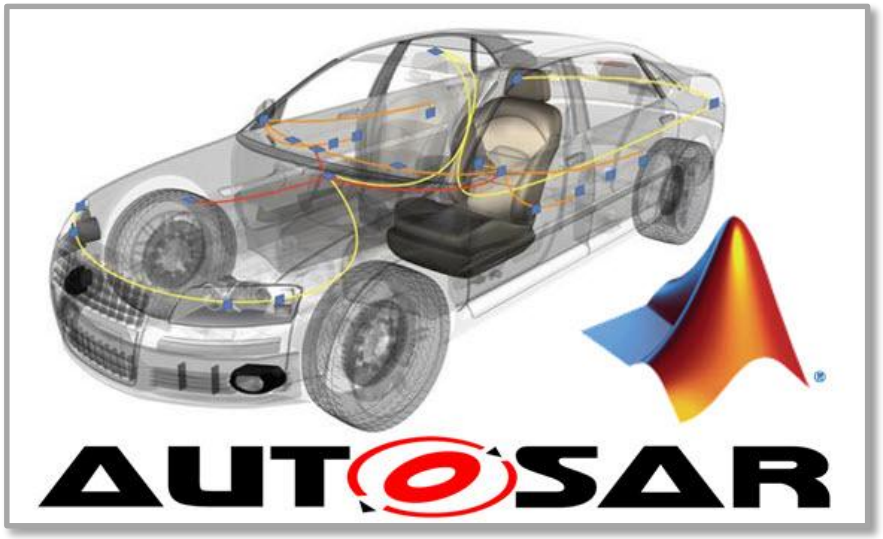
开发AUTOSAR兼容的嵌入式软件

龚小平
迈斯沃克中国区
高级软件项目支持工程师



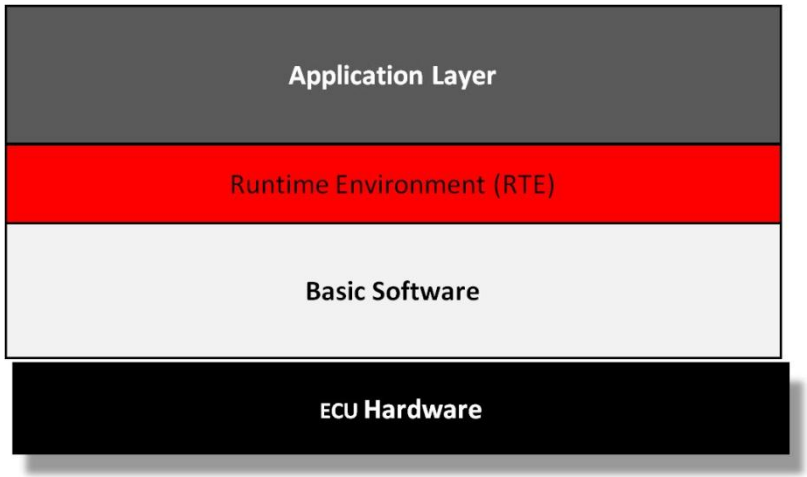
© 2014 The MathWorks, Inc.

内容

- AUTOSAR应用开发介绍
- AUTOSAR的工作流程
  - 从下往上：从Simulink已有模型起步
  - 从上往下：从软件组件描述文件起步
- AUTOSAR中的标定和测量
- 结论和方向

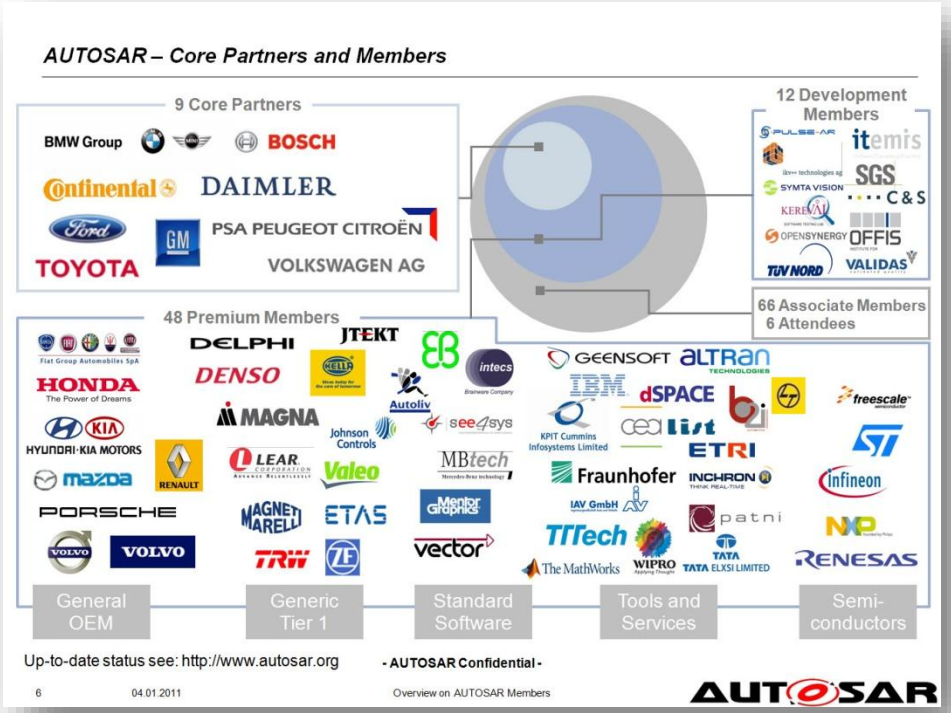
AUTOSAR应用开发介绍

AUTOSAR是什么?



- AUTOSAR® (AUTomotive Open System ARchitecture) 是一个开放的标准化的汽车软件架构

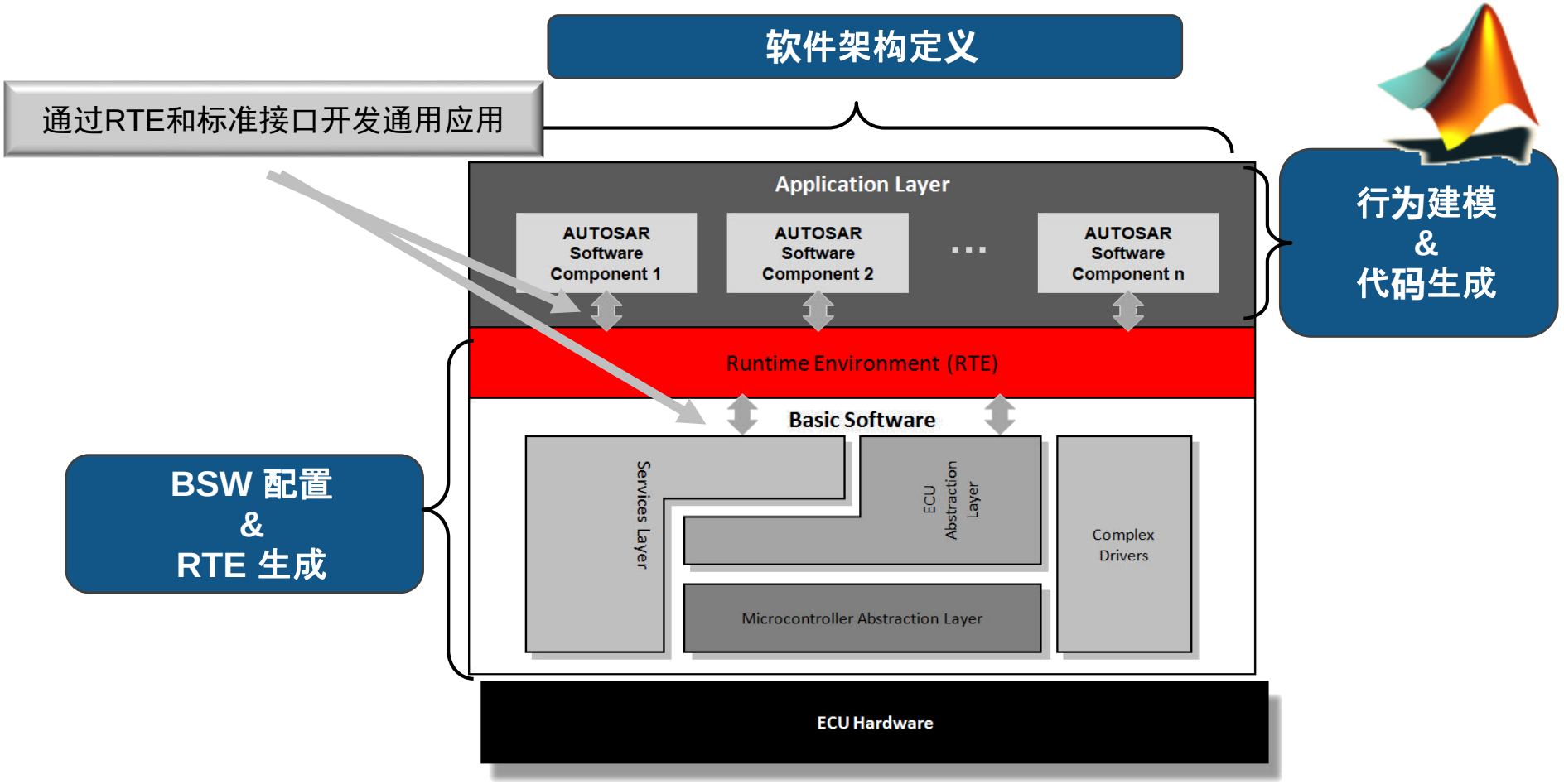
- 由来自全球汽车行业的超过140个成员组成



Source: AUTOSAR, status 04.01.2011

AUTOSAR是什么?

三层式架构



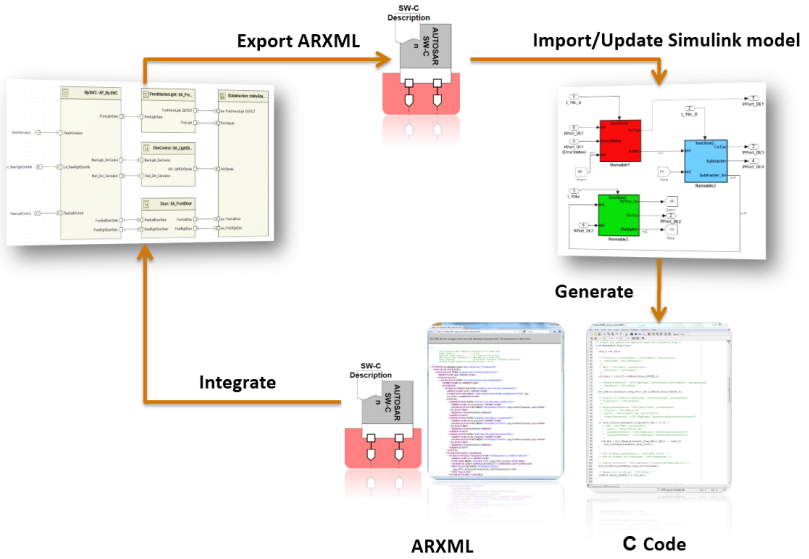
MathWorks的AUTOSAR路径

无需独立的AUTOSAR模块集	• 通过映射生成代码
AUTOSAR 软件组件到Simulink路径	• Simulink用于行为建模（算法开发） • 导入和导出软件组件描述文件(ARXML)
同步生成C代码和ARXML文件	• 代码和软件描述文件一致

