MATLAB EXPO

2024.06.11 | 그랜드 인터컨티넨탈 서울 파르나스

AUTOSAR Blockset을 활용한 ASW 개발 및 Mobilgene 통합

장동근 연구원, 현대자동차

MathWorks®

Contents

1. Introduction to the presenter

2. What is AUTOSAR Blockset?

3. MBD Workflow with a simple system
   - Different development process between Embedded Coder and AUTOSAR Blockset
   - Workflow with mobilgene and workaround solutions for some conflicts

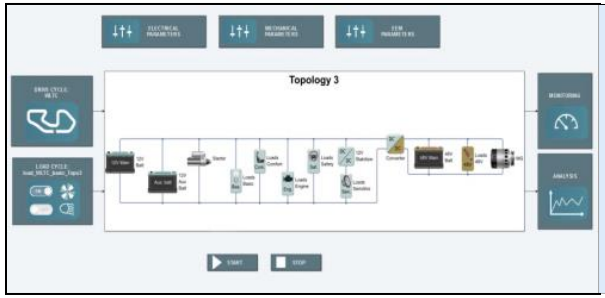
4. Useful modeling techniques

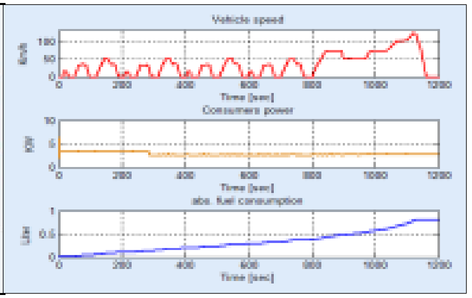
5. Our use case

Introduction to the presenter

2017~

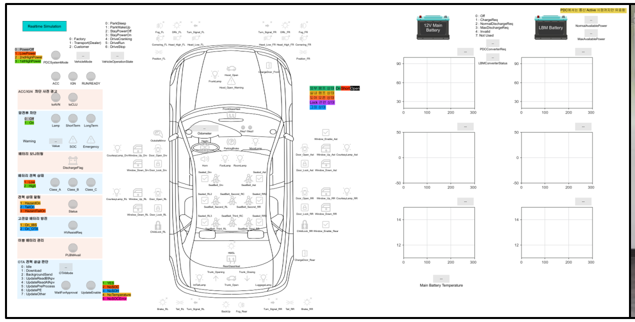
Development of simulation tools for vehicle power-net architecture design & analysis





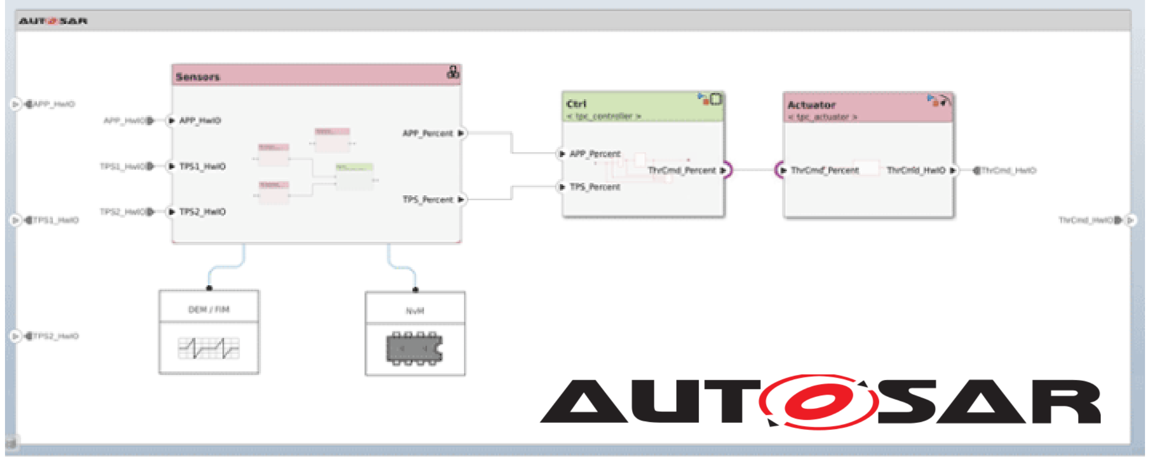
2021~

Verification of integrated MILs and HILs for power-net domain controller



2023~

AUTOSAR software development and test

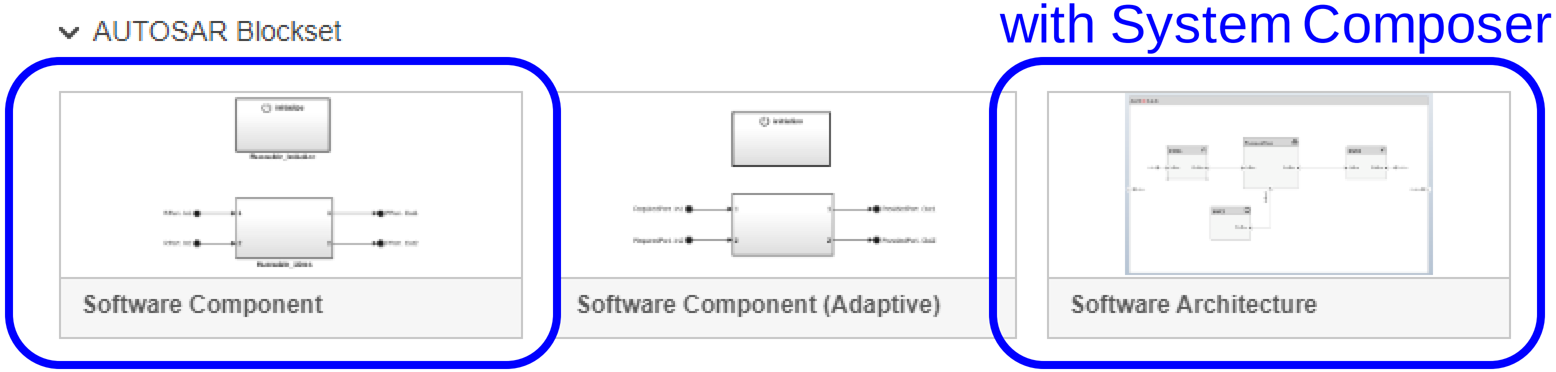


ASW Modeling skills using AUTOSAR Blockset

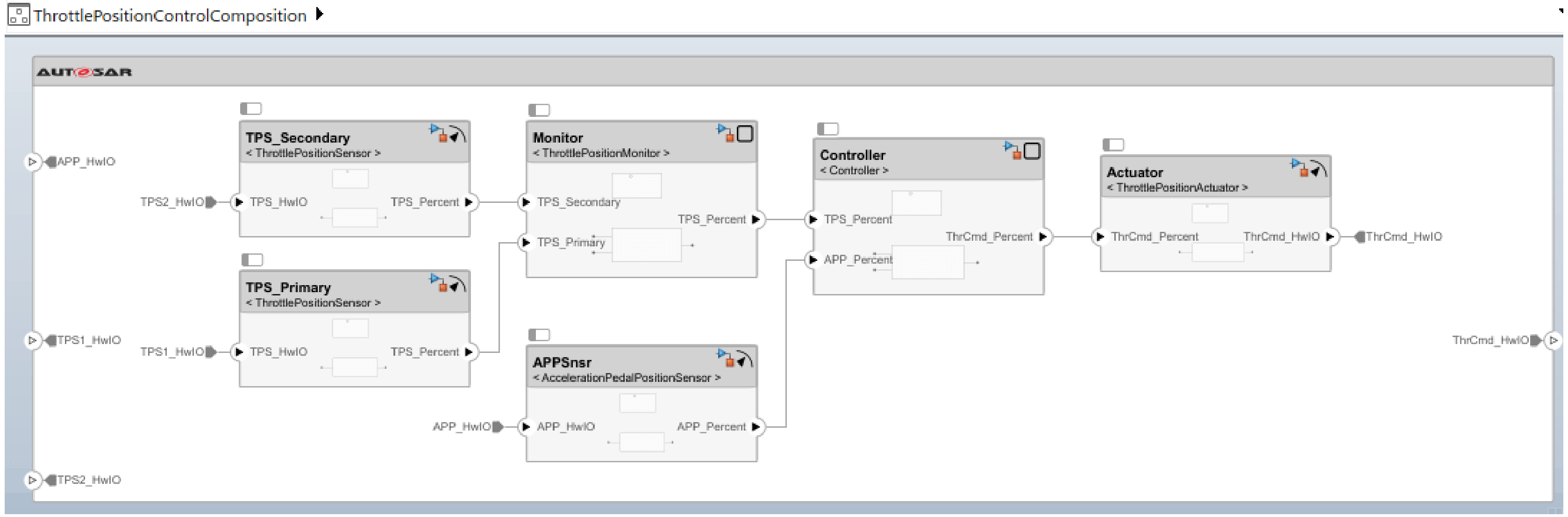
Limitation & workaround for Mobilgene SWP

What is AUTOSAR Blockset?

