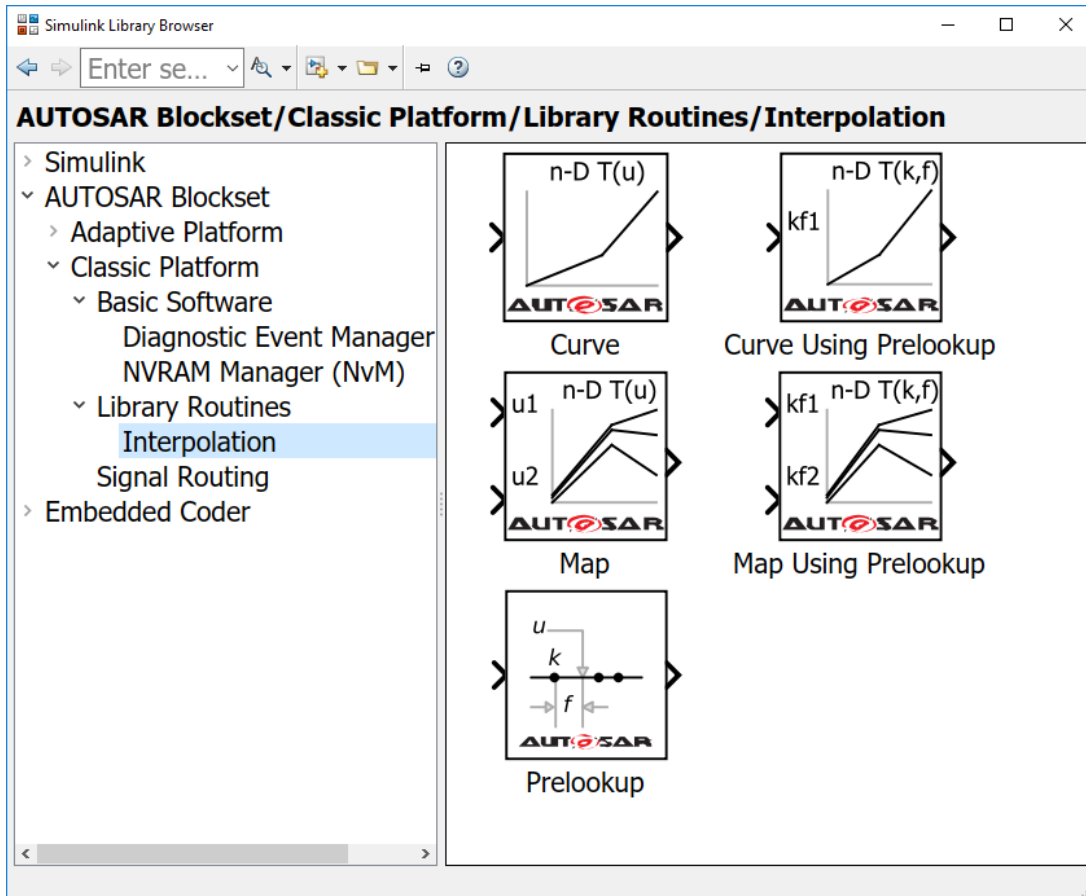
R2017b

1 of 475

Document ID 019: AUTOSAR_SWS_DiagnosticEventManager
— AUTOSAR CONFIDENTIAL —

详细规范

AUTOSAR例程库支持



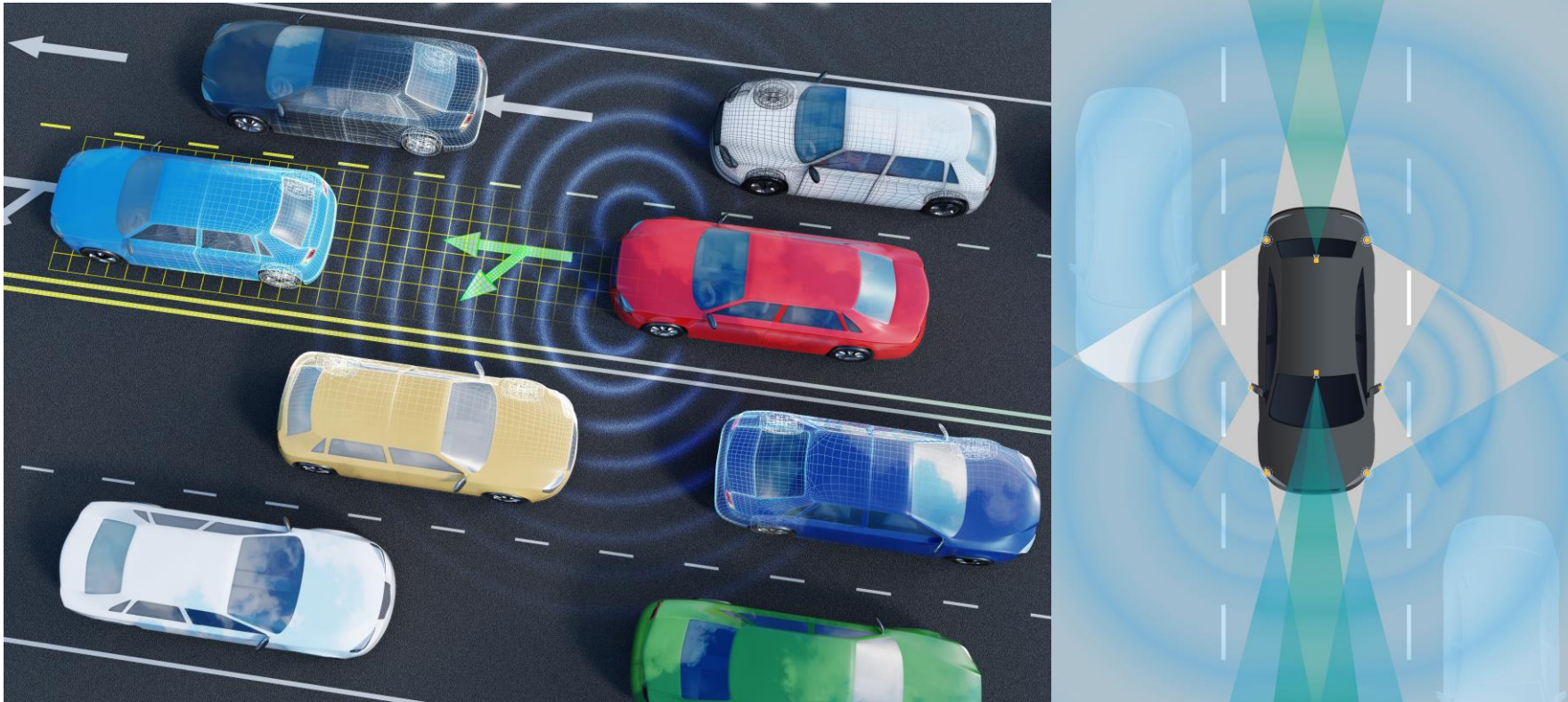
```
Rte_IWrite_Runnable_Step_Out1_Out1 (Ifl_IntIpoCur_f32_f32
(Rte_IRead_Runnable_Step_In1_In1(), Rte_CData_L_4_single()->Nx,
Rte_CData_L_4_single()->Bp1, Rte_CData_L_4_single()->Table));
```


内容

- AUTOSAR应用概述
- Simulink的AUTOSAR开发支持
- Simulink的Adaptive平台支持
 - 新平台的驱动力
 - Adaptive平台概述
 - 与Simulink的映射
 - 组件代码生成