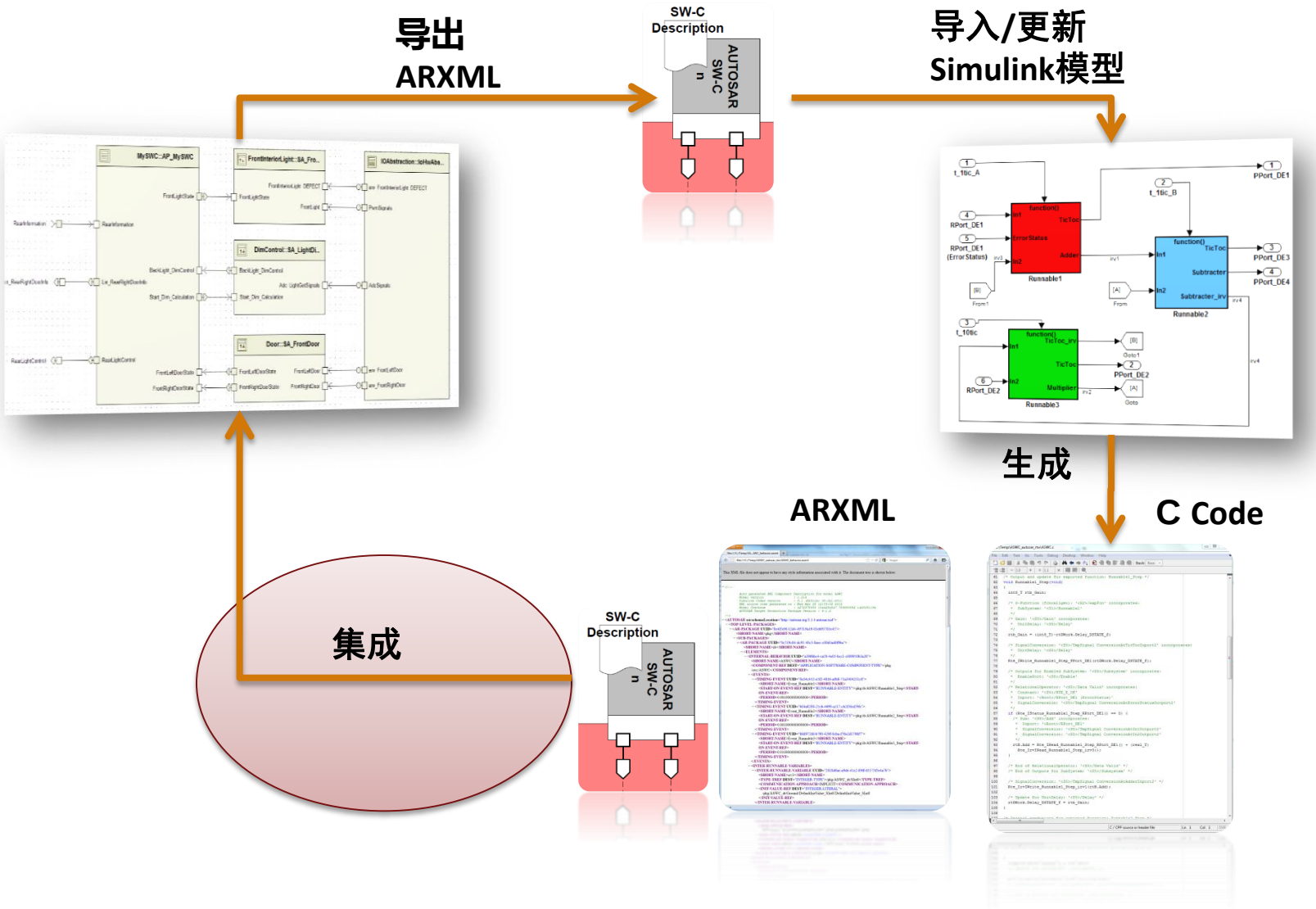
AUTOSAR开发起步点

- 从已有的Simulink模型起步

- 从包含AUTOSAR组件描述的ARXML文件起步



AUTOSAR工作流程及与AAT的集成



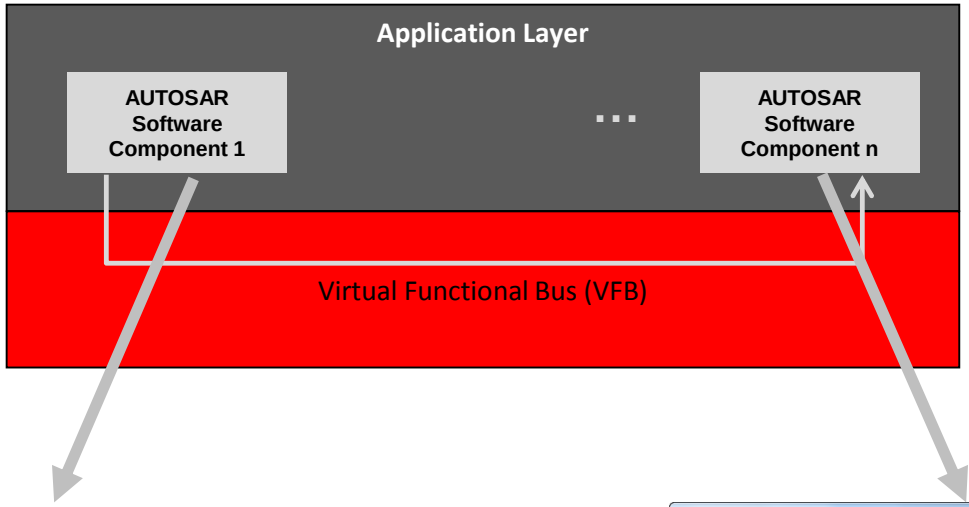
可生成的ARXML

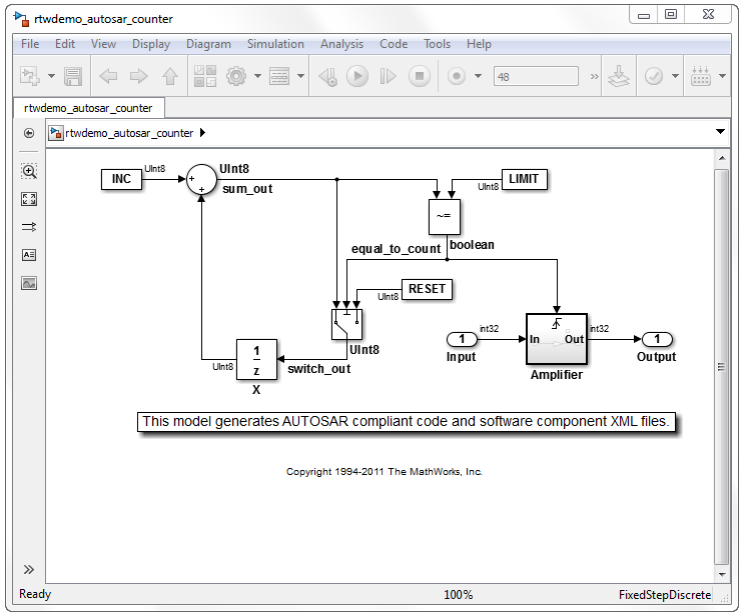
- 从Simulink生成的ARXML包括：
  - Component and Internal Behavior
  - Data Types
  - Implementation information
    - lists all generated source and arxml files
  - Interfaces
  - Other Entities