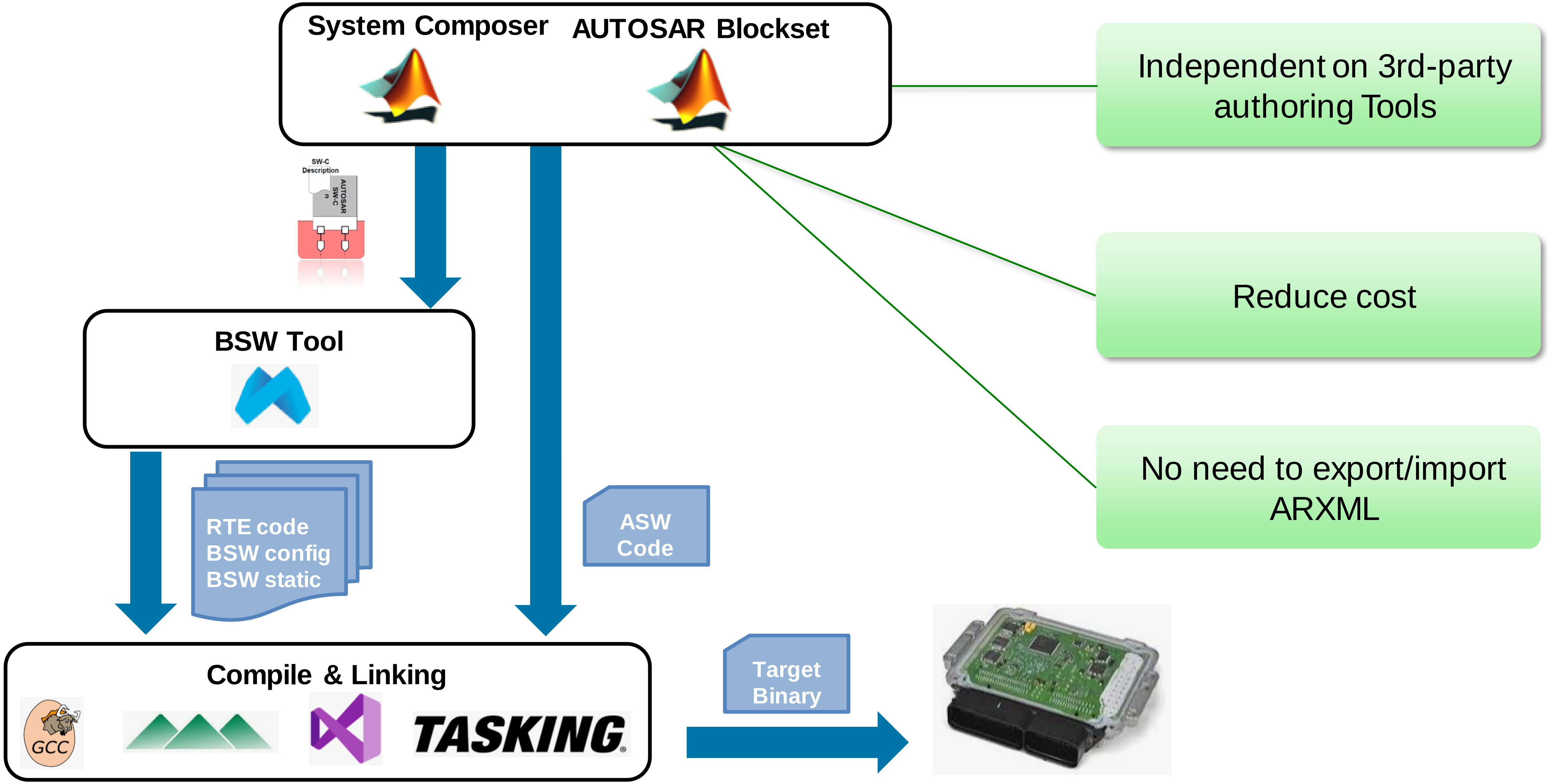
AUTOSAR Blockset provides apps and blocks for developing AUTOSAR Classic and Adaptive software using Simulink models.



with System Composer



MBD Workflow with System Composer and AUTOSAR Blockset



System Composer AUTOSAR Blockset

BSW Tool

RTE code
BSW config
BSW static

ASW
Code

Compile & Linking

Target
Binary

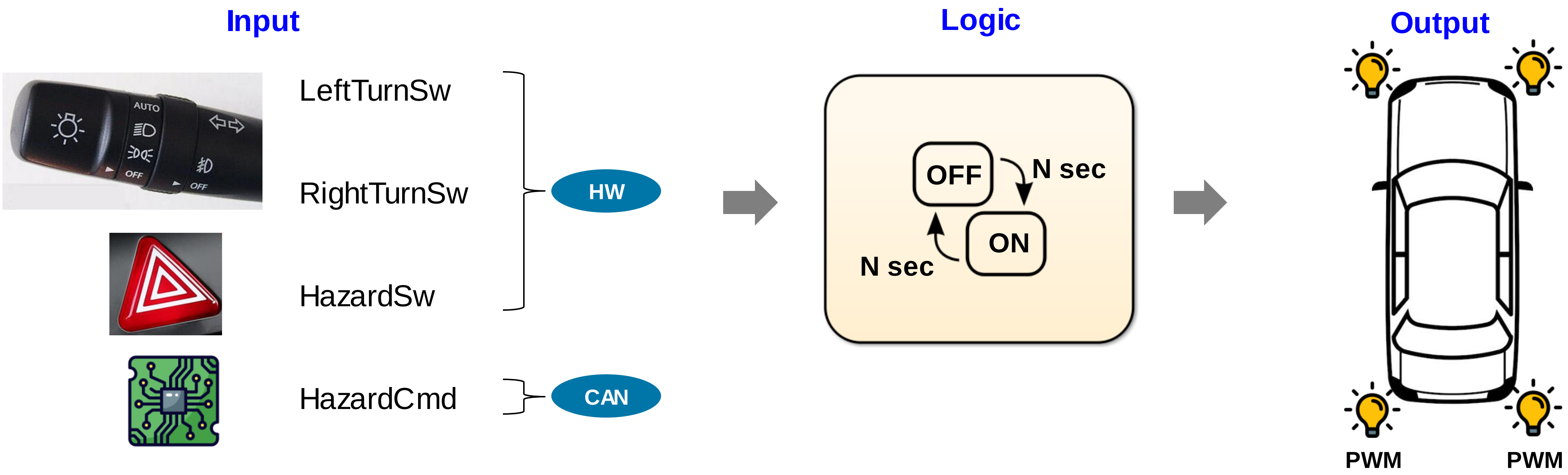
Independent on 3rd-party authoring Tools

Reduce cost

No need to export/import ARXML

Define simple system requirements of vehicle turn signal lamps

- It has 4 inputs(2 Analog + 1 Digital + 1 Comm.) and 4 outputs for each lamp
- When the lamps are turned on, they blink every N seconds. (N is from NvM)
- Rear lamps operate using PWM with a duty cycle of 80%