AUTOSAR Adaptive的驱动力

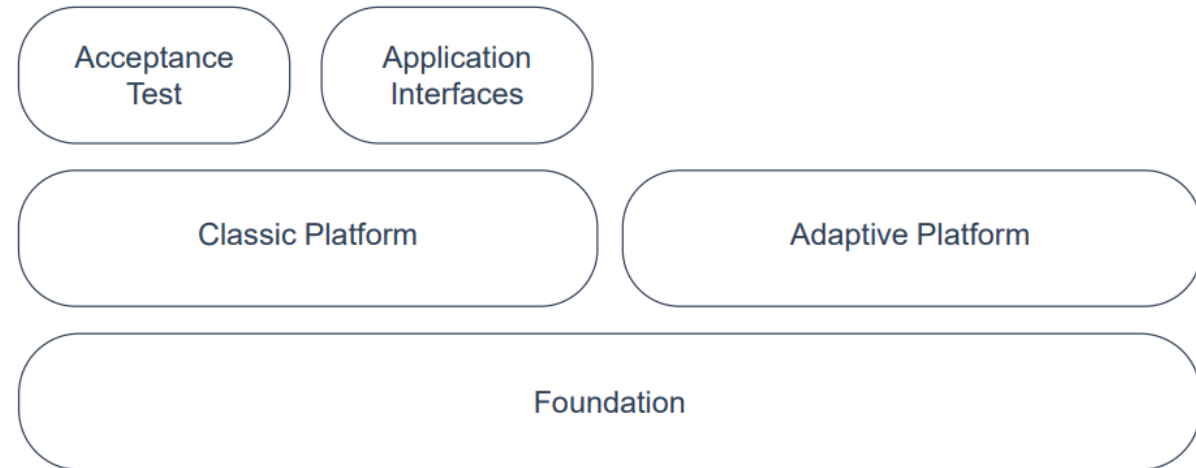
- 主要驱动 – 自动驾驶和车联网的应用



AUTOSAR迎接自动驾驶时代

- 2016年启动新平台标准制定工作
- 2017年3月发布第1版 Adaptive平台标准

The platforms are organized by 5 AUTOSAR standards



AUTOSAR

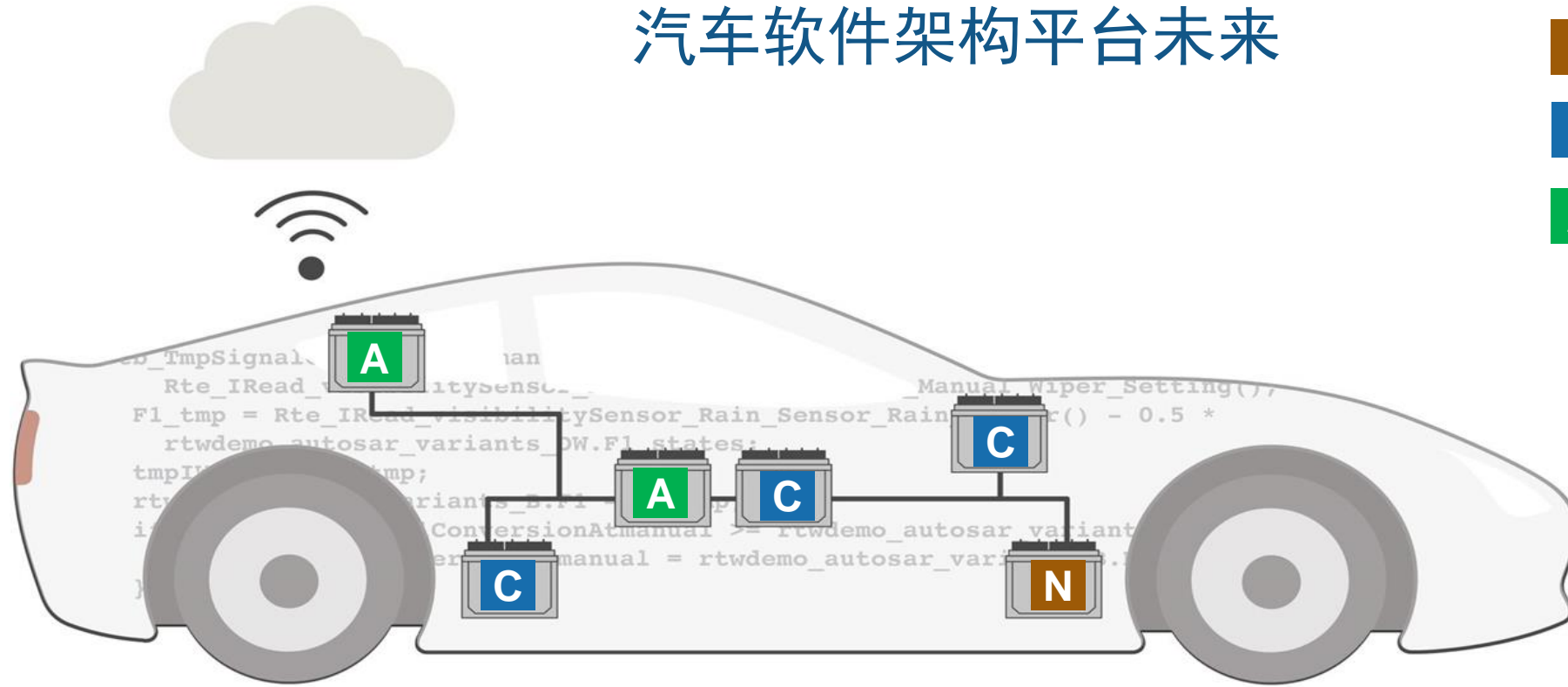
Introduction | July 18 | 34

From AUTOSAR.org – AUTOSAR Introduction

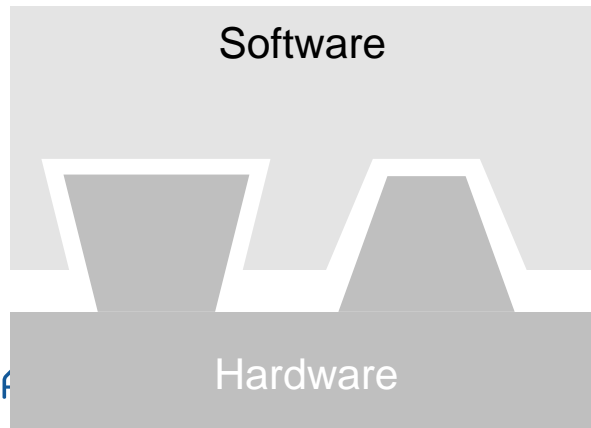
OVER THE AIR UPDATE

汽车软件架构平台未来

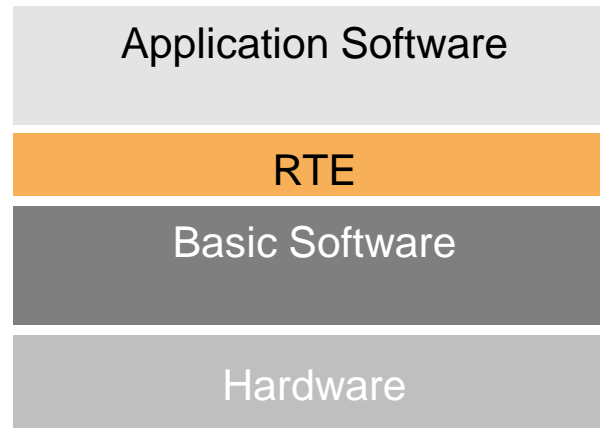
- N** Non - AUTOSAR
- C** Classic - AUTOSAR
- A** Adaptive - AUTOSAR