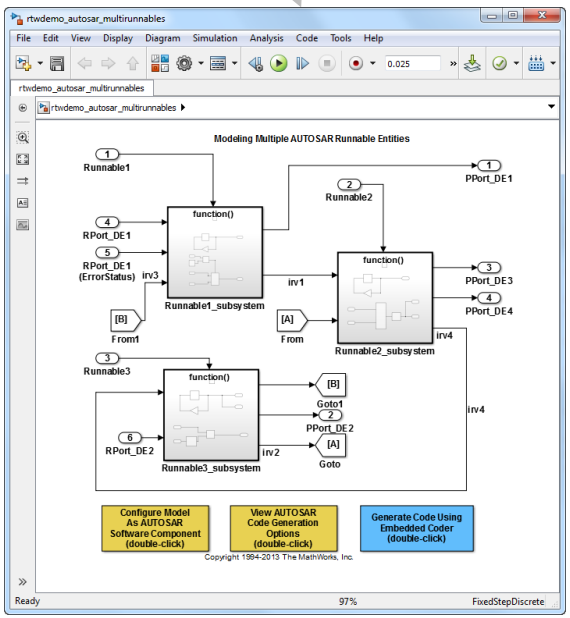
AUTOSAR工作流程

- 从Simulink模型起步

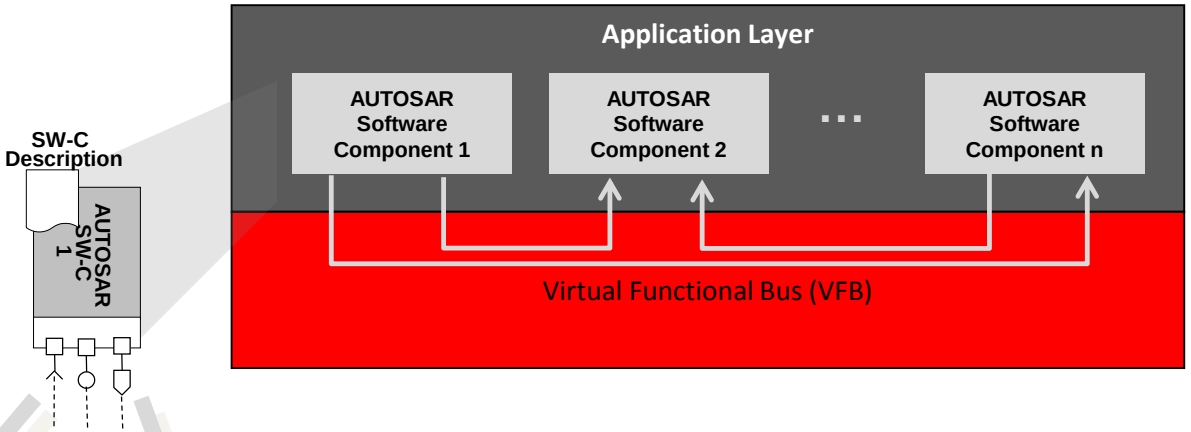
概览: AUTOSAR组件建模

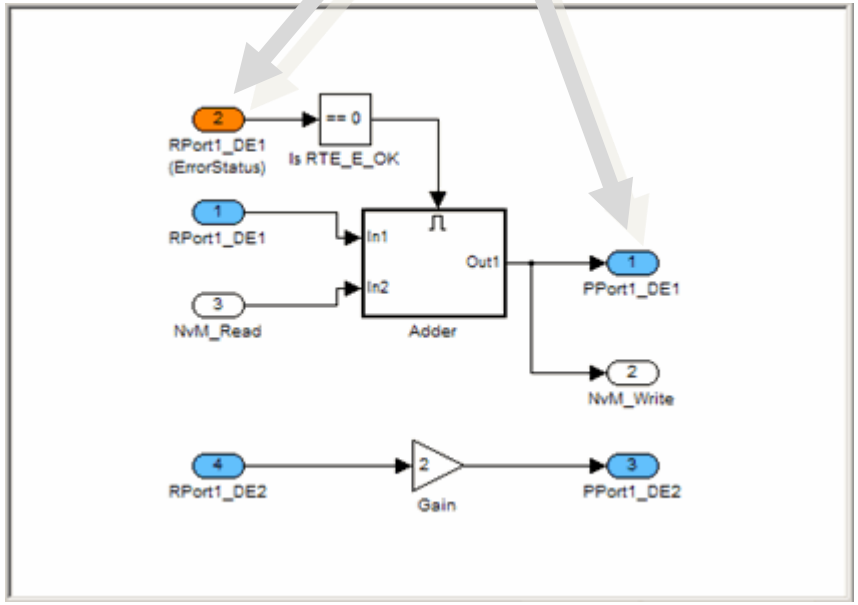






概览: 映射Simulink到 AUTOSAR

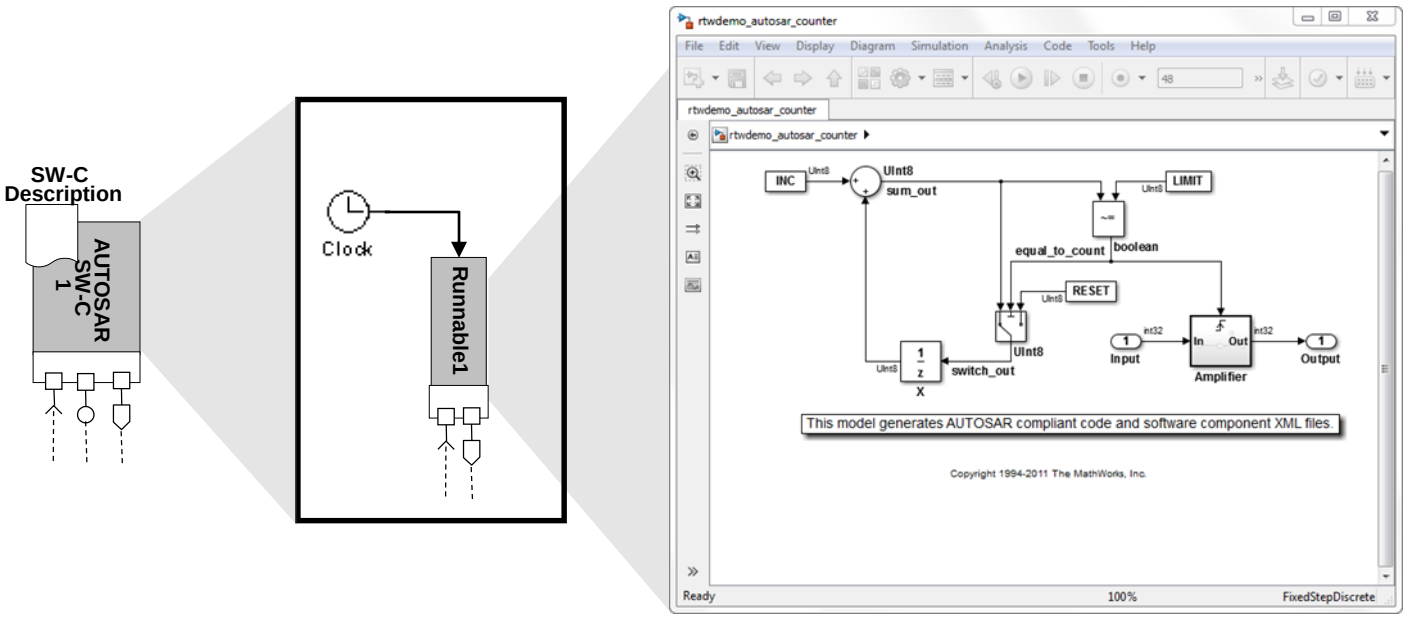




Data Elements of AUTOSAR S/R Ports Implemented with Simulink I/O

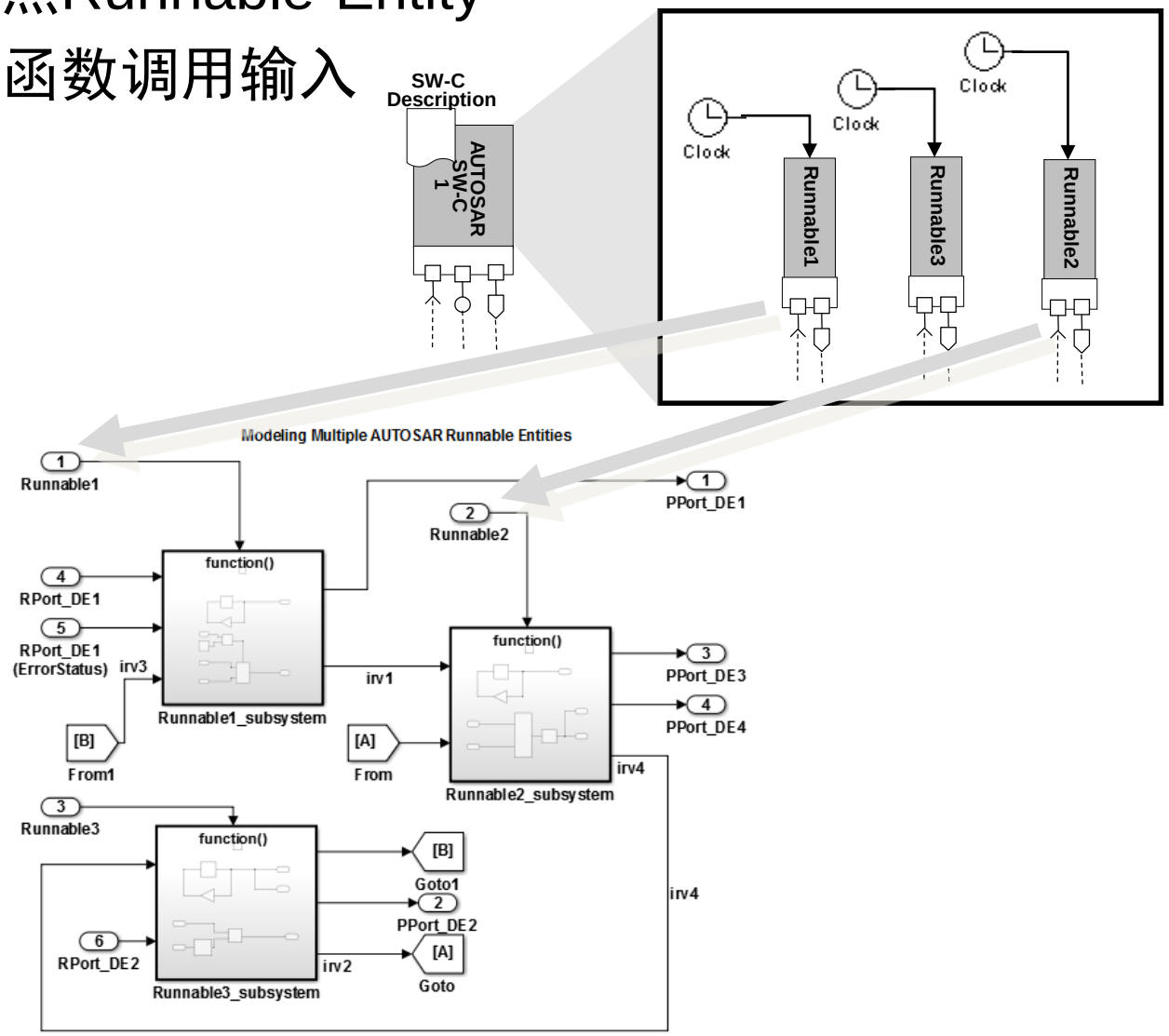
映射Simulink到 AUTOSAR

- 单周期Runnable-Entity
- 采用顶层模型实现

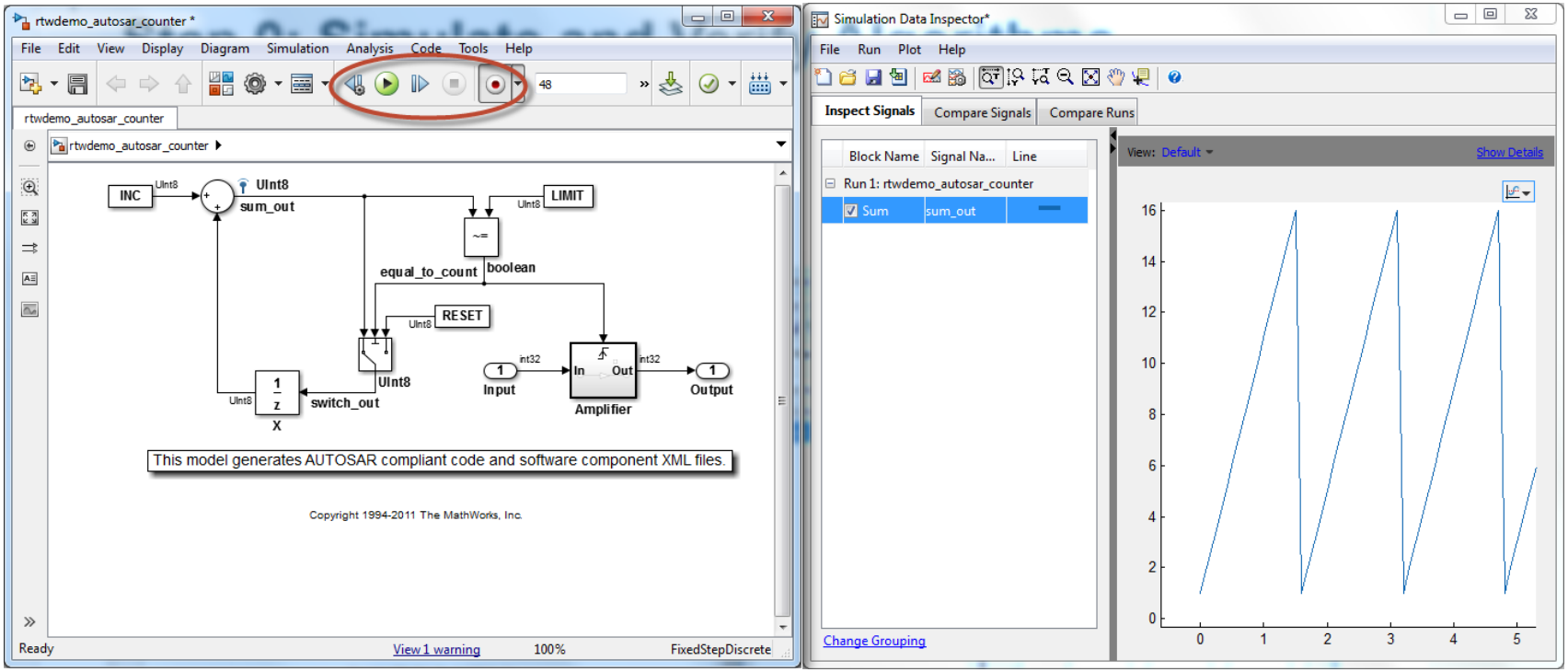


映射Simulink到 AUTOSAR