Input

LeftTurnSw

RightTurnSw

HazardSw

HW

HazardCmd

CAN

Logic

OFF

N sec

ON

N sec

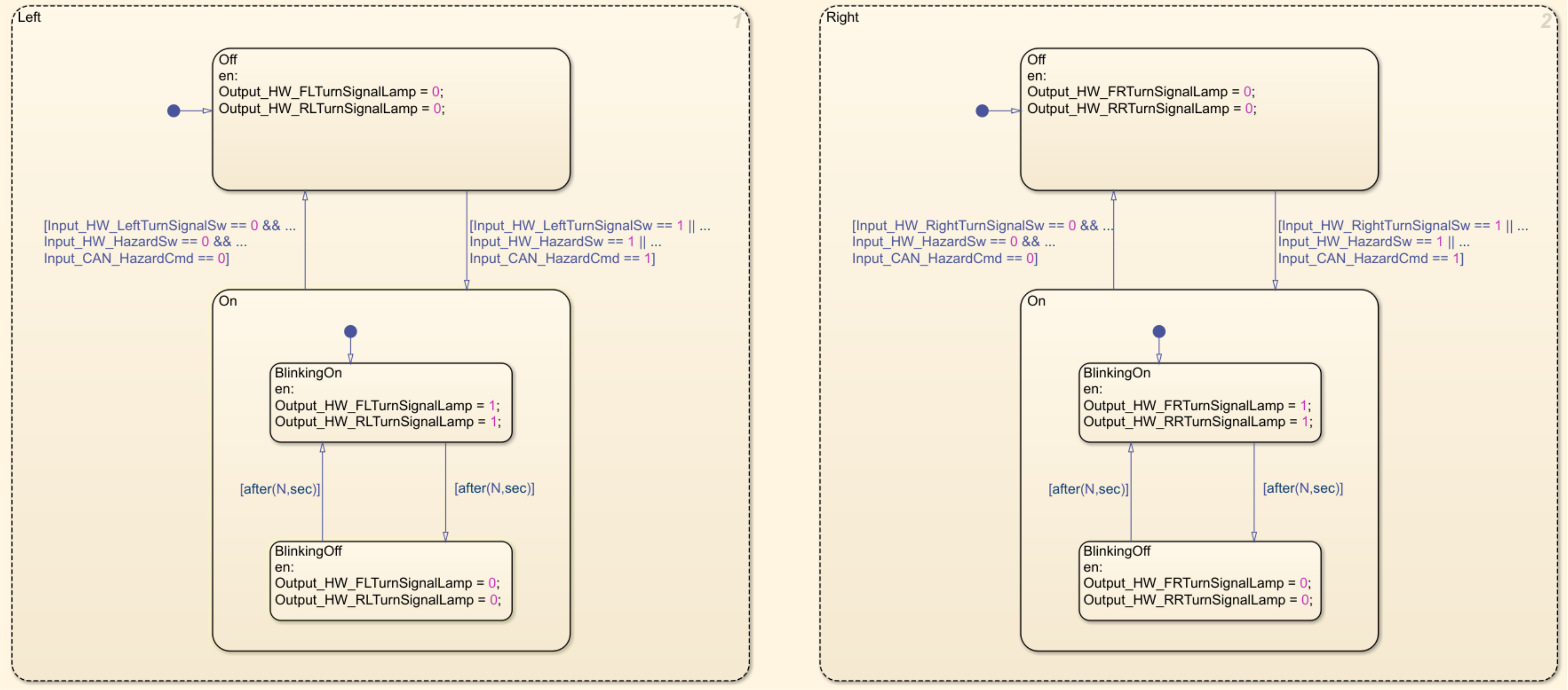
Output

PWM

PWM

Development of the logic model

- Lamp blinking logic model using Simulink Chart

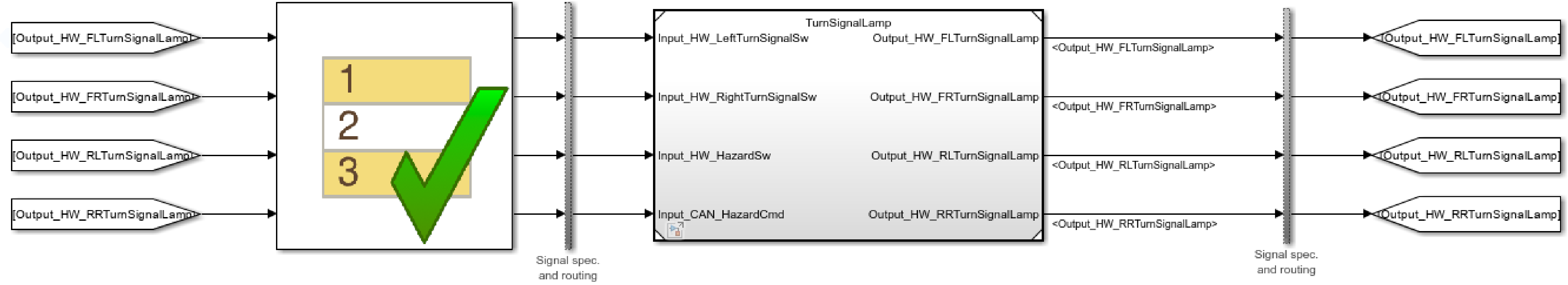


```
Left

Off
en:
Output_HW_FLTurnSignalLamp = 0;
Output_HW_RLTurnSignalLamp = 0;

[Input_HW_LeftTurnSignalSw == 0 && ...
Input_HW_HazardSw == 0 && ...
Input_CAN_HazardCmd == 0]

[Input_HW_LeftTurnSignalSw == 1 || ...
Input_HW_HazardSw == 1 || ...
Input_CAN_HazardCmd == 1]

On

BlinkingOn
en:
Output_HW_FLTurnSignalLamp = 1;
Output_HW_RLTurnSignalLamp = 1;

[after(N,sec)]

[after(N,sec)]

BlinkingOff
en:
Output_HW_FLTurnSignalLamp = 0;
Output_HW_RLTurnSignalLamp = 0;
```

```
Right

Off
en:
Output_HW_FRTurnSignalLamp = 0;
Output_HW_RRTurnSignalLamp = 0;

[Input_HW_RightTurnSignalSw == 0 && ...
Input_HW_HazardSw == 0 && ...
Input_CAN_HazardCmd == 0]

[Input_HW_RightTurnSignalSw == 1 || ...
Input_HW_HazardSw == 1 || ...
Input_CAN_HazardCmd == 1]

On

BlinkingOn
en:
Output_HW_FRTurnSignalLamp = 1;
Output_HW_RRTurnSignalLamp = 1;

[after(N,sec)]

[after(N,sec)]

BlinkingOff
en:
Output_HW_FRTurnSignalLamp = 0;
Output_HW_RRTurnSignalLamp = 0;
```