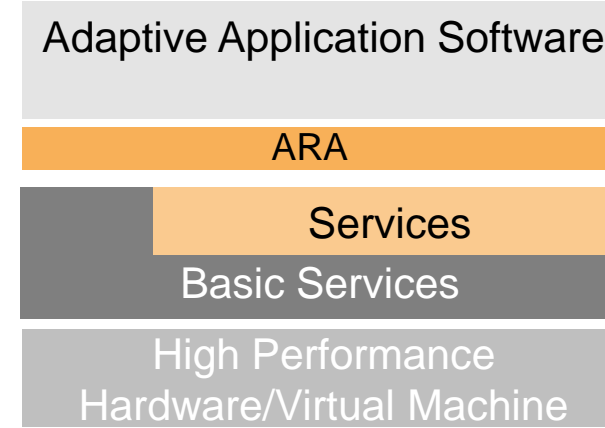
Non- AUTOSAR



Classic AUTOSAR



Adaptive AUTOSAR



Simulink同时支持两个AUTOSAR平台

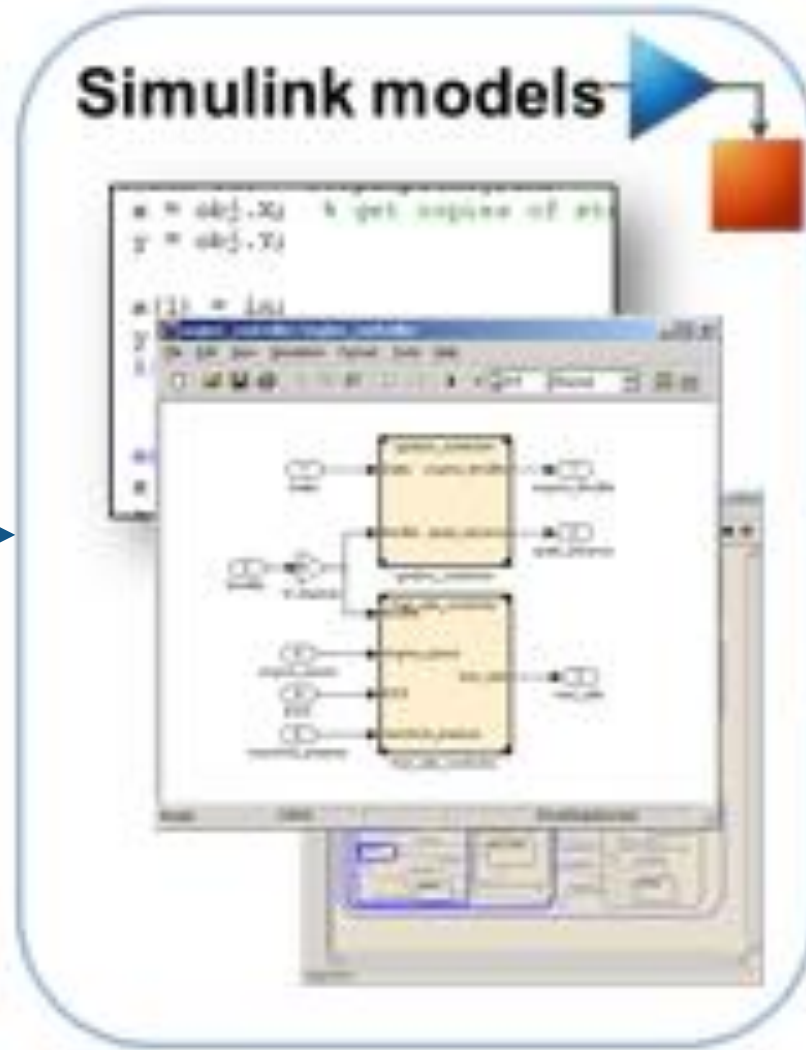
Classic AUTOSAR

Application Software

RTE

Basic Software

Hardware



Adaptive AUTOSAR

Adaptive Application Software

ARA

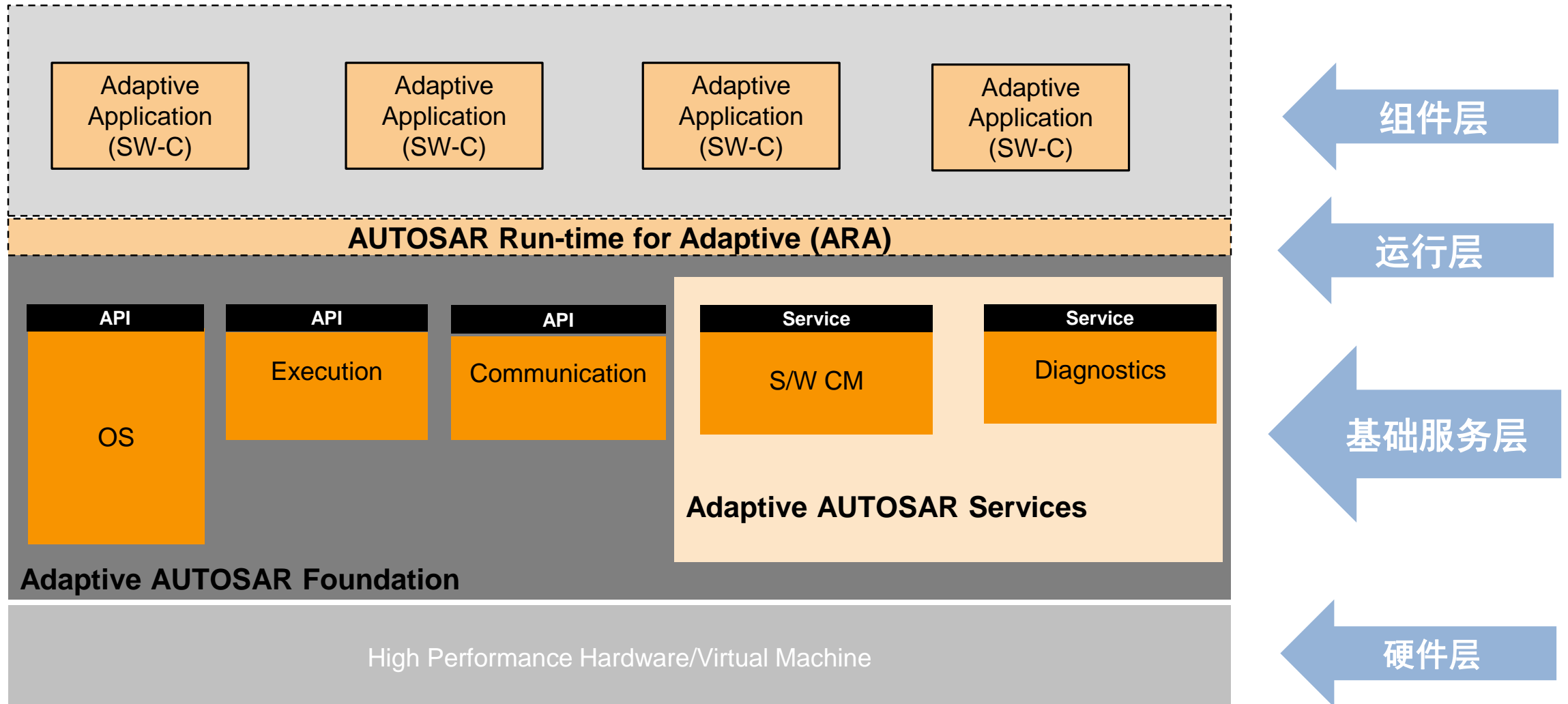
Services

Basic Services

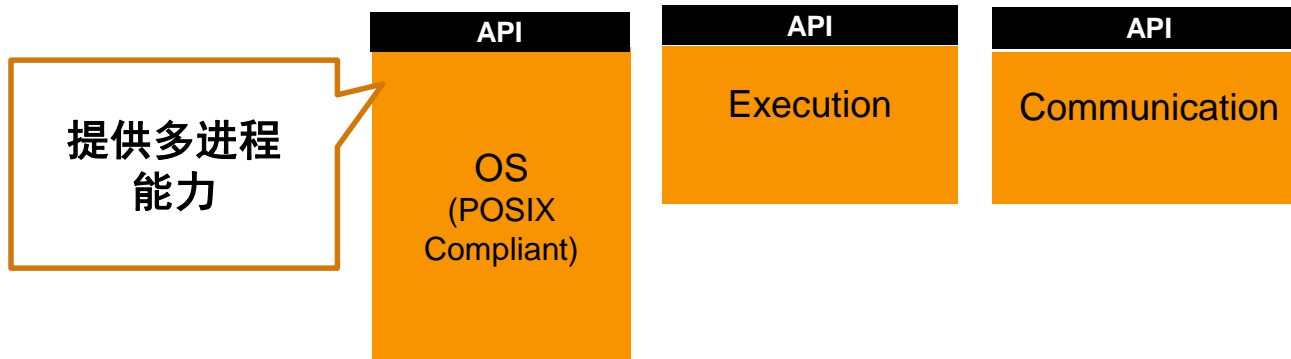
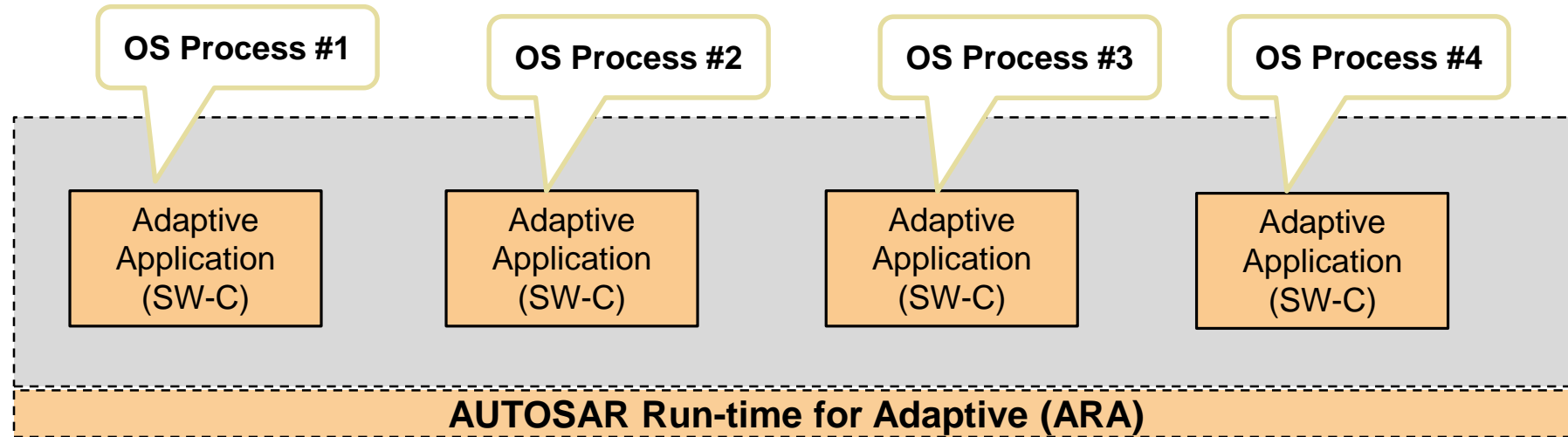
High Performance
Hardware/Virtual Machine

应用层仿真+算法开发

Adaptive AUTOSAR 软件分层架构

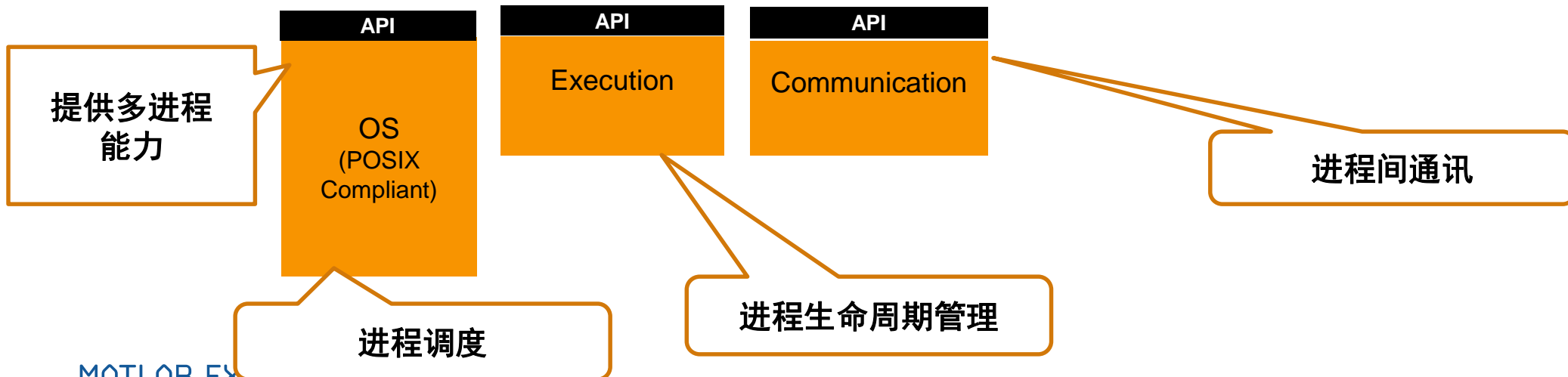
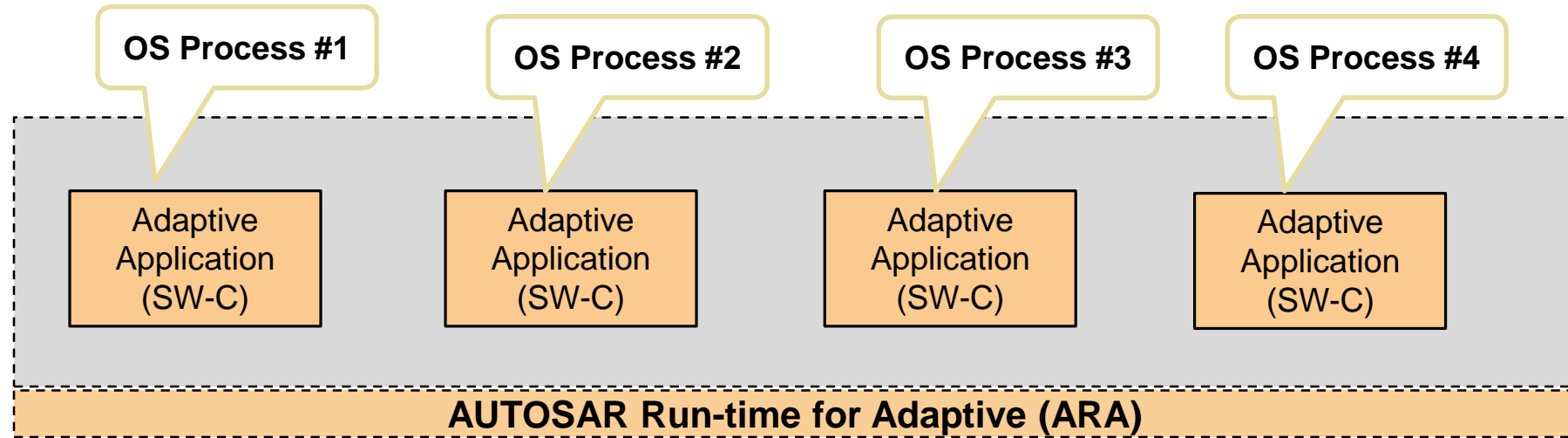