- 多入口点Runnable-Entity
- 映射到函数调用输入

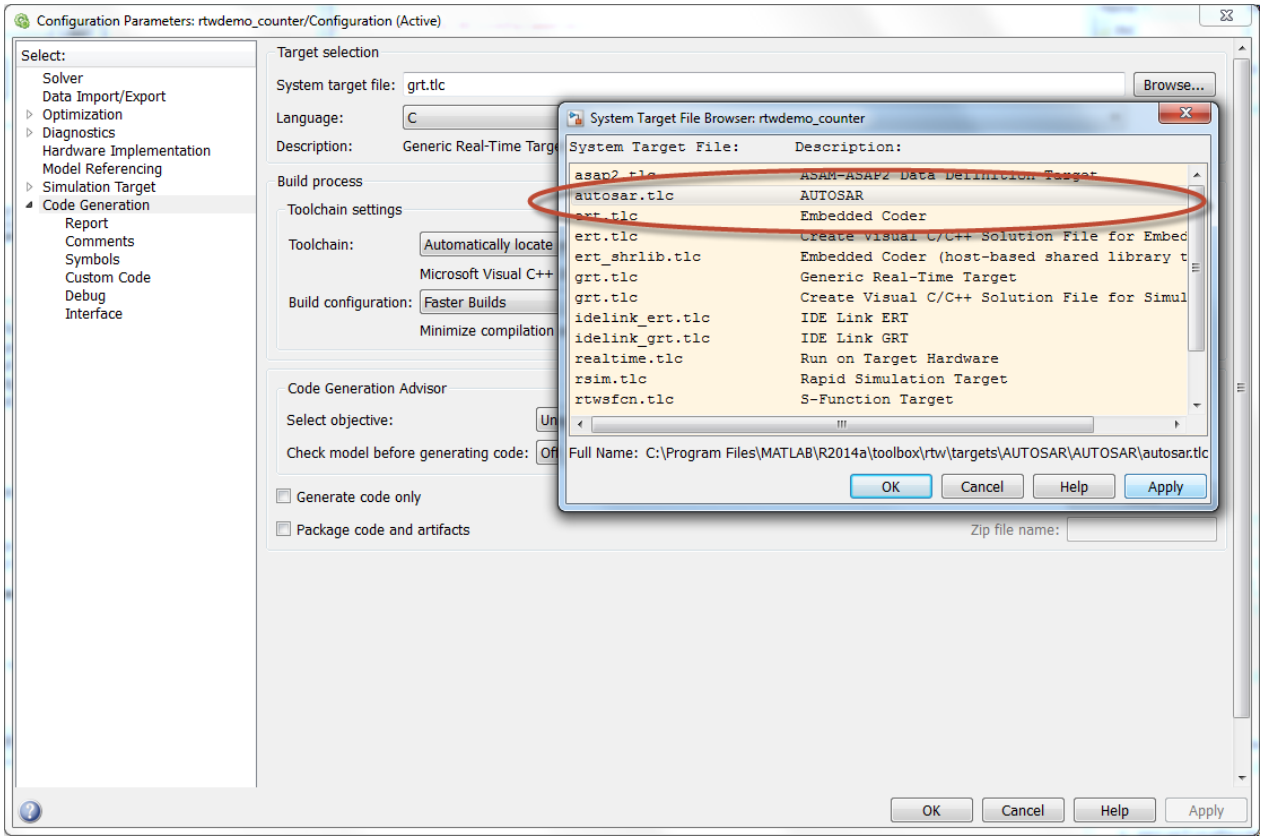


Step 0: 算法仿真和验证

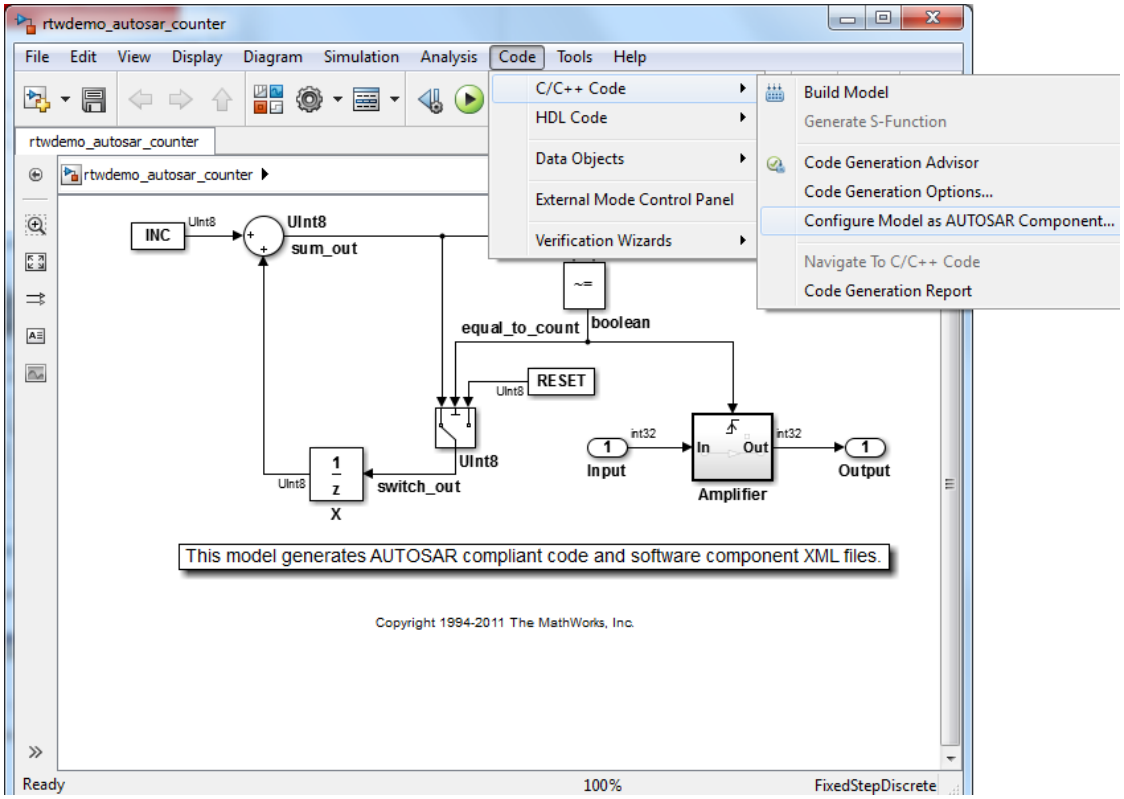


>> rtwdemo_autosar_counter

Step1: 选择AUTOSAR目标TLC

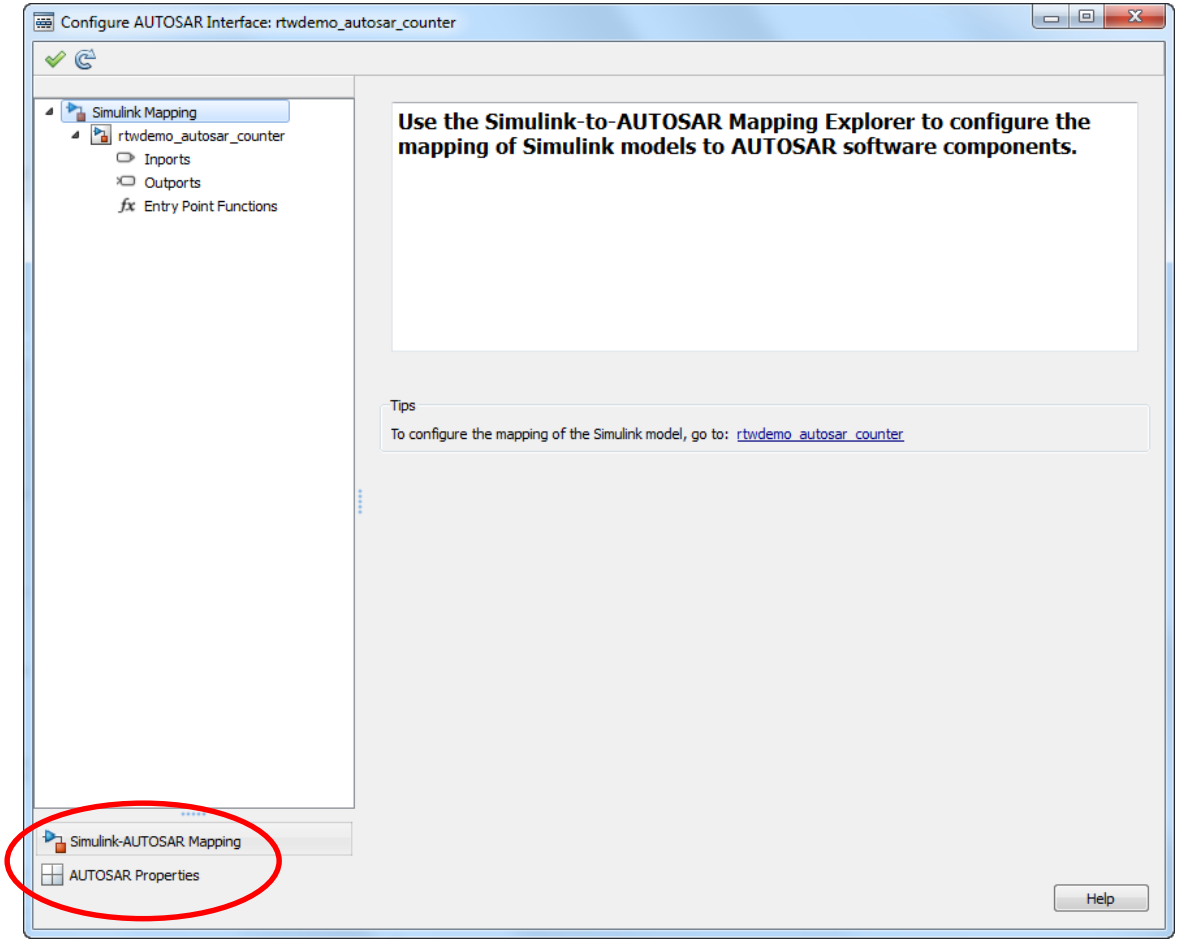


Step 2: 启动AUTOSAR配置



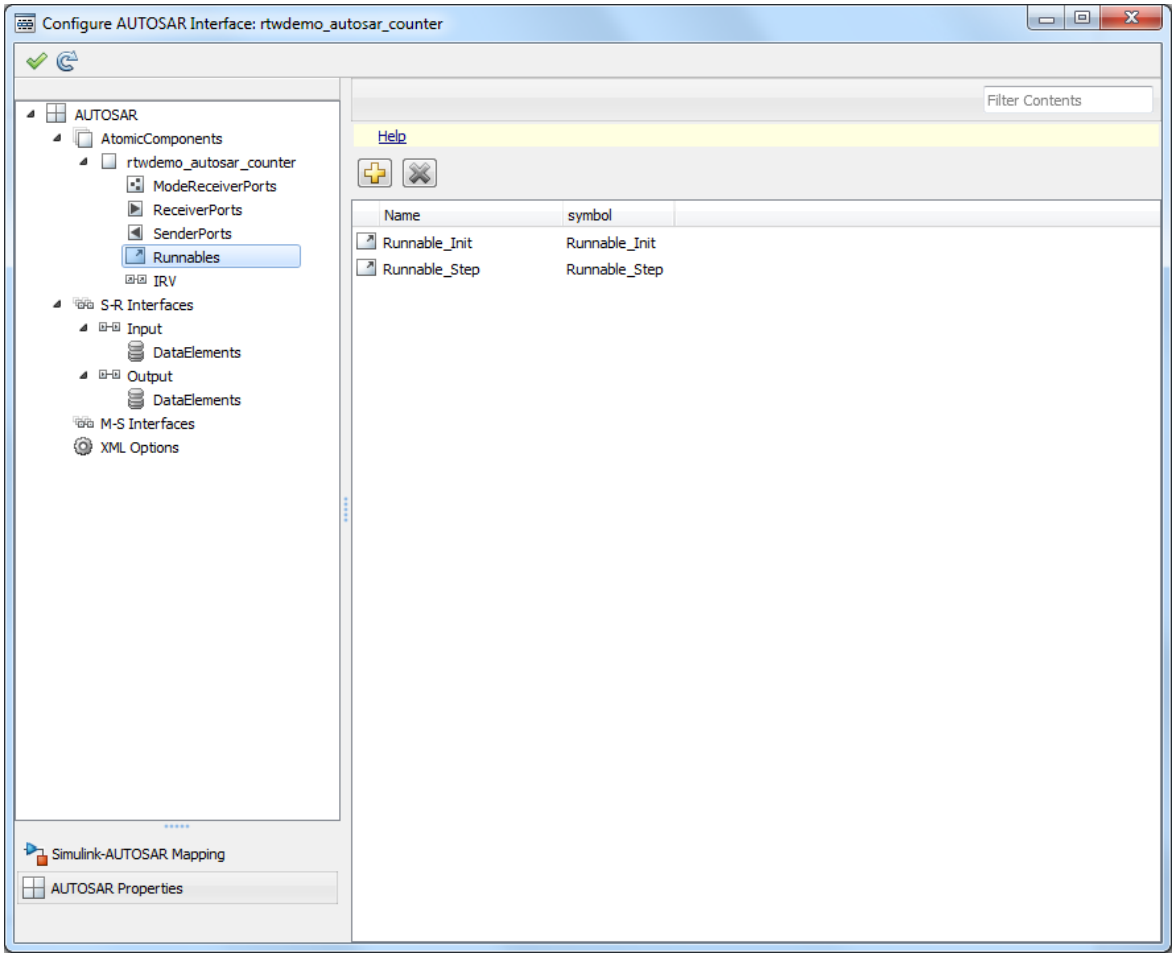
>> rtwdemo_autosar_counter

Simulink-AUTOSAR 映射编辑器



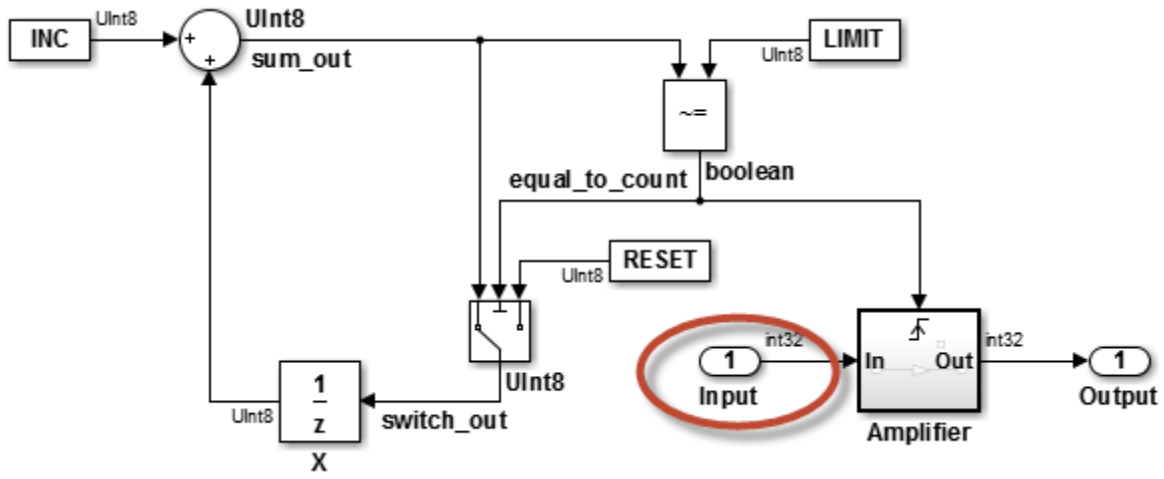
- 查看/编辑 AUTOSAR 属性
- 在Simulink Mapping view 中 执行 Simulink实体到AUTOSAR的映射