Verification of the logic model

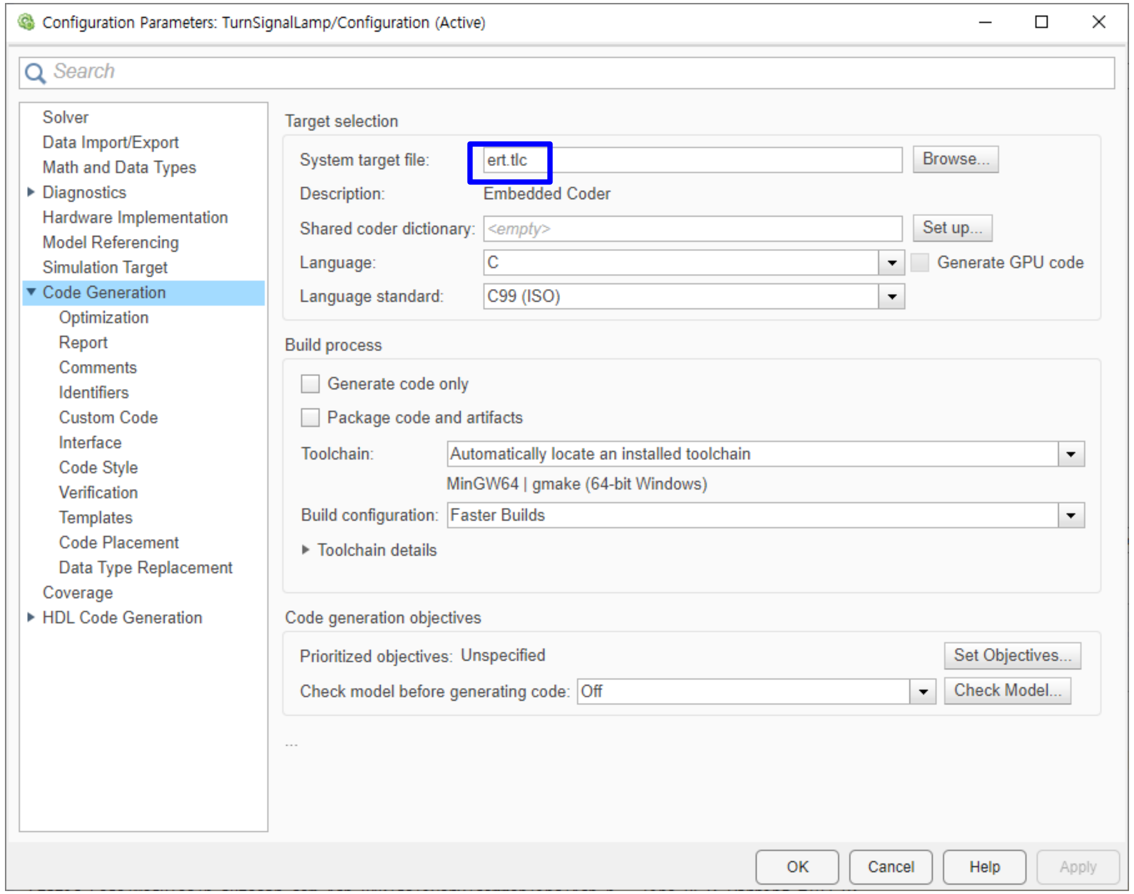
- Create a test harness and verify if the model satisfies the functional requirement



[Output_HW_FLTurnSignalLamp>
[Output_HW_FRTurnSignalLamp>
[Output_HW_RLTurnSignalLamp>
[Output_HW_RRTurnSignalLamp>
1
2
3
Signal spec. and routing
TurnSignalLamp
Input_HW_LeftTurnSignalSw
Input_HW_RightTurnSignalSw
Input_HW_HazardSw
Input_CAN_HazardCmd
Output_HW_FLTurnSignalLamp
Output_HW_FRTurnSignalLamp
Output_HW_RLTurnSignalLamp
Output_HW_RRTurnSignalLamp
<Output_HW_FLTurnSignalLamp>
<Output_HW_FRTurnSignalLamp>
<Output_HW_RLTurnSignalLamp>
<Output_HW_RRTurnSignalLamp>
Signal spec. and routing
<Output_HW_FLTurnSignalLamp]
<Output_HW_FRTurnSignalLamp]
<Output_HW_RLTurnSignalLamp]
<Output_HW_RRTurnSignalLamp]

General workflow using Embedded Coder

- Set system target file of code generation to ert.tlc and generate code/header



Configuration Parameters: TurnSignalLamp/Configuration (Active)

Target selection

System target file: ert.tlc

Description: Embedded Coder

Shared coder dictionary: <empty>

Language: C

□ Generate GPU code

Language standard: C99 (ISO)

Build process

□ Generate code only

□ Package code and artifacts

Toolchain: Automatically locate an installed toolchain

MinGW64 | gmake (64-bit Windows)

Build configuration: Faster Builds

Toolchain details

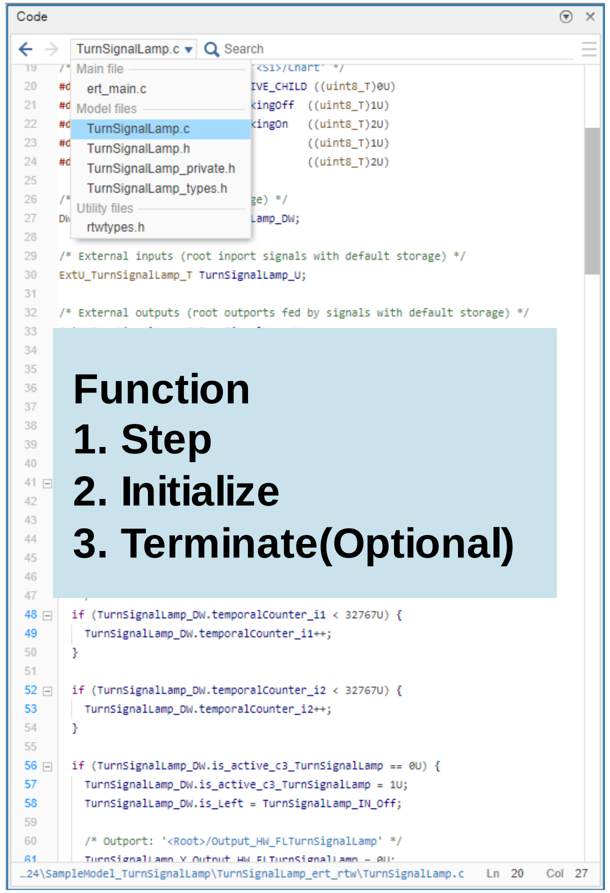
Code generation objectives

Prioritized objectives: Unspecified

Check model before generating code: Off

...

Build



Code

TurnSignalLamp.c

```
/* External inputs (root inport signals with default storage) */
ExtU_TurnSignalLamp_T TurnSignalLamp_U;

/* External outputs (root outports fed by signals with default storage) */
```

Function

1. Step

2. Initialize

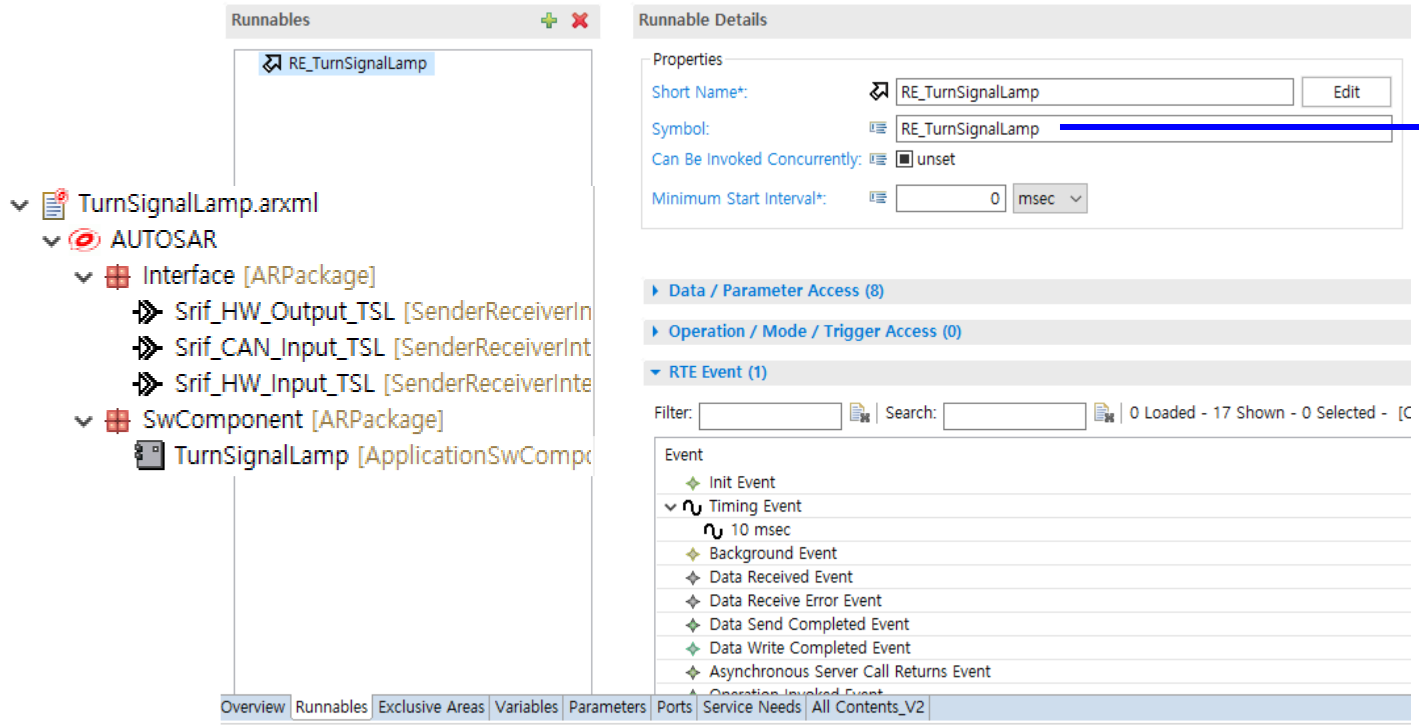
3. Terminate(Optional)

```
  if (TurnSignalLamp_DW.temporalCounter_i1 < 32767U) {
    TurnSignalLamp_DW.temporalCounter_i1++;
  }

  if (TurnSignalLamp_DW.temporalCounter_i2 < 32767U) {
    TurnSignalLamp_DW.temporalCounter_i2++;
  }

  if (TurnSignalLamp_DW.is_active_c3_TurnSignalLamp == 0U) {
    TurnSignalLamp_DW.is_active_c3_TurnSignalLamp = 1U;
    TurnSignalLamp_DW.is_Left = TurnSignalLamp_IN_Off;

    /* Outport: '<Root>/Output_HW_FLTurnSignalLamp' */
```