关键概念 #1 – 一切都是OS中的进程

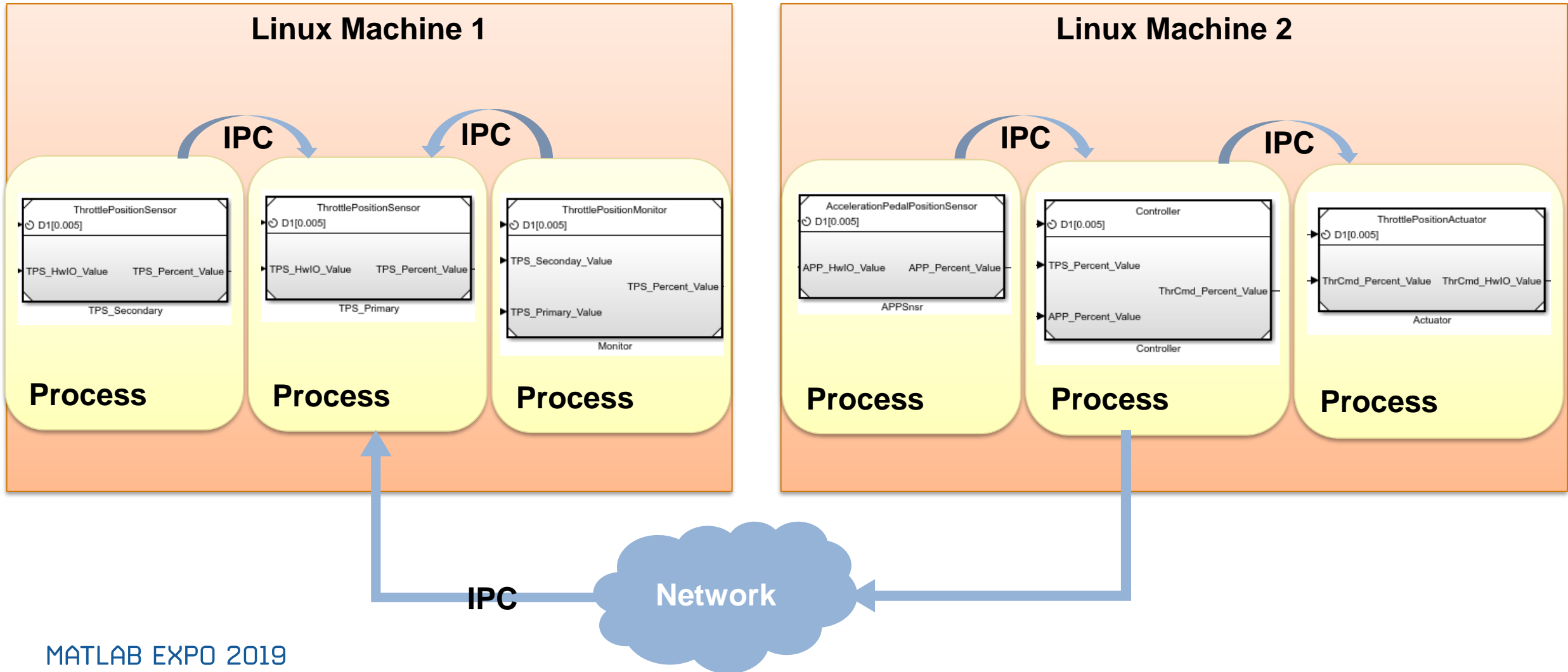


- 注: 每个OS进程
- 对应代码中的main()函数
 - 拥有独立的内存空间和命名空间
 - 可以是单线程或多线程

关键概念 #1 – 一切都是OS中的进程



关键概念 #2 – 面向服务的进程间通讯



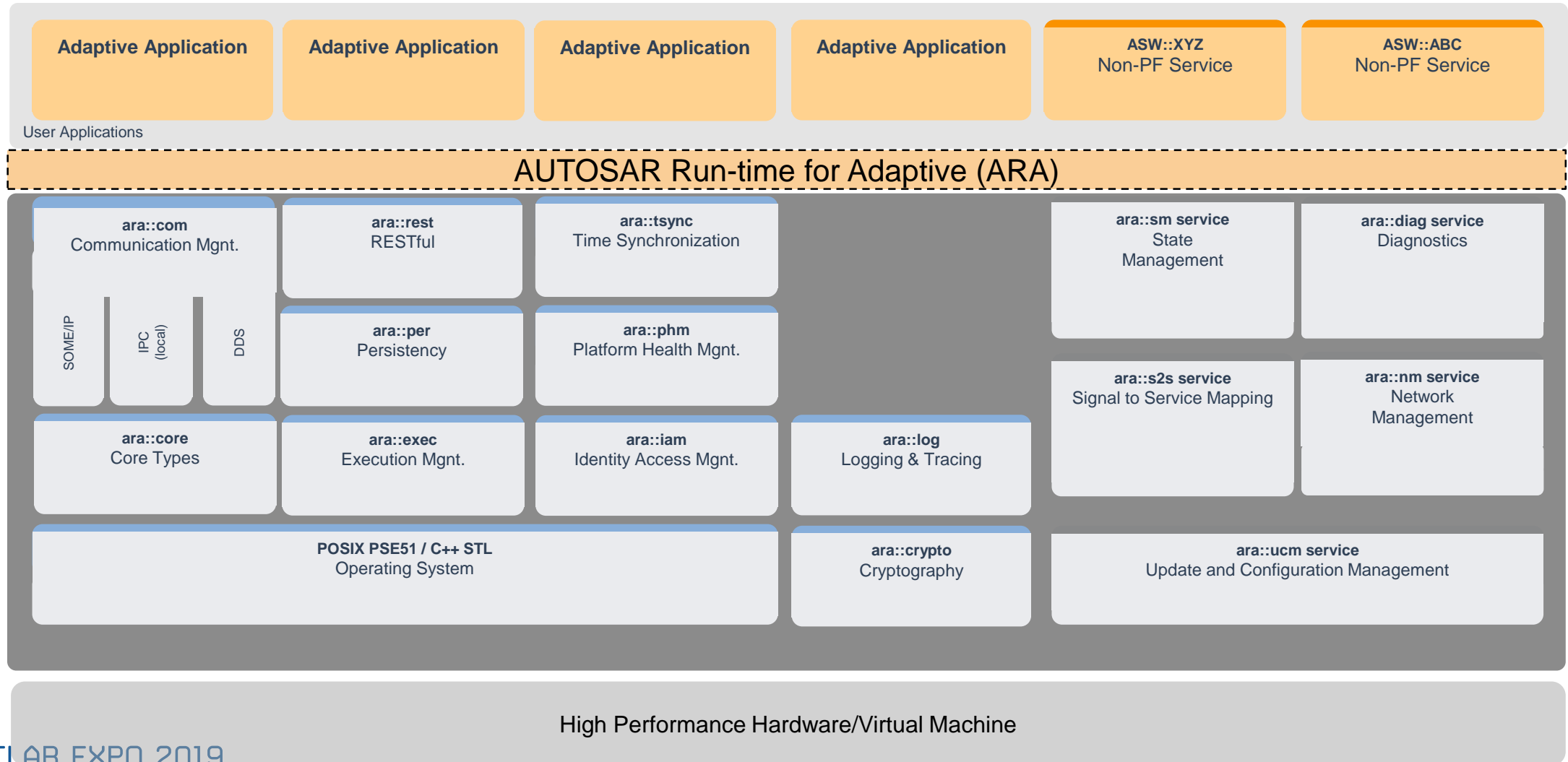
关键概念 #2 – 面向服务的进程间通讯

- 服务接口可以包含：

- 方法 (函数)
- 时间 (消息)
- 字段 (数据)