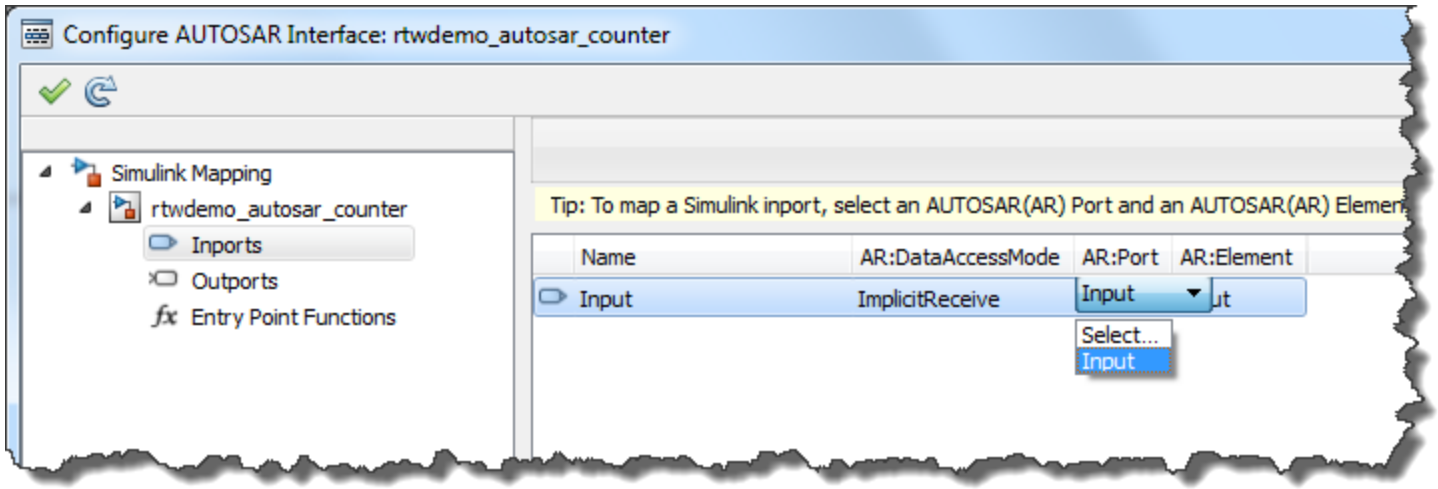
编辑AUTOSAR属性



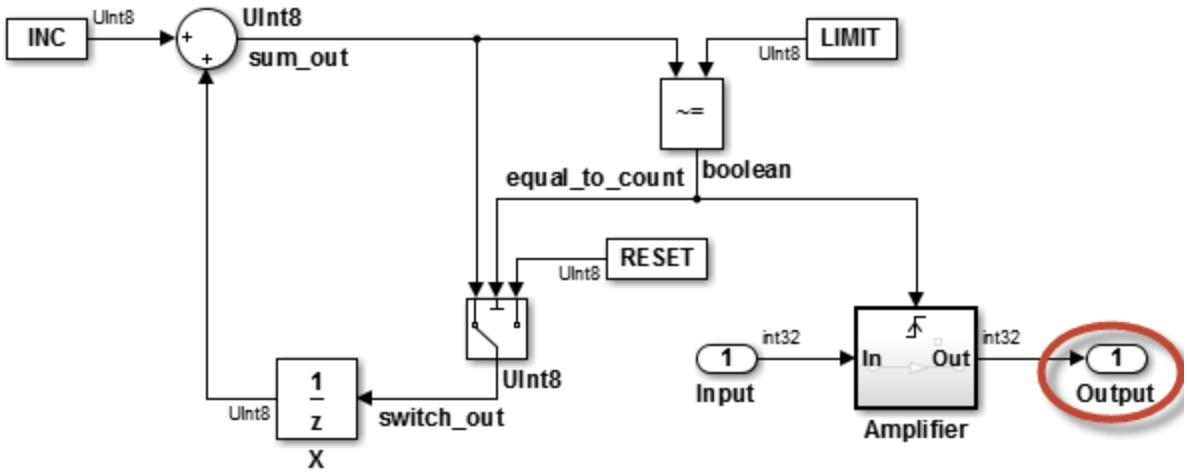
- 增加/移除AUTOSAR实体
- 编辑AUTOSAR实体属性
- 配置ARXML选项