...24\SampleModel_TurnSignalLamp\TurnSignalLamp_ert_rtw\TurnSignalLamp.c Ln 20 Col 27

Workflow using Embedded Coder

- Create SWC and configure AUTOSAR properties such as Port, Interface, Runnable etc.
- Use RTE APIs for data read/write and call the generated functions



Runnables

RE_TurnSignalLamp

TurnSignalLamp.arxml
- AUTOSAR
  - Interface [ARPackage]
    - Srif_HW_Output_TSL [SenderReceiverIn
    - Srif_CAN_Input_TSL [SenderReceiverInt
    - Srif_HW_Input_TSL [SenderReceiverInte
  - SwComponent [ARPackage]
    - TurnSignalLamp [ApplicationSwComp

Runnable Details

Properties

Short Name*: RE_TurnSignalLamp Edit

Symbol: RE_TurnSignalLamp

Can Be Invoked Concurrently: ■ unset

Minimum Start Interval*: 0 msec

- Data / Parameter Access (8)
- Operation / Mode / Trigger Access (0)
- RTE Event (1)

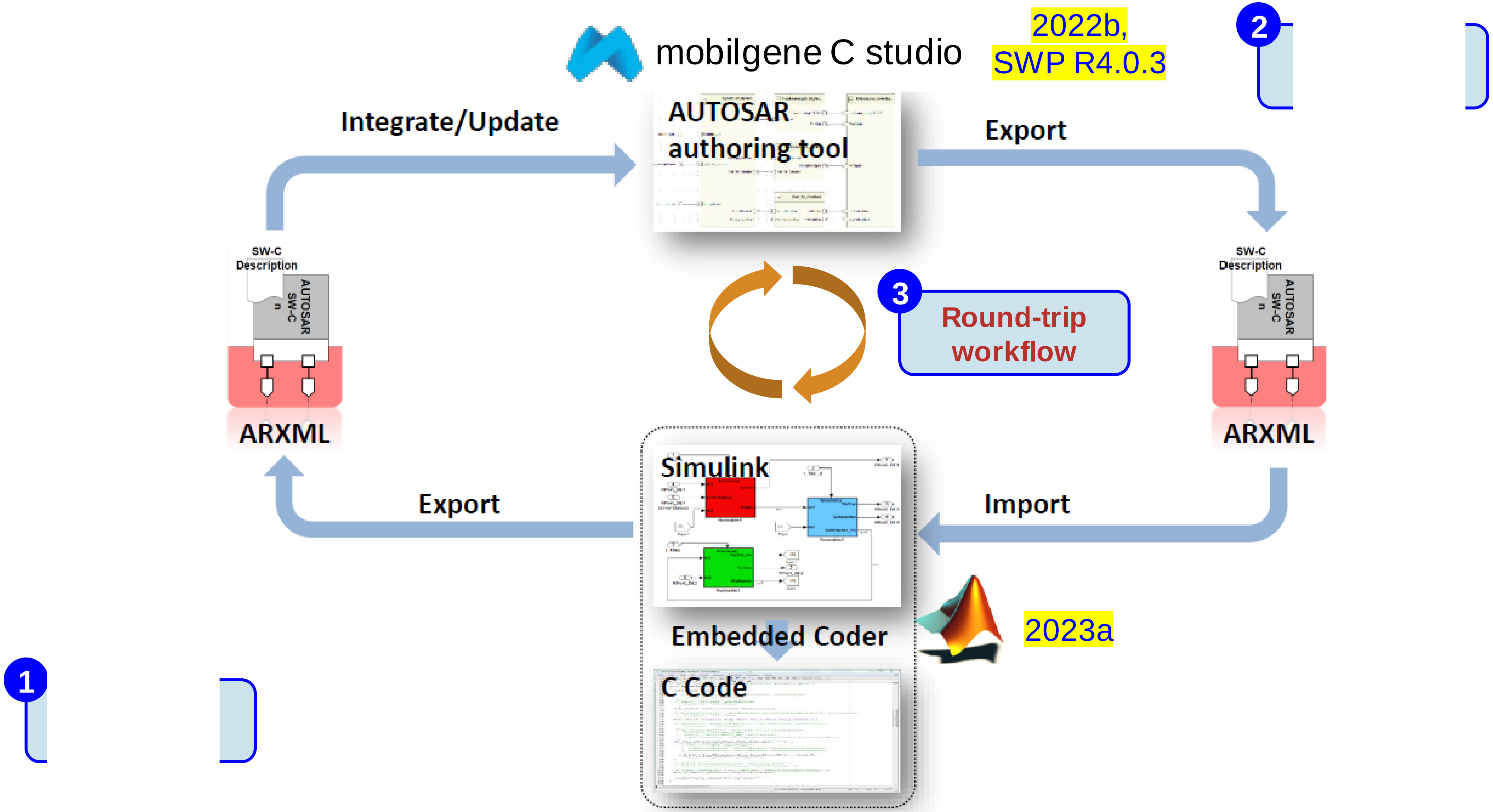
Filter: Search: 0 Loaded - 17 Shown - 0 Selected - [C

Event
- Init Event
- Timing Event
  - 10 msec
- Background Event
- Data Received Event
- Data Receive Error Event
- Data Send Completed Event
- Data Write Completed Event
- Asynchronous Server Call Returns Event

Overview | Runnables | Exclusive Areas | Variables | Parameters | Ports | Service Needs | All Contents_V2

```
#include "Rte_TurnSignalLamp.h"
#include "TurnSignalLamp.h"
void RE_TurnSignalLamp()
{
  Rte_Read_***_***_***(&A);
  Rte_Read_***_***_***(&B);

  TSL_Input. LeftTurnSignalSw = A. LeftTurnSignalSw;
  …

  TurnSignalLamp_step();

  …

  Rte_Write_***_***_***(&A);
  Rte_Write_***_***_***(&B);
}
```