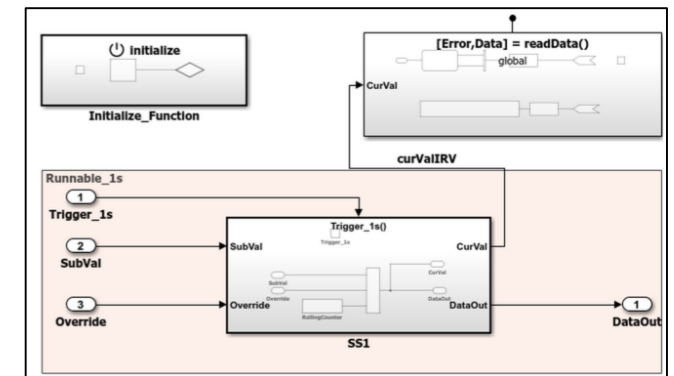
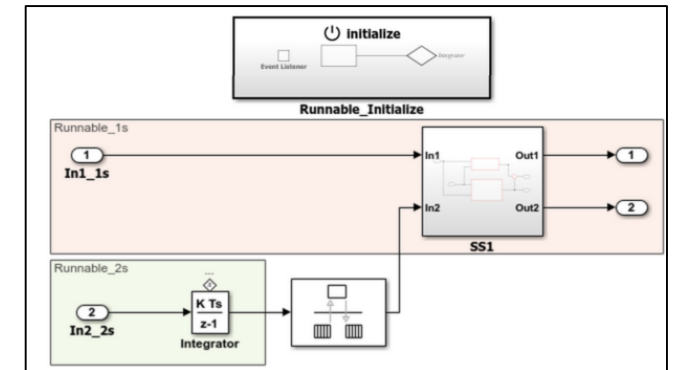
```
    <<interface example>>  
    RadarService  
  
    • result = Calibrate(config)  
    • [success, out_pos] = Adjust(in_pos)  
  
    • BrakeEvent  
  
    • UpdateRate
```

关键概念 #3 – 以C++为实施方式



Simulink为什么要支持Adaptive平台

- 在AUTOSAR Classic中Simulink被广泛使用
- 客户不断要求Simulink提供对Adaptive平台的支持
- Simulink本身支持面向服务的建模
- Embedded Coder支持C和C++代码生成
- MathWorks深度参与Classic和Adaptive AUTOSAR标准开发



```

void autosar_Lane_Guidance_IfActionSS(real_T rtu_In1, real_T *rty_Out1)
{
    // Inport: '<S18>/In1'
    *rty_Out1 = rtu_In1;
}

// Function for Chart: '<S1>/Event_Receive'
boolean_T autosar_Lane_GuidanceModelClass::
autosar_Lane_Guidance_sf_msg_pop_EvtIn(void)
{
    boolean_T isPresent;
    const ara::com::SampleContainer< ara::com::SamplePtr< const real_T > >
    *sampleContainer;
    ara::com::SamplePtr< const real_T > samples;
    if (autosar_Lane_Guidance_DW.EvtIn_isValid_i) {
        isPresent = true;
    }
}
    
```

从Adaptive概念到Simulink的映射 1/2



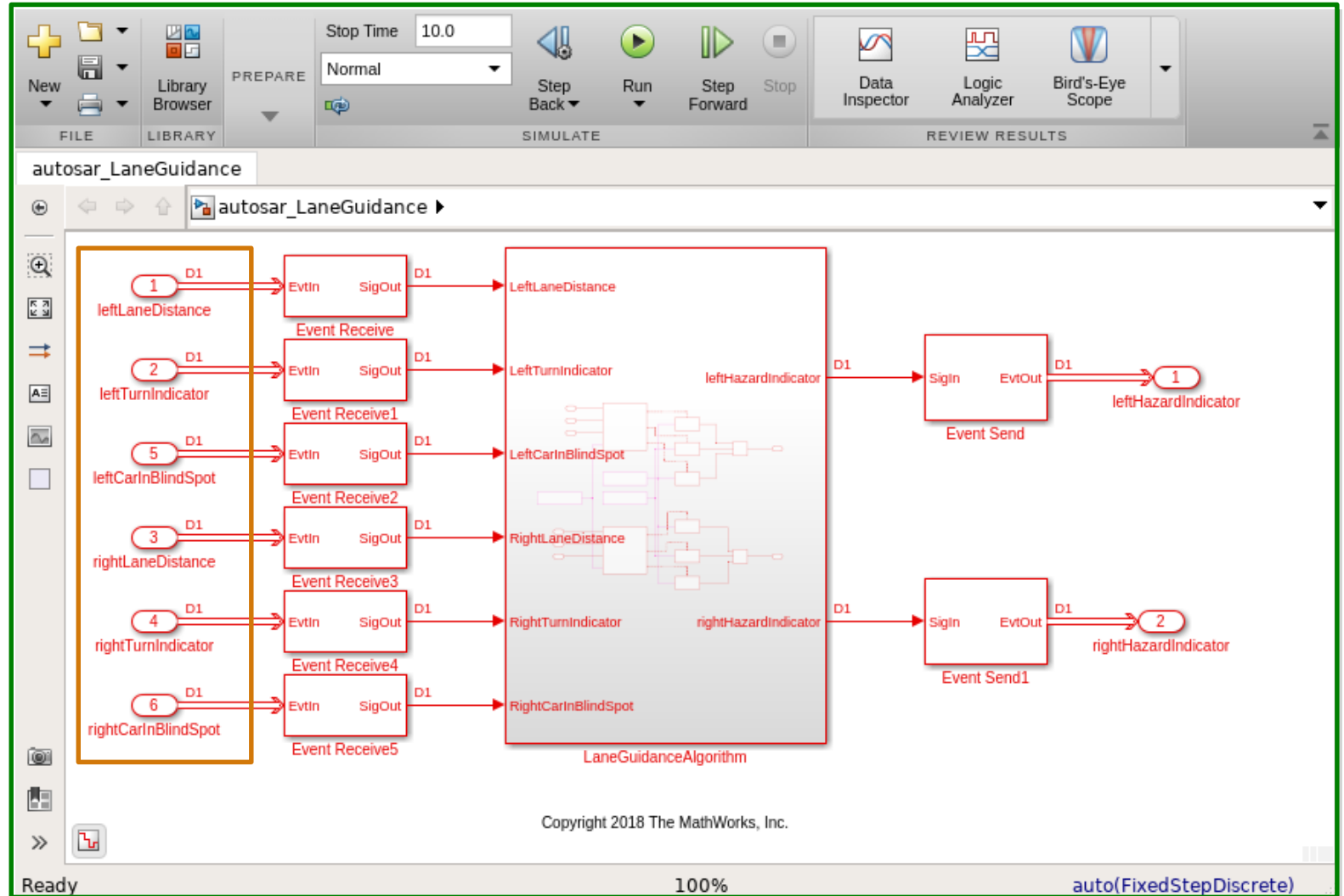
Adaptive Application

RequiredPort

```

"Radar" : {
  // events
  "event" : {
    "leftLaneDistance"
    "leftTurnIndicator"
    "leftCarInBlindSpot"
    "rightLandDistance"
    "rightTurnIndicator"
    "rightCarInBlindSpot"
  },
  // methods
  "method" : {
    "Calibrate"
    "Adjust"
  },
  // fields
  "field" : {
    "updateRate"
  }
}
    
```