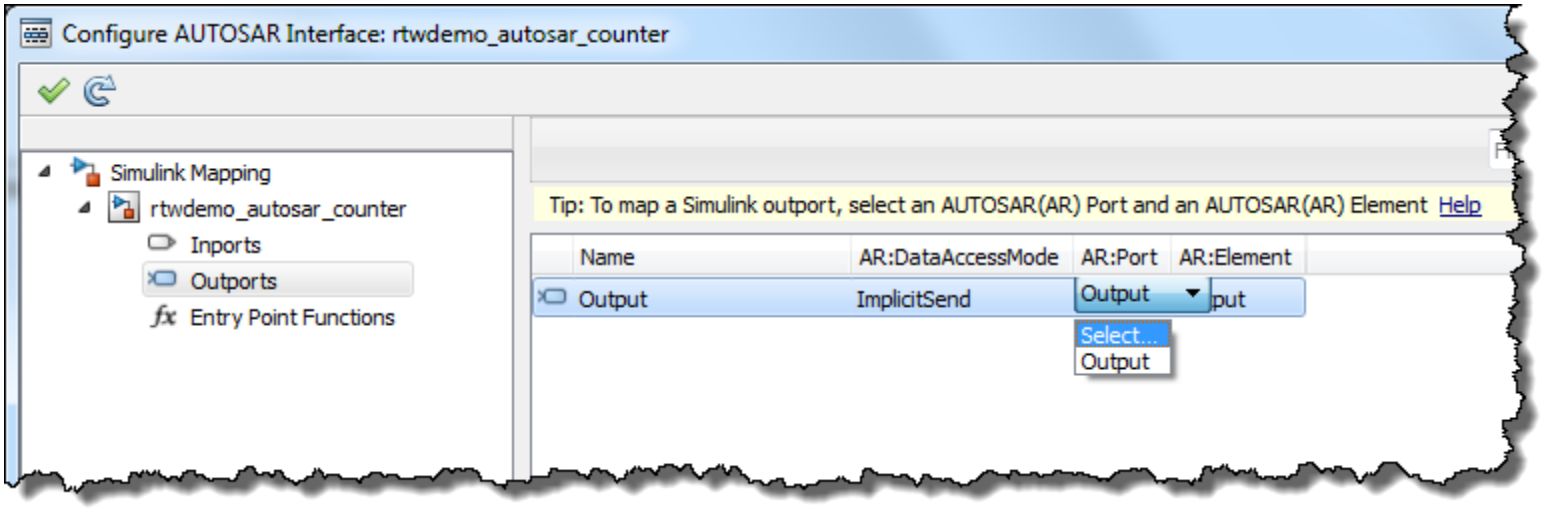
映射输入端口到AUTOSAR的Receiver Ports



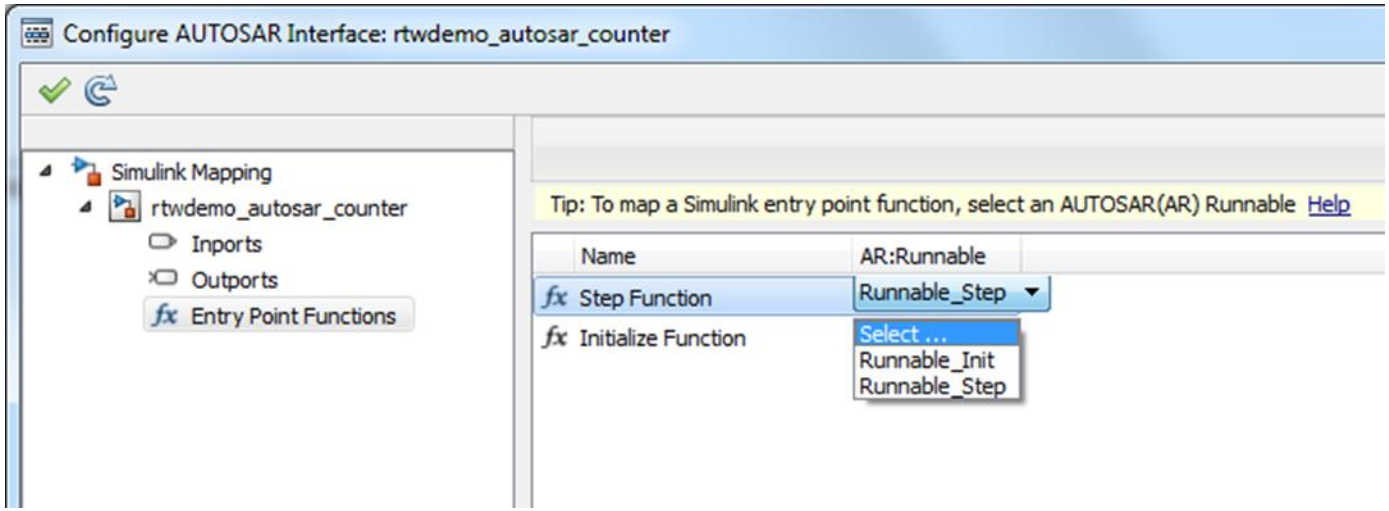


映射输出端口到AUTOSAR的Sender Ports

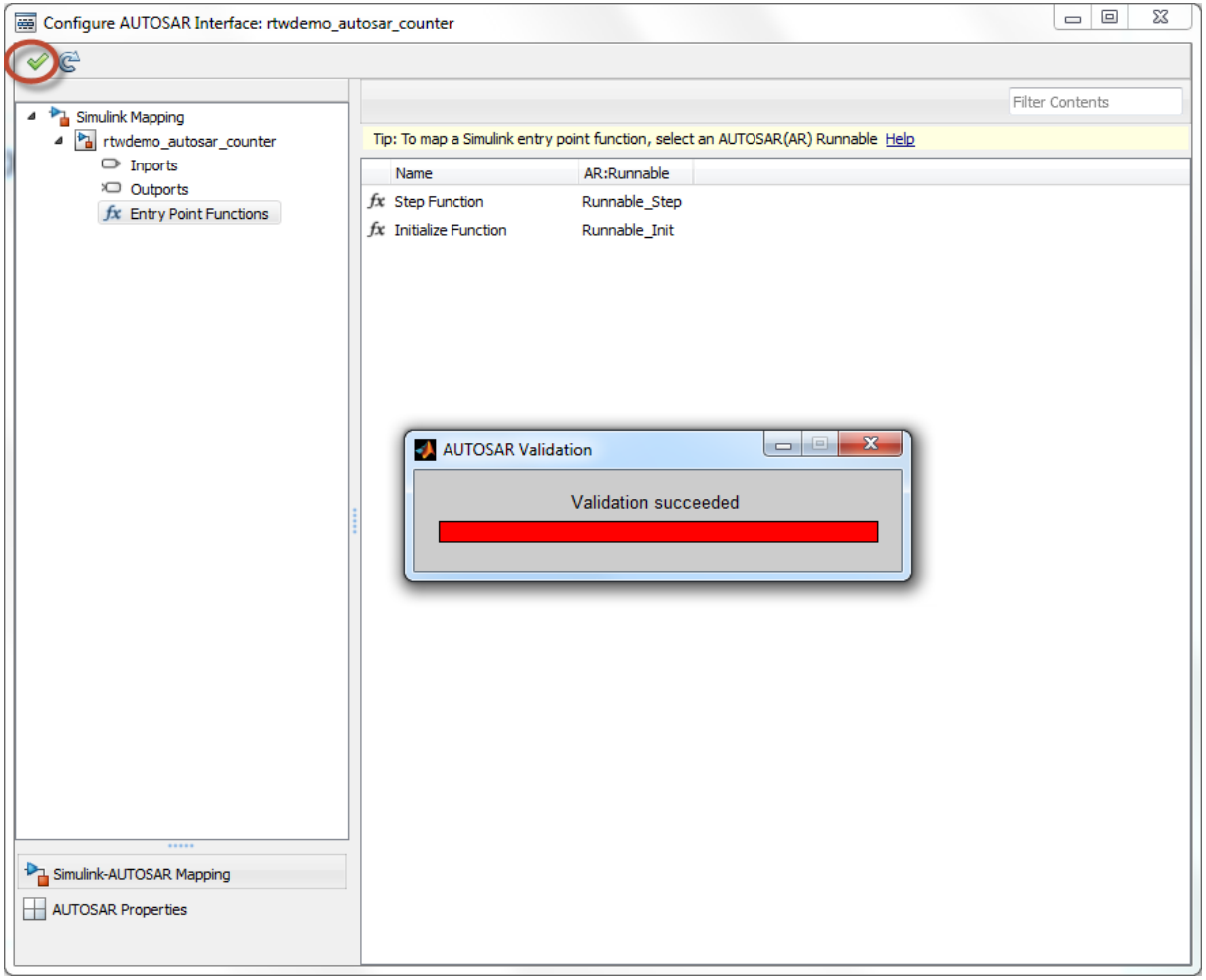




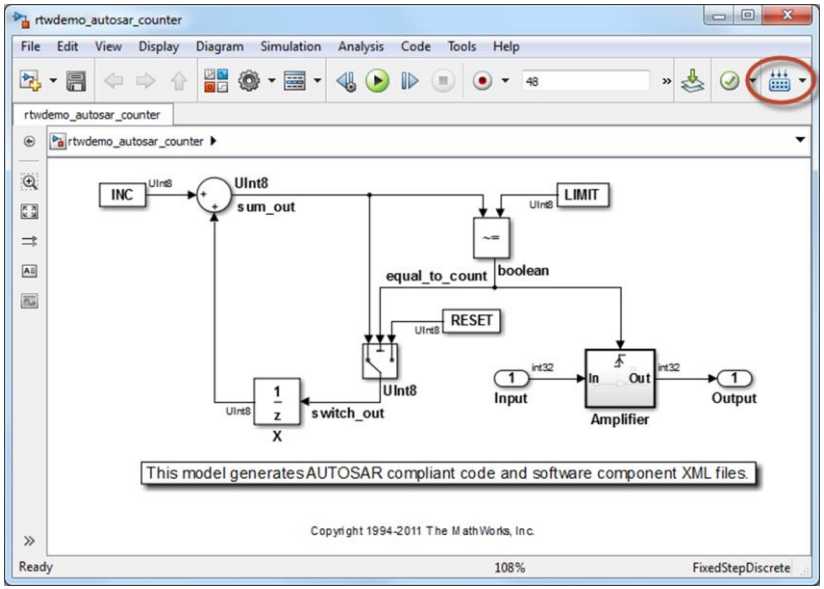
映射模型的入口点函数


Configure AUTOSAR Interface: rtwdemo_autosar_counter
Simulink Mapping
rtwdemo_autosar_counter
Inports
Outports
Entry Point Functions
Tip: To map a Simulink entry point function, select an AUTOSAR(AR) Runnable Help
Name
AR:Runnable
Step Function
Runnable_Step
Initialize Function
Select ...
Runnable_Init
Runnable_Step

确认配置

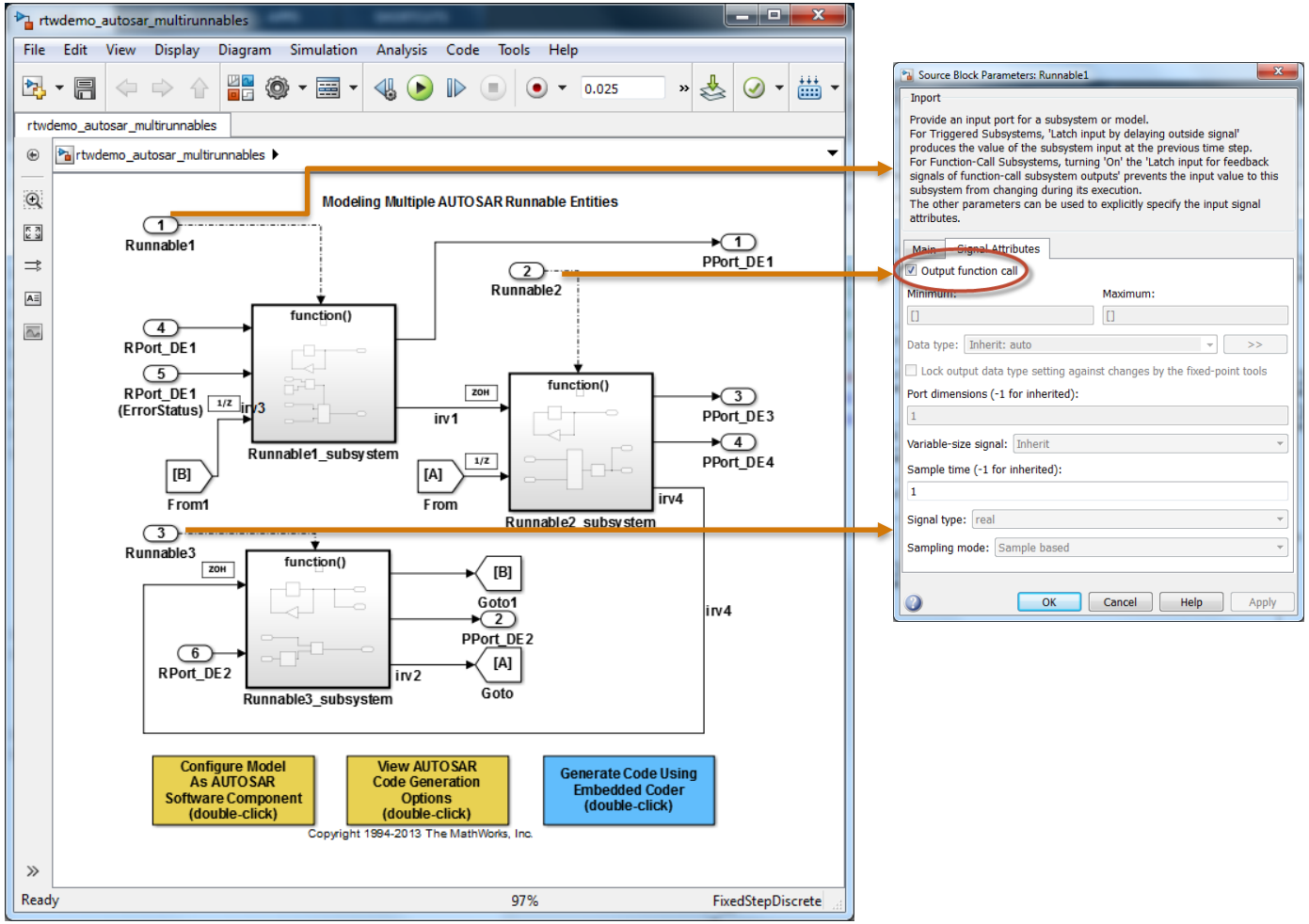

Configure AUTOSAR Interface: rtwdemo_autosar_counter
Simulink Mapping
rtwdemo_autosar_counter
Inports
Outports
Entry Point Functions
Filter Contents
Tip: To map a Simulink entry point function, select an AUTOSAR(AR) Runnable Help
Name
AR:Runnable
Step Function
Runnable_Step
Initialize Function
Runnable_Init
AUTOSAR Validation
Validation succeeded
Simulink-AUTOSAR Mapping
AUTOSAR Properties

生成代码



- Build模型
- 可同时生成 C & ARXML
- ARXML包含两个入口点
  - Step
  - Initialize
- 代码中使用RTE APIs 作 Sender Receiver ports

多入口点函数建模



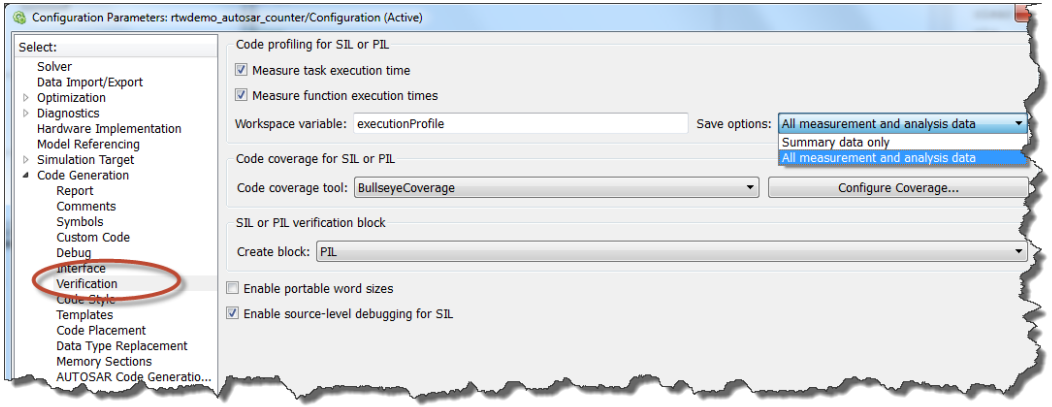
>> rtwdemo_autosar_multirunnables

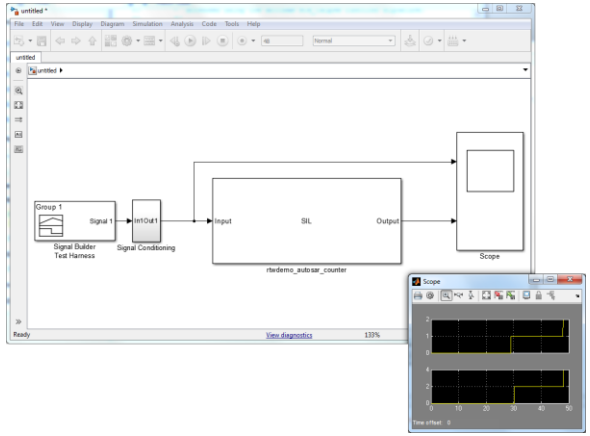
API支持采用脚本实现自动化

```
%% Setup AUTOSAR Configuration programmatically

model = 'rtwdemo_autosar_counter';

% Modify AUTOSAR Properties
autosarProps = autosar.api.getAUTOSARProperties(model);
set(autosarProps, 'Input', 'IsService', true);
set(autosarProps, 'XmlOptions', 'ArxmlFilePackaging','SingleFile');

% Modify Simulink Mapping to AUTOSAR
slMap = autosar.api.getSimulinkMapping(model);
mapInport(slMap, 'Input' , 'Input', 'Input' , 'ExplicitReceive');
mapOutport(slMap, 'Output' , 'Output', 'Output' , 'ExplicitSend');
```