Workflow of AUTOSAR MBD



mobilgene C studio

2022b,
SWP R4.0.3

2

Integrate/Update

AUTOSAR authoring tool

Export

3

Round-trip workflow

Export

Simulink

Import

Embedded Coder

C Code

2023a

1

Bottom-up workflow with an existing model

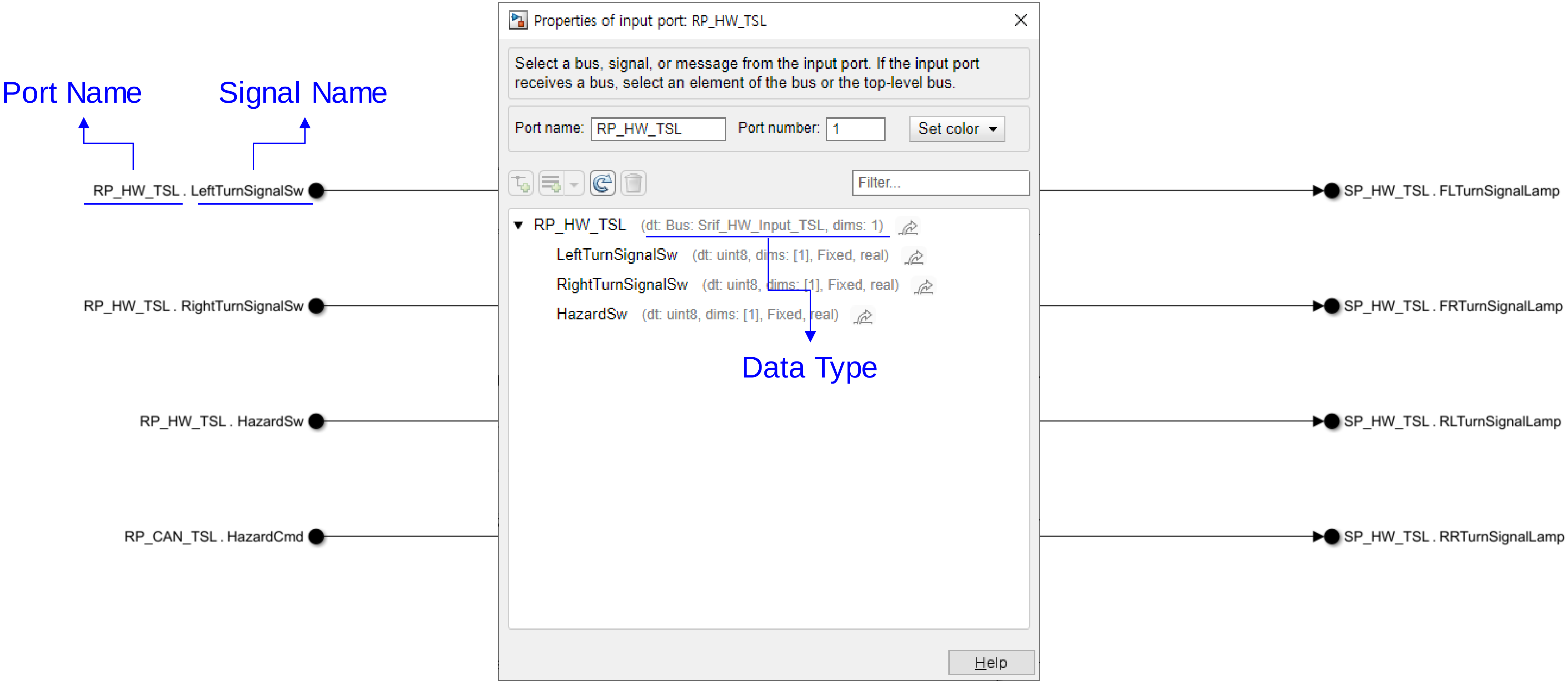
- Relationships between Simulink and AUTOSAR properties

Simulink	AUTOSAR Properties
In Bus Element Block	Receiver Port
Out Bus Element Block	Sender Port
Data Type (Simulink Bus defined in .sldd)	Sender Receiver Interface
Port Name	Port
Signal Name	Data Element

Properties of In/Out Bus Element Block

Bottom-up workflow with an existing model

- Convert In/Outport blocks to In/Out Bus Element blocks.



Port Name
Signal Name
RP_HW_TSL . LeftTurnSignalSw
RP_HW_TSL . RightTurnSignalSw
RP_HW_TSL . HazardSw
RP_CAN_TSL . HazardCmd
SP_HW_TSL . FLTurnSignalLamp
SP_HW_TSL . FRTurnSignalLamp
SP_HW_TSL . RLTurnSignalLamp
SP_HW_TSL . RRTurnSignalLamp
Data Type
Filter...
Set color
Help

Properties of input port: RP_HW_TSL

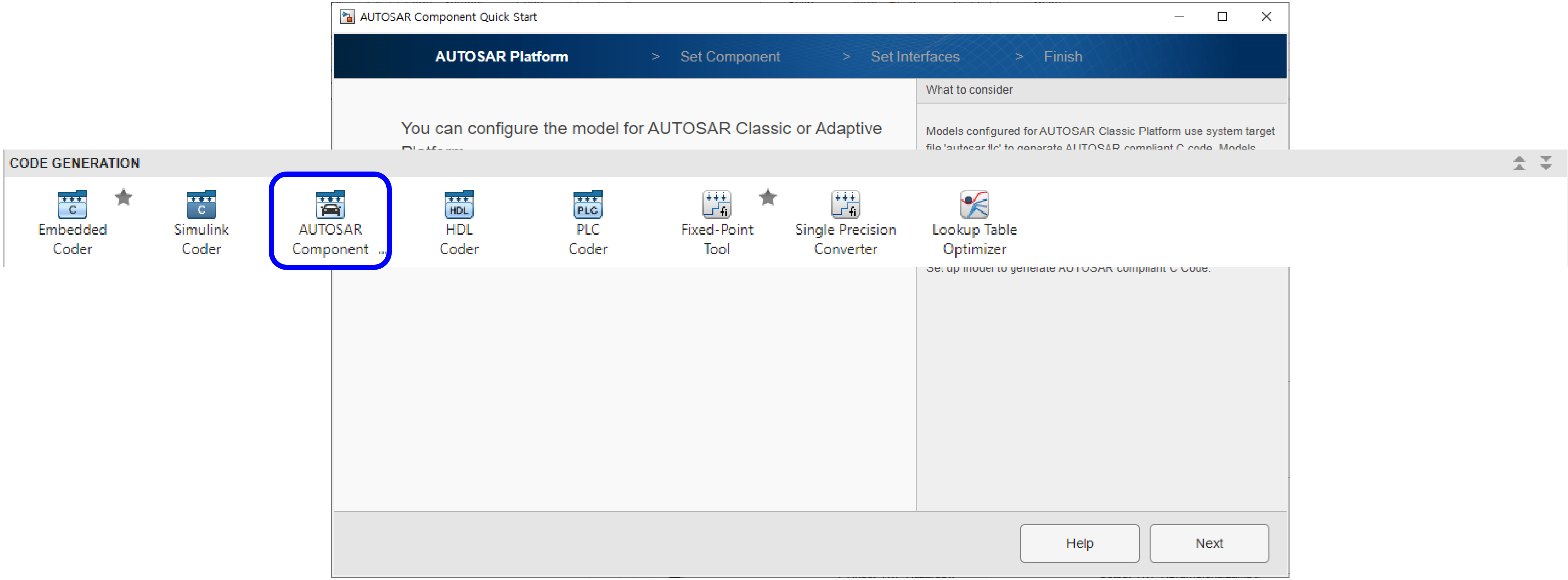
Select a bus, signal, or message from the input port. If the input port receives a bus, select an element of the bus or the top-level bus.