MATLAB EXPO 2019



从Adaptive概念到Simulink的映射 2/2

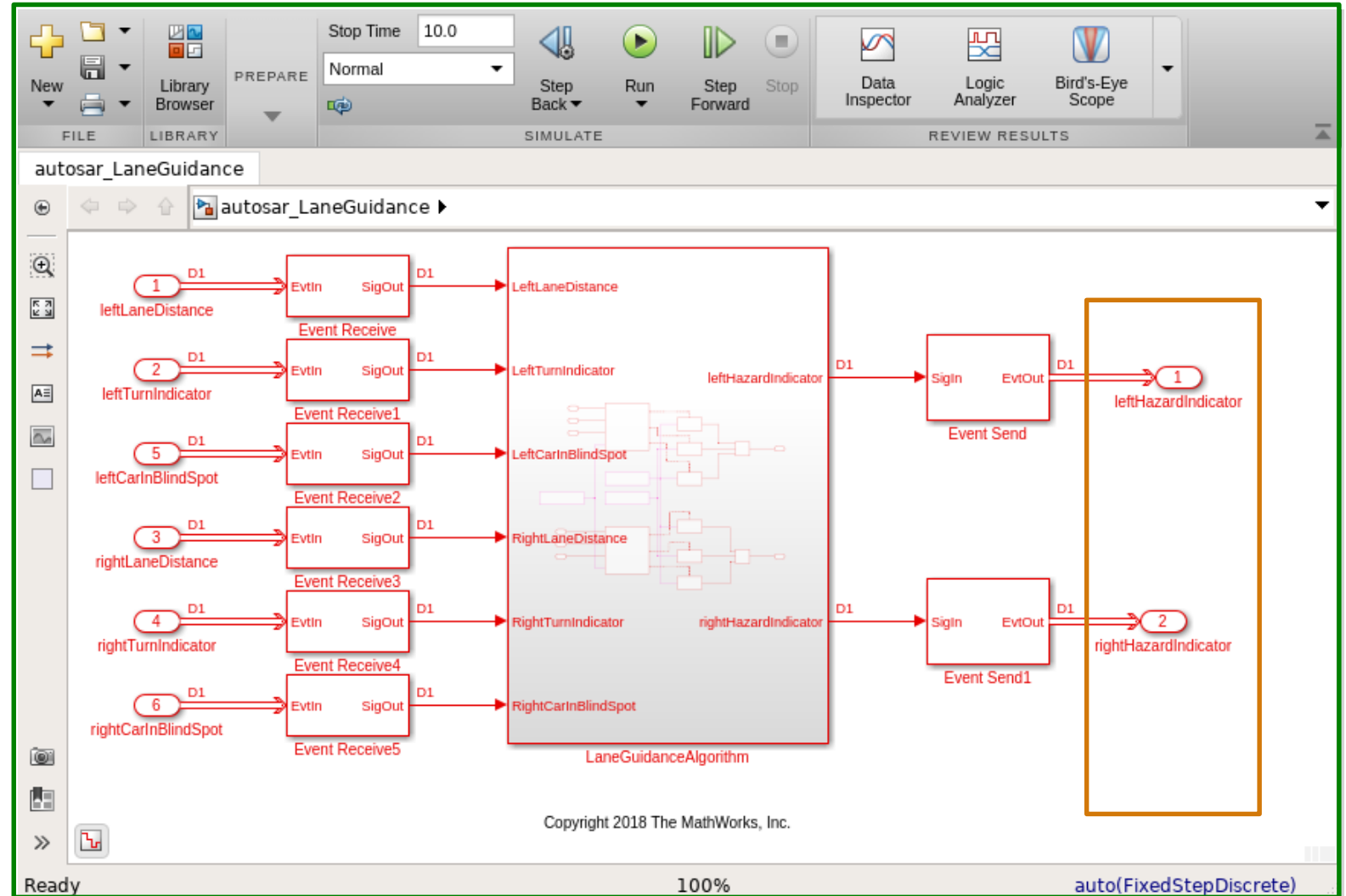


Adaptive Application

ProvidedPort

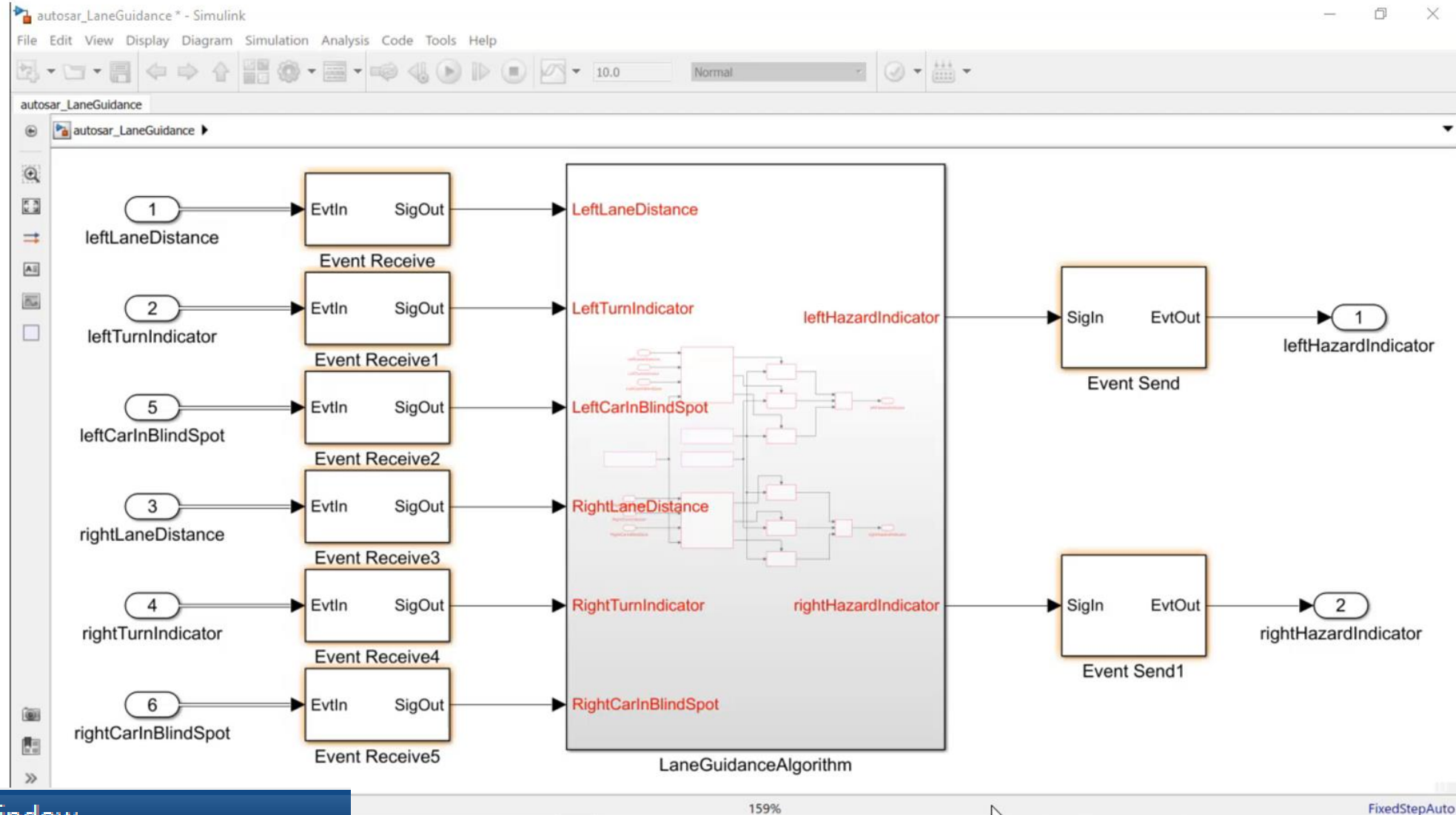
```

"Radar" : {
  // events
  "event" : {
    "leftHazardIndicator"
    "rightHazardIndicator"
  },
  // methods
  "method" : {
    "Calibrate"
    "Adjust"
  },
  // fields
  "field" : {
    "updateRate"
  }
}
    
```



Copyright 2018 The MathWorks, Inc.

案例展示 – 从Simulink模型生成Adaptive平台代码 1/10



Command Window

```
>> autosar_LaneGuidance
```

159%

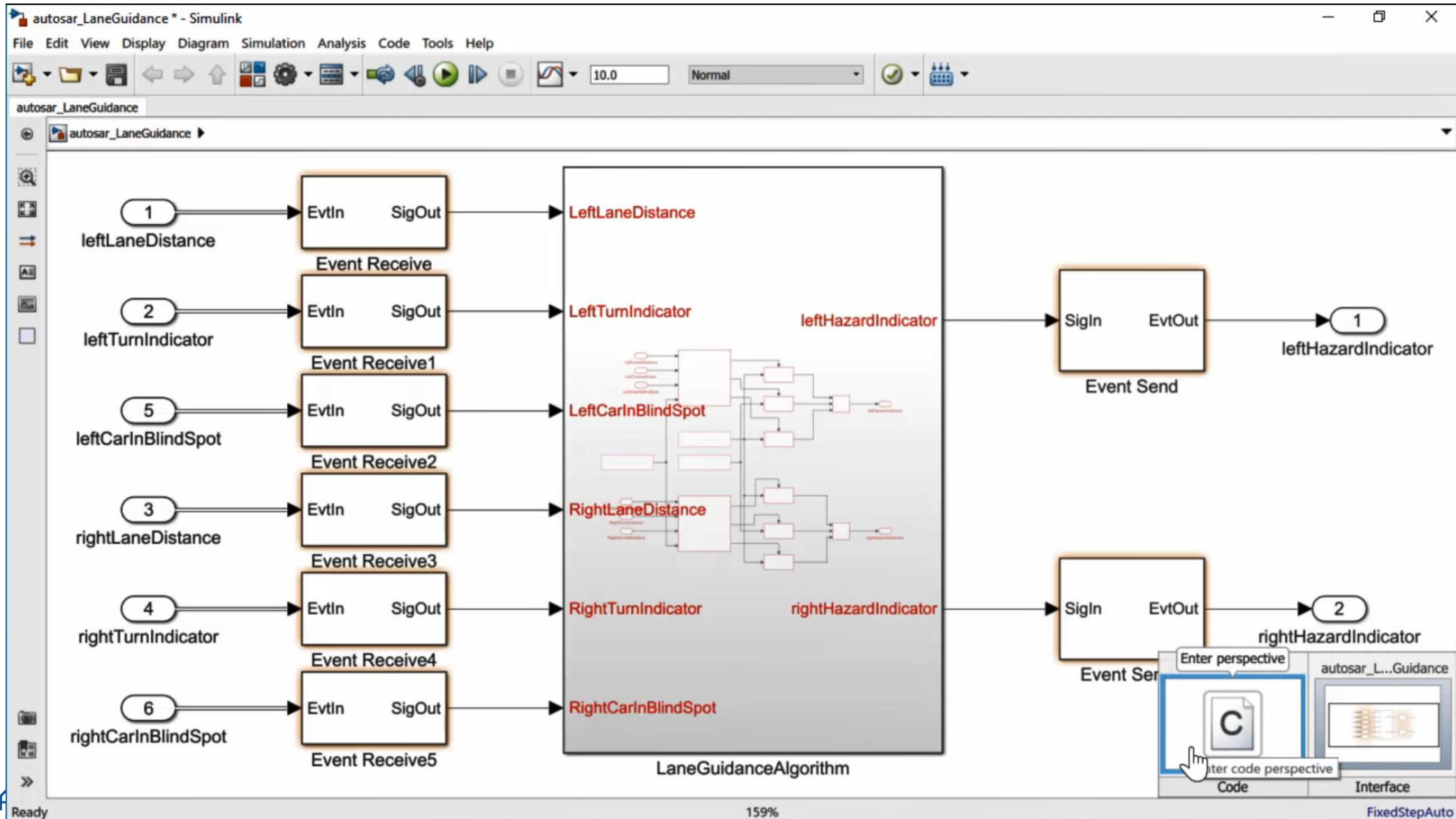
FixedStepAuto

案例展示 – 选择Adaptive平台目标TLC 2/10

The screenshot shows the Simulink Configuration Parameters dialog for the 'autosar_LaneGuidance' model. The 'Code Generation' section is expanded, and the 'System Target File Browser' dialog is open, displaying a list of target files. The selected target is 'autosar_adaptive.tlc', which is described as 'AUTOSAR Adaptive'. The full name of the selected target is shown as 'C:\Program Files\MATLAB\R2019a\toolbox\rtw\targets\AUTOSAR\AUTOSAR\autosar_adaptive.tlc'.