处理器在环(PIL)验证

- 支持AUTOSAR的PIL验证
- 可以获取目标板上运行的代码和执行时间信息

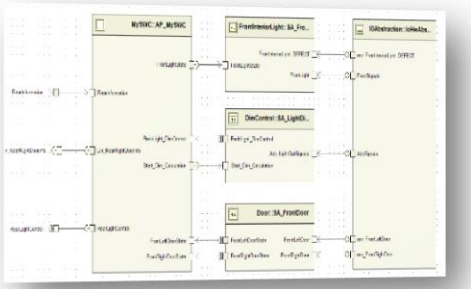


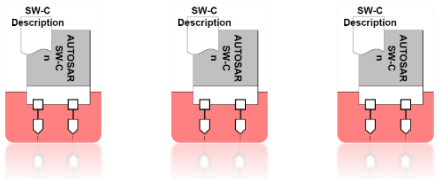


AUTOSAR工作流程

- 从软件组件描述文件起步

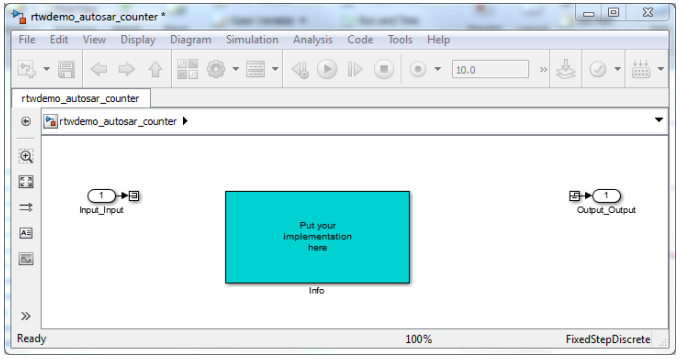
由上而下的工作流程



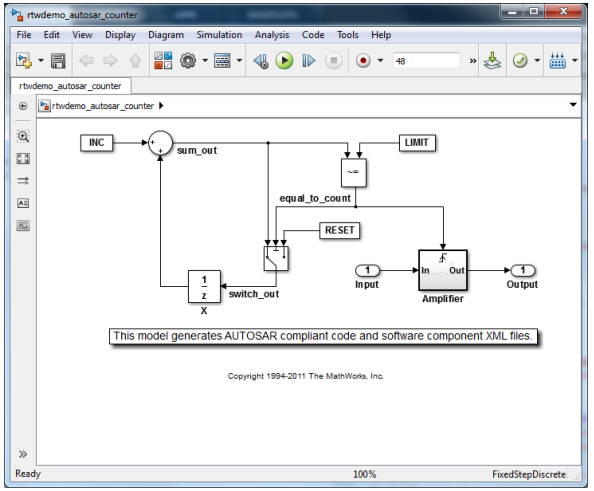


ARXML

导入生成新的Simulink模型

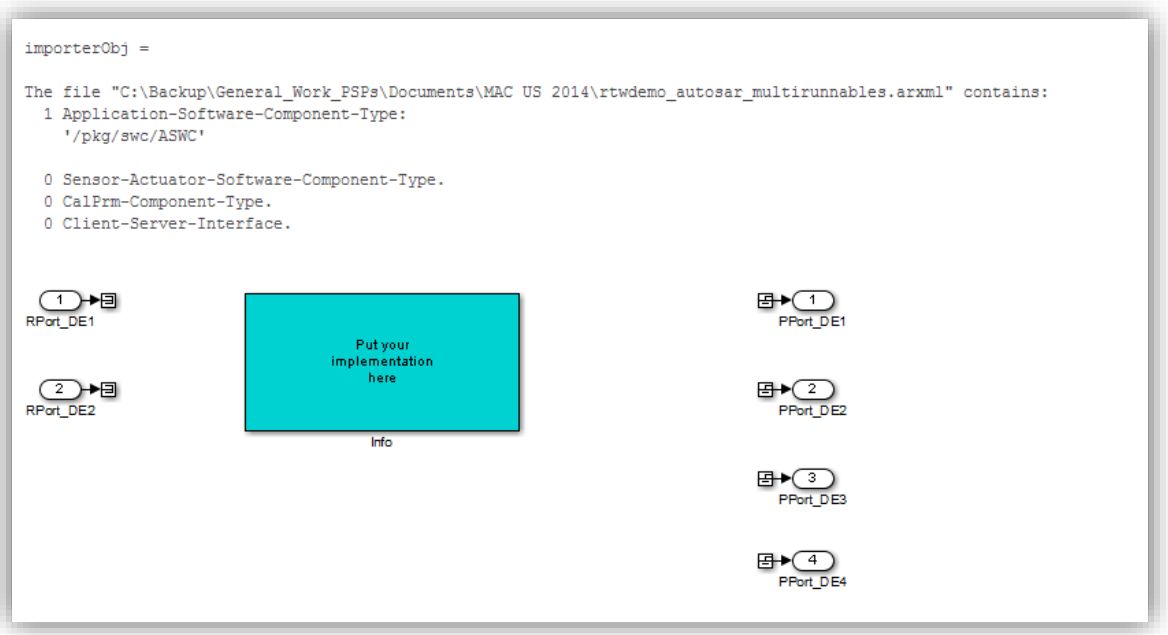


更新已有的Simulink模型



导入ARXML文件

```
%Import ARXML Files
importerObj = arxml.importer('rtwdemo_autosar_multirunnables.arxml')

%Create new model with interfaces
model = importerObj.createComponentAsModel('/pkg/swc/ASWC');
```



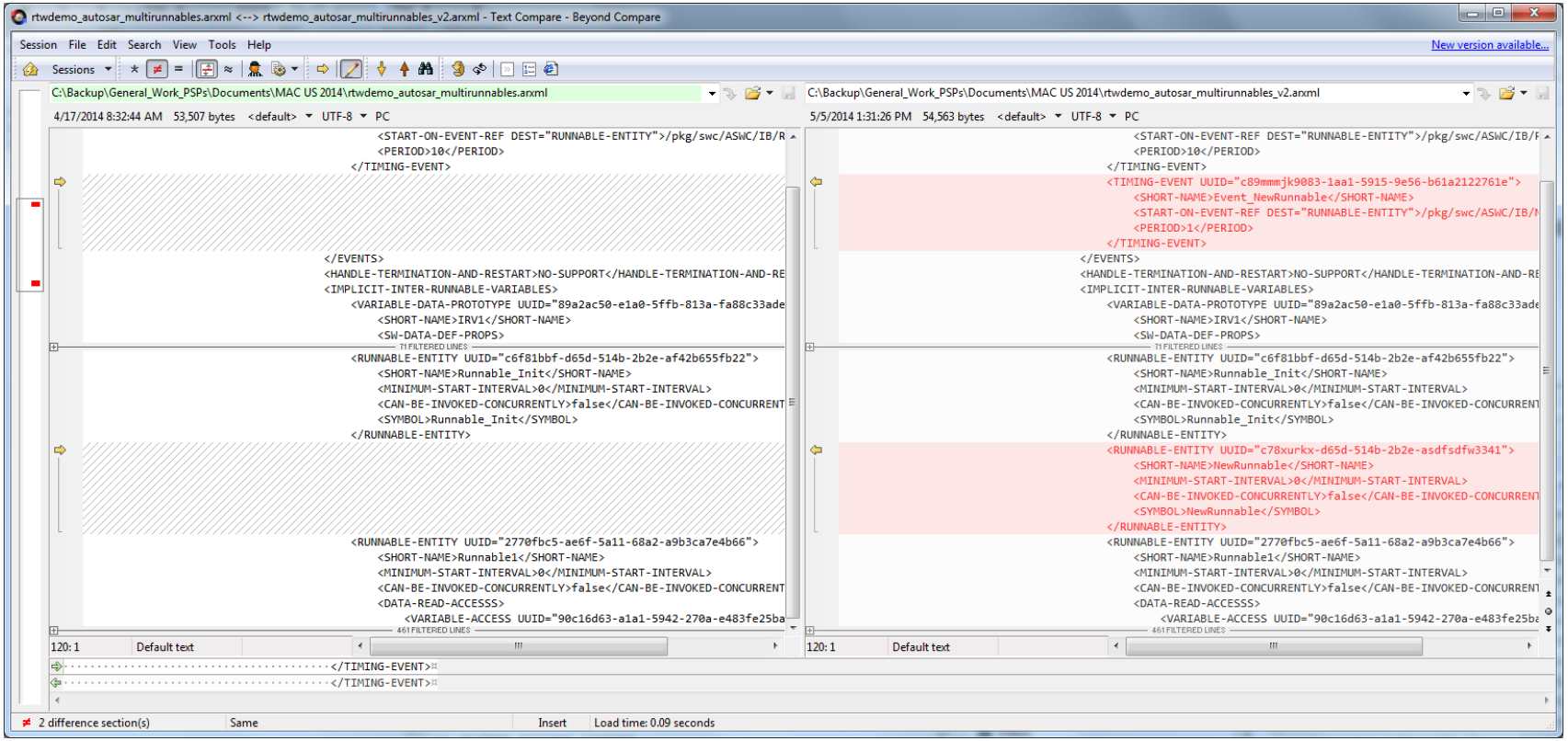
导入时引入Internal Behavior

```
%Import ARXML Files
importerObj =
arxml.importer('rtwdemo_autosar_multirunnables.arxml')

%Create new model with interfaces and internal behavior
model =
importerObj.createComponentAsModel('/pkg/swc/ASWC', ...
    'CreateInternalBehavior',true);
```

从ARXML更新已有的模型（1）

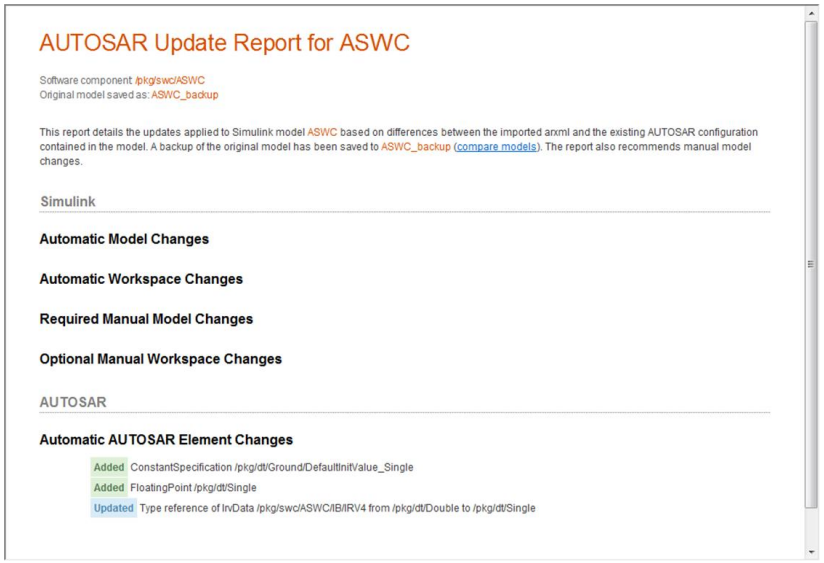
V1.arxml

Updated to V2.arxml



从ARXML更新已有的模型（2）

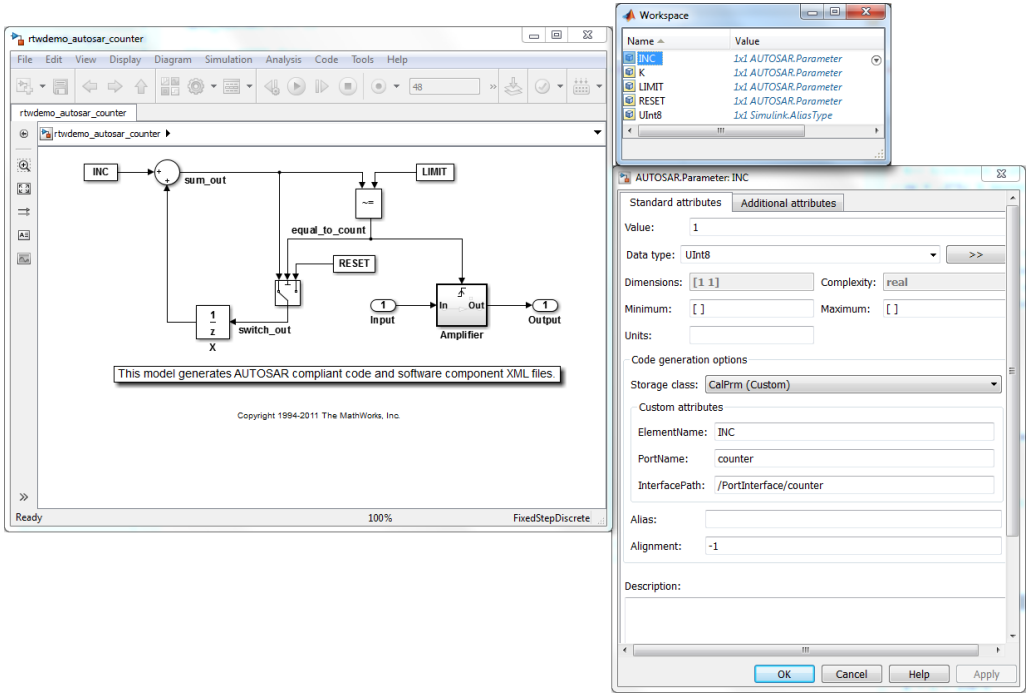
```
%Import ARXML Files
importerObj = arxml.importer('rtwdemo_autosar_multirunnables_v2.arxml')

%Update existing model
importerObj.updateModel('ASWC')
```