Port name: RP_HW_TSL Port number: 1

```
▼ RP_HW_TSL   (dt: Bus: Srif_HW_Input_TSL, dims: 1)
    LeftTurnSignalSw   (dt: uint8, dims: [1], Fixed, real)
    RightTurnSignalSw   (dt: uint8, dims: [1], Fixed, real)
    HazardSw   (dt: uint8, dims: [1], Fixed, real)
```

Bottom-up workflow with an existing model

- Apps > AUTOSAR Component Designer



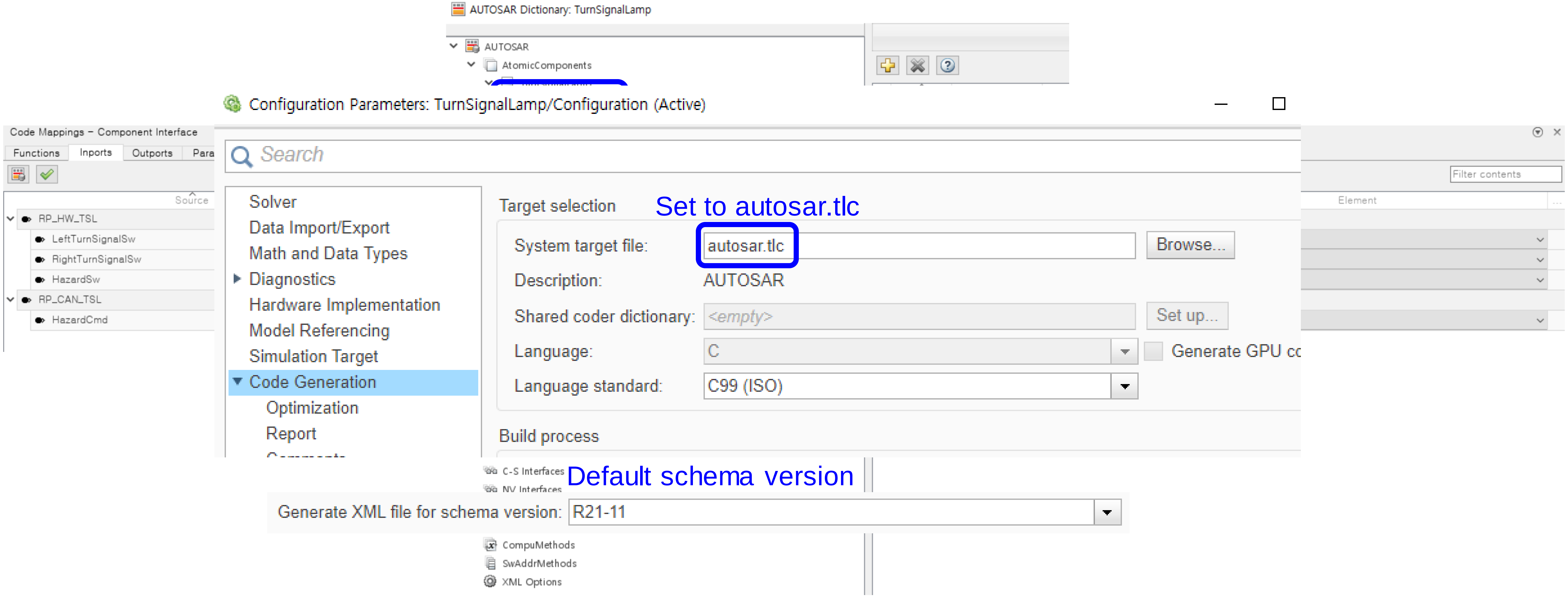
You can configure the model for AUTOSAR Classic or Adaptive Platform

Models configured for AUTOSAR Classic Platform use system target file 'autosar.tlc' to generate AUTOSAR compliant C code. Models

Set up model to generate AUTOSAR compliant C Code.

Bottom-up workflow with an existing model

- AUTOSAR properties will be automatically created and mapped



AUTOSAR Dictionary: TurnSignalLamp

Configuration Parameters: TurnSignalLamp/Configuration (Active)

Target selection

Set to autosar.tlc

System target file: autosar.tlc

Description: AUTOSAR

Shared coder dictionary: <empty>

Language: C

Language standard: C99 (ISO)

□ Generate GPU co

Build process

Default schema version

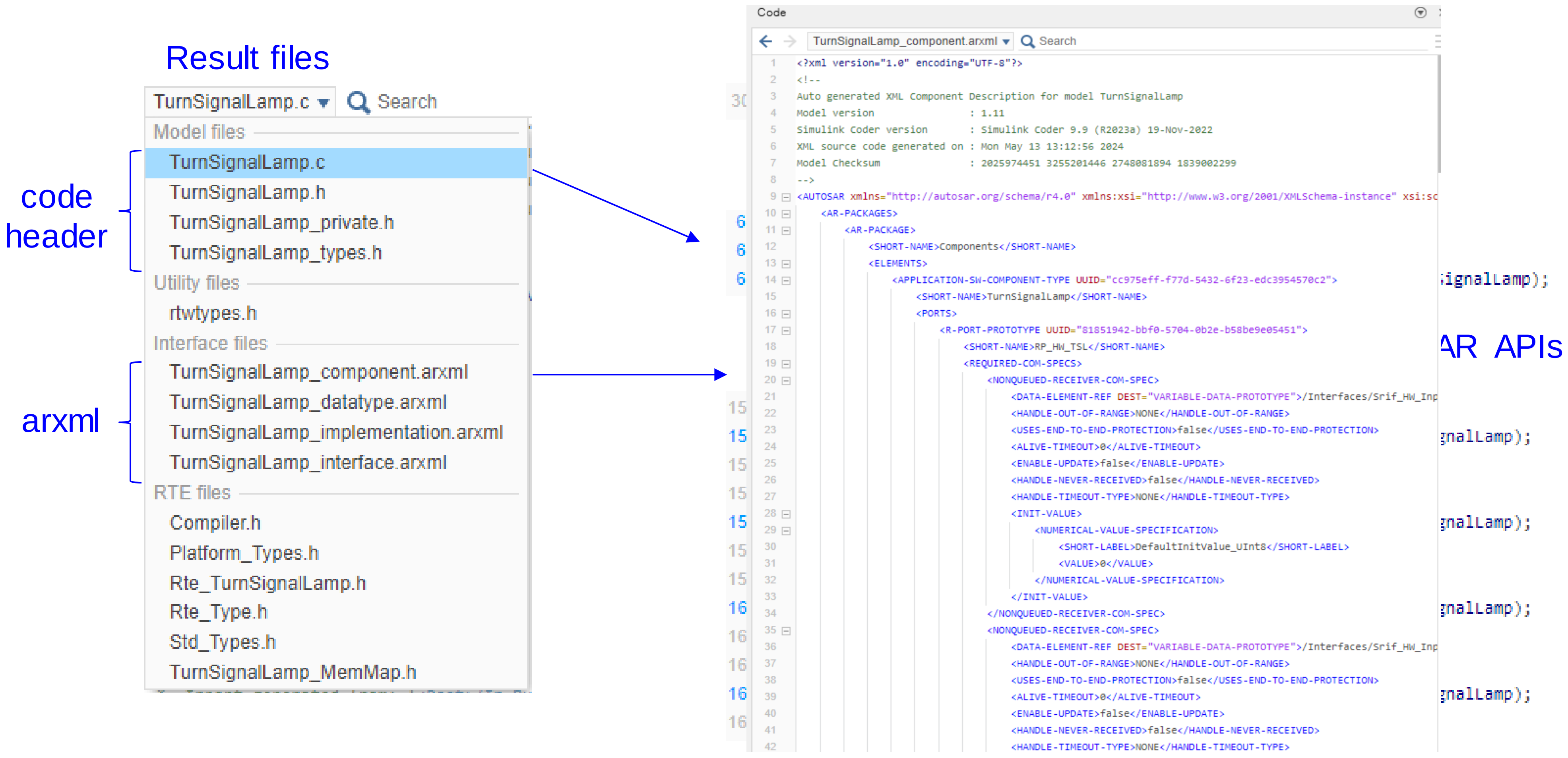
Generate XML file for schema version: R21-11

Code Mappings - Component Interface

- RP_HW_TSL
  - LeftTurnSignalSw
  - RightTurnSignalSw
  - HazardSw
- RP_CAN_TSL
  - HazardCmd