System Target File:	Description:
asap2.tlc	ASAM-ASAP2 Data Definition Target
autosar.tlc	AUTOSAR
autosar_adaptive.tlc	AUTOSAR Adaptive
ert.tlc	Embedded Coder
ert.tlc	Create Visual C/C++ Solution File for Embedded Coder
ert_shrplib.tlc	Embedded Coder (host-based shared library target)
grt.tlc	Generic Real-Time Target
grt.tlc	Create Visual C/C++ Solution File for Simulink Coder

案例展示 – 进入代码视图启动配置过程 3/10



案例展示 – 利用AUTOSAR快速组件设置功能 4/10

The screenshot shows the Simulink interface with the 'AUTOSAR Component Quick Start' dialog box open. The dialog box has a dark blue header with the text 'Set Component'. Below the header, the main content area is divided into two sections. The left section is titled 'Configure AUTOSAR software component properties' and 'Component details:'. It contains the instruction 'Map model to AUTOSAR software component (Adaptive)'. Under this instruction, there are two input fields: 'Component name:' with the value 'autosar_Lane_Guidance' and 'Component package:' with the value '/Company/Powertrain/Components'. The right section is titled 'What to consider' and contains a paragraph of text: 'AUTOSAR Component Quick Start maps a Simulink model to an AUTOSAR software component. For the component, specify an AUTOSAR short name, package path, and component type, or accept default values. Package paths can use an organizational naming pattern, such as /Company/Powertrain/Components. Component type determines the APIs available to the component in the run-time environment.' At the bottom of the dialog box, there are two buttons: 'Help' and 'Next'. The background shows a Simulink model with several blocks labeled 'leftLa', 'leftTu', 'leftCa', 'rightLa', 'rightTu', and 'rightCa'. Two callout boxes with arrows point to 'leftHazardIndicator' (labeled '1') and 'rightHazardIndicator' (labeled '2'). The Simulink window title is 'autosar_Lane_Guidance - Simulink'. The status bar at the bottom shows 'Ready', '159%', and 'FixedStepAuto'.

Set Component

Configure AUTOSAR software component properties

Component details:

Map model to AUTOSAR software component (Adaptive)

Component name:
autosar_Lane_Guidance

Component package:
/Company/Powertrain/Components

What to consider

AUTOSAR Component Quick Start maps a Simulink model to an AUTOSAR software component. For the component, specify an AUTOSAR short name, package path, and component type, or accept default values. Package paths can use an organizational naming pattern, such as /Company/Powertrain/Components. Component type determines the APIs available to the component in the run-time environment.

1 leftHazardIndicator

2 rightHazardIndicator

Help Next

Ready 159% FixedStepAuto

案例展示 - 设置端口事件的代码映射 5/10

autosar_Lane_Guidance - Simulink

File Edit View Display Diagram Simulation Analysis Code Tools Help

autosar_Lane_Guidance

Property Inspector

Inports: leftTurnIndicator

NAME	VALUE
Source	leftTurnIndicator

Code

Port	RequiredPort
Event	leftTurnIndicator

Code Mappings - AUTOSAR SW Component (Adaptive)

Inports Outports

Source	Port	Event
leftLaneDistance	RequiredPort	leftLaneDistance
leftTurnIndicator	RequiredPort	leftTurnIndicator
rightLaneDistance	RequiredPort	rightLaneDistance

Ready View diagnostics 107% auto(FixedStepDiscrete)

案例展示 – Adaptive AUTOSAR属性词典检查 6/10

The screenshot shows the Simulink interface with the AUTOSAR Dictionary tool open. The tool displays a tree view of the dictionary structure. The 'Events' folder under 'RequiredInterface' is selected. The main pane shows a table of event properties.

Name	SwCalibrationAccess	DisplayFormat
leftCarInBlindSpot	ReadOnly	
leftLaneDistance	ReadOnly	
leftTurnIndicator	ReadOnly	
rightCarInBlindSpot	ReadOnly	
rightLaneDistance	ReadOnly	
rightTurnIndicator	ReadOnly	

On the right, the Property Inspector shows the 'Code' section for the selected event:

Port	RequiredPort
Event	leftTurnIndicator

案例展示 – 从AUTOSAR模型生成代码 7/10

Code Generation Report
- □ ×

Find:
↑ ↓ Match Case

Contents

- [Summary](#)
- [Subsystem Report](#)
- [Code Interface Report](#)
- [Traceability Report](#)
- [Static Code Metrics Report](#)
- [Code Replacements Report](#)
- [Coder Assumptions](#)

Generated Code

- [-] **Model files**
 - [autosar_LaneGuidance.cpp](#)
 - [autosar_LaneGuidance.h](#)
- [+] **Shared files (1)**
- [+] **Interface files (2)**
- [+] **Other files (2)**

Code Generation Report for 'autosar_LaneGuidance'

Model Information

Author	The MathWorks, Inc.
Last Modified By	The MathWorks, Inc.
Model Version	1.224
Tasking Mode	SingleTasking

[Configuration settings at time of code generation](#)

Code Information

System Target File	autosar_adaptive.tlc
Hardware Device Type	Intel->x86-64 (Linux 64)
Simulink Coder Version	9.1 (R2019a) 23-Nov-2018
Timestamp of Generated Source Code	Wed Apr 24 17:34:51 2019
Location of Generated Source Code	C:\00_mdSB\R2019a_inst\R2019a\work\autosar_LaneGuidance_autosar_adaptive\
Type of Build	Model
Objectives Specified	Unspecified

Additional Information

Code Generation Advisor	Not run
-------------------------	---------

OK Help

案例展示 – 生成Adaptive C++代码 8/10

Code Generation Report

Find: ara Match Case

Contents

- Summary
- Subsystem Report
- Code Interface Report
- Traceability Report
- Static Code Metrics Report
- Code Replacements Report
- Coder Assumptions

Generated Code

- [-] Model files
 - autosar_LaneGuidance.cpp
 - autosar_LaneGuidance.h
- [+] Shared files (1)
- [+] Interface files (2)
- [-] Other files
 - MainUtils.hpp
 - main.cpp

```

20 // <S27>/IfActionSS4
27 // '<S9>/IfActionSS5'
28 //
29 static void autosar_LaneGuidance>IfActionSS(re
30 {
31 // Inport: '<S18>/In1'
32 *rty_Out1 = rtu_In1;
33 }
34
35 // Function for Chart: '<S1>/Event Receive'
36 boolean_T autosar_LaneGuidanceModelClass::auto
37 (void)
38 {
39 boolean_T isPresent;
40 const ara::com::SampleContainer< ara::com::S
41 *sampleContainer;
42 if (autosar_LaneGuidance_DW.EvtIn_isValid_i)
43 isPresent = true;
44 } else {
45 // Fetch data for event "LeftLaneDistance"
46 if (RequiredPort->leftLaneDistance.Update(
47 // Access event data
48 sampleContainer = &RequiredPort->leftLaneDistance.GetCachedSamples();
49
50 // Copy event data to application
51 autosar_LaneGuidance_DW.EvtIn_msgData_ga = **sampleContainer->begin();
52 autosar_LaneGuidance_DW.EvtIn_msgDataPtr_o =
53 &autosar_LaneGuidance_DW.EvtIn_msgData_ga;
54
55 // Received new event data
56 isPresent = true;
57
58 // Explicitly clean the event data cache
59 RequiredPort->leftLaneDistance.Cleanup();
60 } else {
61 // Event data not received

```

```

34
35 // Function for Chart: '<S1>/Event Receive'
36 boolean_T autosar_LaneGuidanceModelClass::autosar_LaneGuidance_sf_msg_pop_EvtIn
37 (void)
38 {
39 boolean_T isPresent;
40 const ara::com::SampleContainer< ara::com::SamplePtr< const real_T > >
41 *sampleContainer;
42 if (autosar_LaneGuidance_DW.EvtIn_isValid_i) {
43 isPresent = true;
44 } else {
45 // Fetch data for event "LeftLaneDistance" from ARA middleware
46 if (RequiredPort->leftLaneDistance.Update()) {
47 // Access event data
48 sampleContainer = &RequiredPort->leftLaneDistance.GetCachedSamples();
49
50 // Copy event data to application

```

OK Help

案例展示 – 生成软件组件描述文件 9/10

Code Generation Report

Find: Match Case

Contents

- [Summary](#)
- [Subsystem Report](#)
- [Code Interface Report](#)
- [Traceability Report](#)
- [Static Code Metrics Report](#)
- [Code Replacements Report](#)
- [Coder Assumptions](#)

Generated Code

- [-] Model files**
 - [autosar_LaneGuidance.cpp](#)
 - [autosar_LaneGuidance.h](#)
- [-] Shared files**
 - [rtwtypes.h](#)
- [-] Interface files**
 - [autosar_LaneGuidance.arxml](#)
 - [rtmodel.h](#)
- [+] Other files (2)**