AUTOSAR中的标定和测量

标定量

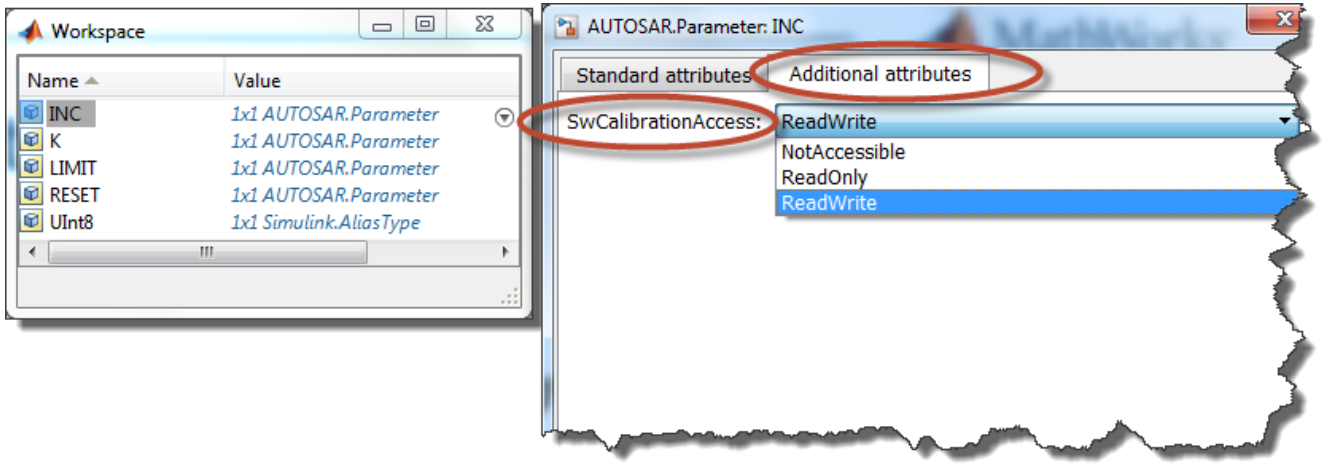
- AUTOSAR采用Simulink数据对象（SDO）标定
- 支持RTE API的两种定制数据类型 （CSC）
  - InternalCalPrm
  - CalPrm
- 无需RTE API直接访问
  - Constant-Memory
  - 带全局CSC的AUTOSAR4.Parameter




```
>> rtwdemo_autosar_counter
```

标定量的ASAP2工作流程

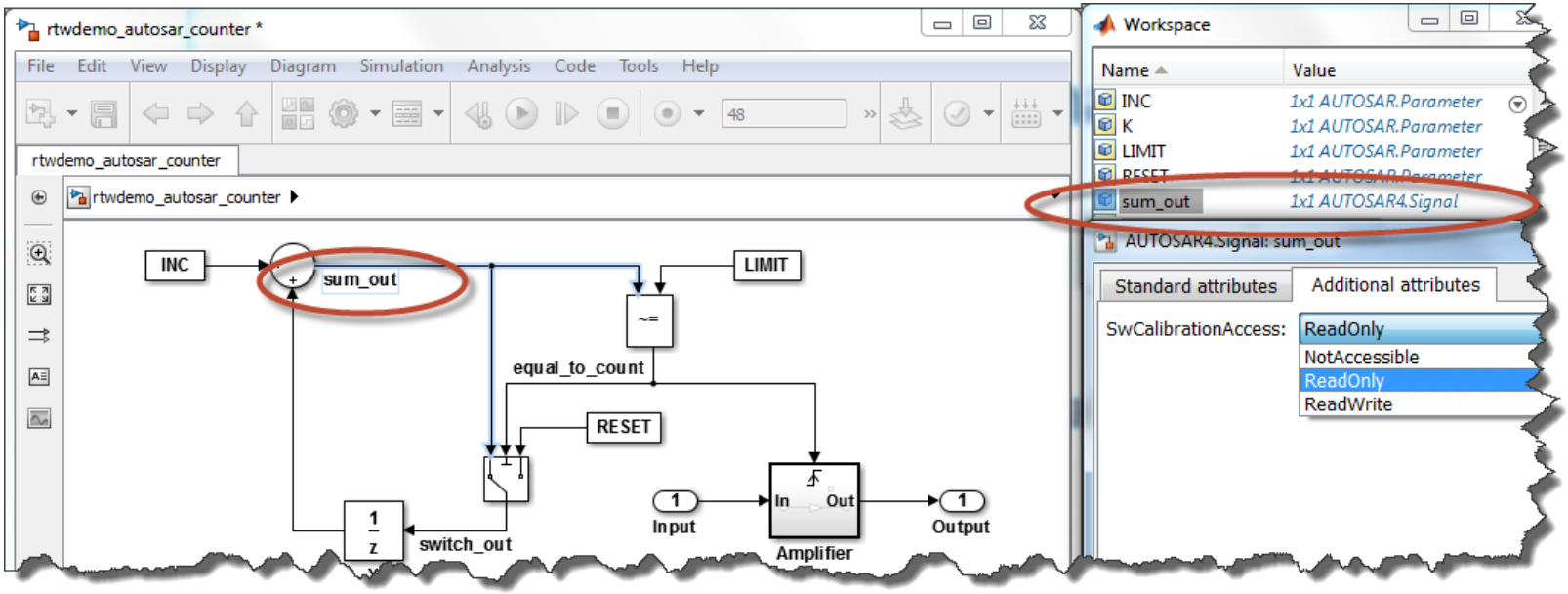
- 某些AAT可从ARXML生成ASAP2
- 控制AUTOSAR标定量的ASAP2属性



内部信号测量

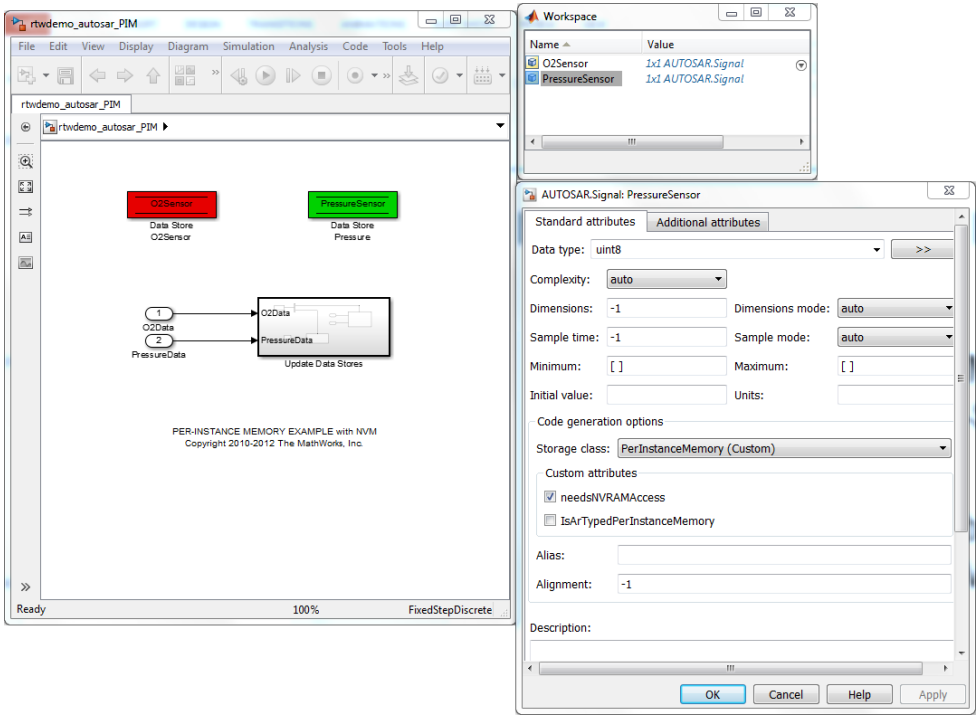
- 用户案例:
  - 采用第三方工具测量内部全局变量信号

- 使用AUTOSAR4.Signal数据对象
  - 对应AUTOSAR概念中的Static-Memory变量



NVM访问： Data-Store-Memory

- 支持 Implicit NVM access via PerInstanceMemory RAM mirror
- 对应到Data Store Memory
- 使用Per-Instance-Memory CSC for DSM
  - 需勾选needsNVRAMAccess
- 需要在AAT中作额外设定




```
>> rtwdemo_autosar_PIM
```

结论和方向

AUTOSAR支持：过去 vs.当前

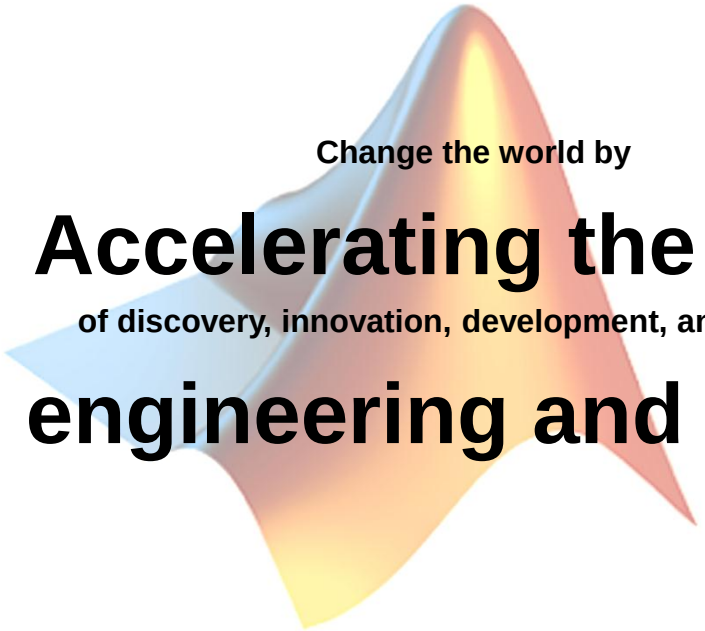
- ATPP: AUTOSAR Target Production Package
- 包含额外的AUTOSAR功能
- 开发独立于正式产品发布周期
- 快速支持AUTOSAR新功能

- R2014a
- 包括之前仅在ATPP才有的许多功能
- 推荐在新项目中使用
- 支持许多新的建模功能

开发方向

- 支持更多直接可用的AUTOSAR功能
- 在各新版本中快速增加新功能
- 与客户和其他AUTOSAR工具厂商紧密合作
- 积极参与AUTOSAR 标准工作组

MathWorks



Change the world by

Accelerating the pace

of discovery, innovation, development, and learning

in engineering and science

Thanks for Your Attention

Q&A