File: autosar_LaneGuidance.arxml

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <!--
3 Auto generated XML Component Description for model autosar_LaneGuidance
4 Model version      : 1.224
5 Simulink Coder version : Simulink Coder 9.1 (R2019a) 23-Nov-2018
6 XML source code generated on : Thu Apr 25 14:13:48 2019
7 Model Checksum      : 1794539629 1582017647 3206248494 1394045784
8 -->
9 <AUTOSAR xmlns="http://autosar.org/schema/r4.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://autosar.org/schema/r4.0 AUTOSAR_00046.x
10 <AR-PACKAGES>
11   <AR-PACKAGE>
12     <SHORT-NAME>LaneGuidance_pkg</SHORT-NAME>
13   <AR-PACKAGES>
14     <AR-PACKAGE>
15       <SHORT-NAME>LaneGuidance_sw</SHORT-NAME>
16     <ELEMENTS>
17       <ADAPTIVE-APPLICATION-SW-COMPONENT-TYPE UUID="6574ed24-7dad-53cc-e7ac-01f60699f406">
18         <SHORT-NAME>LaneGuidance</SHORT-NAME>
19         <PORTS>
20           <R-PORT-PROTOTYPE UUID="a8adc3c3-bbb1-575e-fbc6-0fcf8164f622">
21             <SHORT-NAME>RequiredPort</SHORT-NAME>
22             <REQUIRED-COM-SPECS>
23               <QUEUED-RECEIVER-COM-SPEC>
24                 <DATA-ELEMENT-REF DEST="VARIABLE-DATA-PROTOTYPE">/LaneGuidance_pkg/LaneGuidance_if/RequiredInterface/leftLaneDistance</DATA-
25                 <HANDLE-OUT-OF-RANGE>NONE</HANDLE-OUT-OF-RANGE>
26                 <USES-END-TO-END-PROTECTION>>false</USES-END-TO-END-PROTECTION>
27                 <QUEUE-LENGTH>10</QUEUE-LENGTH>
28               </QUEUED-RECEIVER-COM-SPEC>
29               <QUEUED-RECEIVER-COM-SPEC>
30                 <DATA-ELEMENT-REF DEST="VARIABLE-DATA-PROTOTYPE">/LaneGuidance_pkg/LaneGuidance_if/RequiredInterface/leftTurnIndicator</DATA-
31                 <HANDLE-OUT-OF-RANGE>NONE</HANDLE-OUT-OF-RANGE>
32                 <USES-END-TO-END-PROTECTION>>false</USES-END-TO-END-PROTECTION>
33                 <QUEUE-LENGTH>10</QUEUE-LENGTH>
34               </QUEUED-RECEIVER-COM-SPEC>
35               <QUEUED-RECEIVER-COM-SPEC>
36                 <DATA-ELEMENT-REF DEST="VARIABLE-DATA-PROTOTYPE">/LaneGuidance_pkg/LaneGuidance_if/RequiredInterface/rightLaneDistance</DATA-
37                 <HANDLE-OUT-OF-RANGE>NONE</HANDLE-OUT-OF-RANGE>
38                 <USES-END-TO-END-PROTECTION>>false</USES-END-TO-END-PROTECTION>
39                 <QUEUE-LENGTH>10</QUEUE-LENGTH>
40               </QUEUED-RECEIVER-COM-SPEC>
```

OK Help

案例展示 – 为独立应用生成参考主函数入口10/10

The screenshot displays the 'Code Generation Report' window. On the left, a navigation pane shows a tree view of generated files. The 'Other files' section is expanded, showing 'MainUtils.hpp' and 'main.cpp', both highlighted in yellow. The main area on the right shows the C++ source code for the main function, with line numbers 45 through 79. The code includes comments in green and C++ code in black. The 'main' function starts with variable declarations and a logging setup, followed by a try-catch block for reporting the execution state, and ends with semaphore initialization and error handling.

Contents

- Summary
- Subsystem Report
- Code Interface Report
- Traceability Report
- Static Code Metrics Report
- Code Replacements Report
- Coder Assumptions

Generated Code

- [-] Model files
 - [autosar_LaneGuidance.cpp](#)
 - [autosar_LaneGuidance.h](#)
- [+] Shared files (1)
- [+] Interface files (2)
- [-] Other files
 - [MainUtils.hpp](#)
 - [main.cpp](#)

```

45  /*      will inherit the block and not respond to those */
46  /*      signals. */
47  /* - Responds to baserate_tick semaphore posts and runs */
48  /* applicable AsyncFunctionCalls. */
49  int main()
50  {
51      autosar_LaneGuidanceModelClass model;
52
53      /* These tick variables represent how many base rate */
54      /* periods to wait before running a step function. For */
55      /* example, step1_ticks=3 indicates every */
56      /* third base rate tick, we should run step1(). */
57      int const step_ticks = 1;
58      double const baserate = 0.100000;
59      ara::log::Logger & log{
60          ara::log::CreateLogger("autosar_LaneGuidance",
61              "Logger for autosar_LaneGuidance's main function.")
62      }
63
64      ;
65
66      /* Report Execution state */
67      ara::exec::ExecutionClient exec_client;
68      try {
69          exec_client.ReportExecutionState(ara::exec::ExecutionState::kRunning);
70      } catch (std::exception const & e) {
71          log.LogError() << "Unable to report running state: " << e.what();
72          std::exit(EXIT_FAILURE);
73      }
74
75      if (sem_init(&baserate_tick, 0, 0) == -1) {
76          log.LogError() << "Unable to initialize baserate_tick semaphore: " << std::
77              strerror(errno);
78          std::exit(EXIT_FAILURE);
79      }

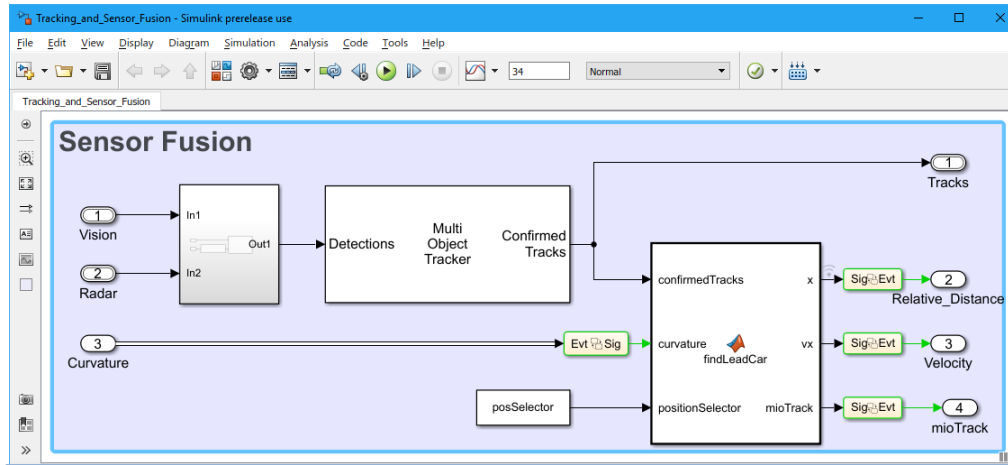
```

[-] Other files

[MainUtils.hpp](#)

[main.cpp](#)

从Adaptive AUTOSAR生成C++代码



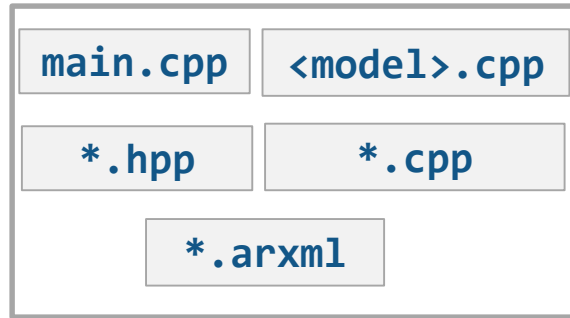
Configuration Parameters:

Target selection

System target file:

Language:

Description:



AUTOSAR内嵌支持

1. 模型配置
 - ✓ 目标TLC
 - ✓ AUTOSAR词典
2. 生成C++代码

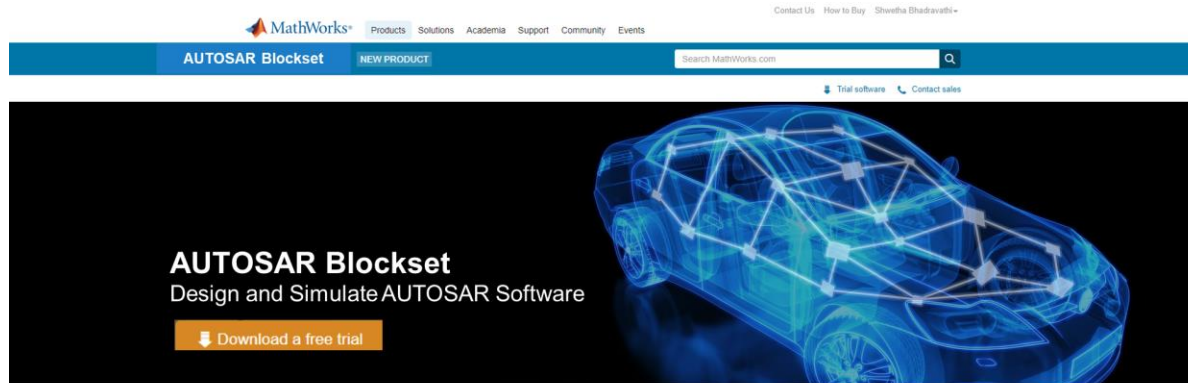
AUTOSAR Dictionary

Filter Contents

- AUTOSAR
 - AdaptiveApplications
 - Tracking_and_Sensor_Fusion
 - RequiredPorts
 - ProvidedPorts
 - Service Interfaces
 - RadarInterface
 - Events**
 - Namespaces
 - ServiceInterface2
 - XML Options

Name	SwCalibrationAccess
Curvature	ReadOnly
Prediction_Time	ReadOnly
Radar	ReadOnly
Vision	ReadOnly

AUTOSAR Blockset 官方主页



AUTOSAR Blockset provides an AUTOSAR dictionary and blocks for developing Classic and Adaptive AUTOSAR software using Simulink® models. You can define AUTOSAR software component properties, interfaces, and datatypes, and map them to existing Simulink models using the AUTOSAR editor. Alternatively, the blockset provides an application interface that lets you automatically generate new Simulink models for AUTOSAR by importing software component and composition descriptions from AUTOSAR XML files.

AUTOSAR Blockset provides blocks and constructs for AUTOSAR library routines and Basic Software (BSW) services, including NVRAM and Diagnostics. By simulating the BSW services together with your application software model, you can verify your AUTOSAR ECU software without leaving Simulink.

AUTOSAR Blockset supports C and C++ production code generation and AUTOSAR XML file export (with Embedded Coder®). It is qualified for use with the ISO 26262 standard (with IEC Certification Kit).