Bottom-up workflow with an existing model

- Build to generate C code and extract arxml files



TurnSignalLamp.c ▾ Search

Model files
- TurnSignalLamp.c
- TurnSignalLamp.h
- TurnSignalLamp_private.h
- TurnSignalLamp_types.h

Utility files
- rtwtypes.h

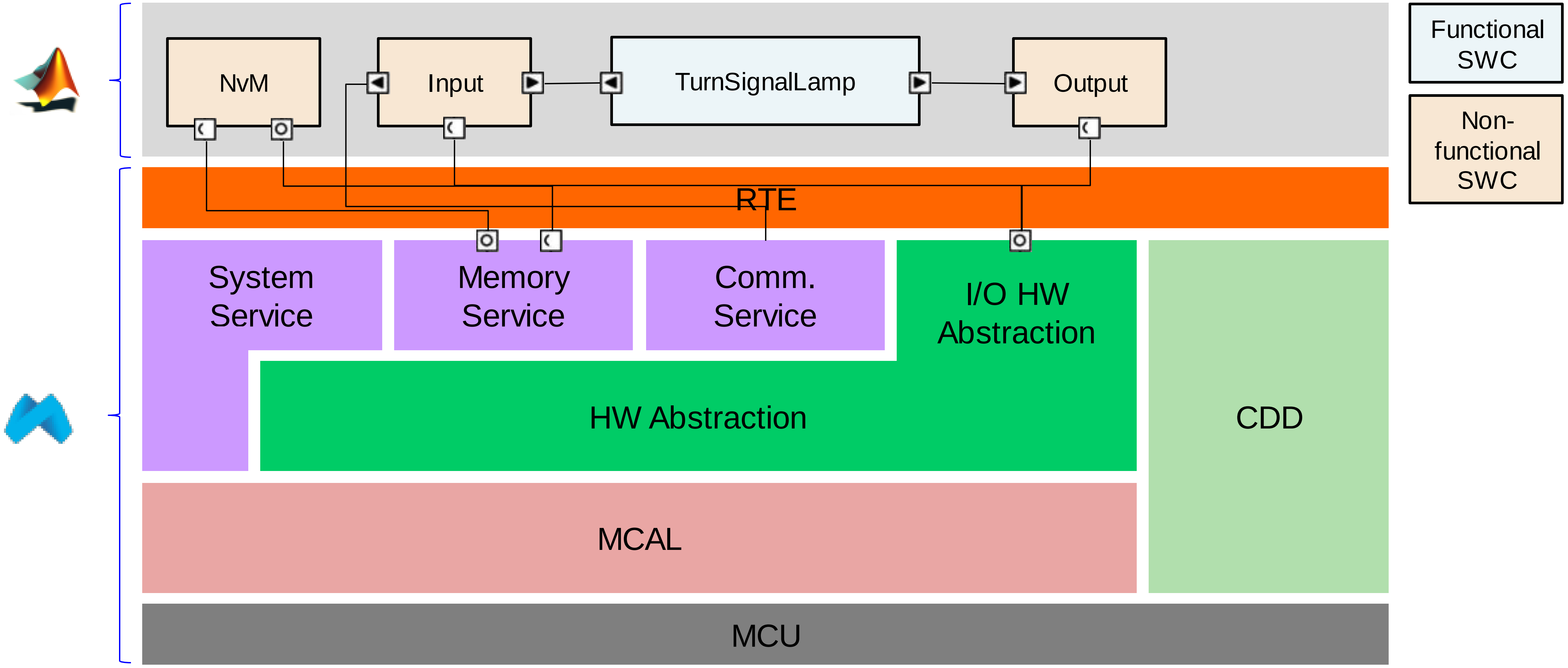
Interface files
- TurnSignalLamp_component.arxml
- TurnSignalLamp_datatype.arxml
- TurnSignalLamp_implementation.arxml
- TurnSignalLamp_interface.arxml

RTE files
- Compiler.h
- Platform_Types.h
- Rte_TurnSignalLamp.h
- Rte_Type.h
- Std_Types.h
- TurnSignalLamp_MemMap.h

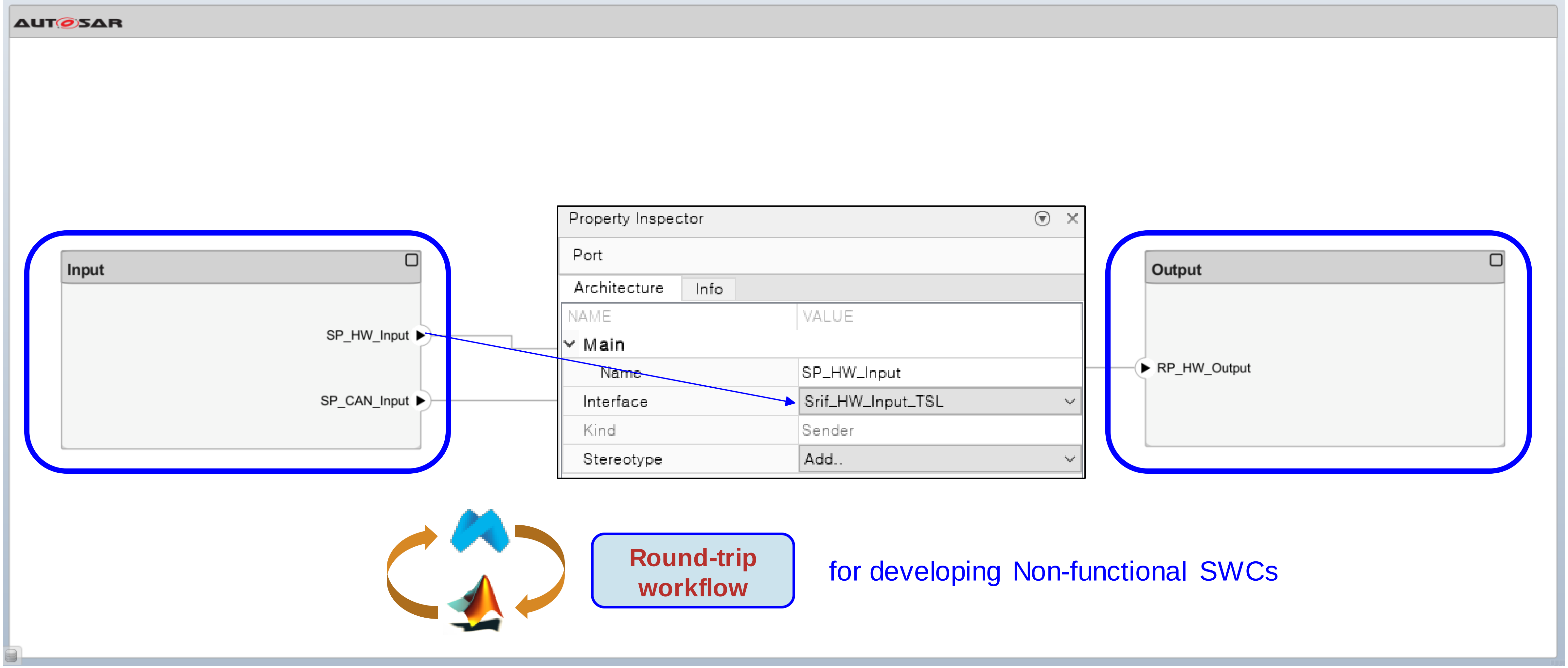
TurnSignalLamp_component.arxml ▾ Search

```
<?xml version="1.0" encoding="UTF-8"?>
<!--
Auto generated XML Component Description for model TurnSignalLamp
Model version                : 1.11
Simulink Coder version       : Simulink Coder 9.9 (R2023a) 19-Nov-2022
XML source code generated on : Mon May 13 13:12:56 2024
Model Checksum               : 2025974451 3255201446 2748081894 1839002299
-->
<AUTOSAR xmlns="http://autosar.org/schema/r4.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:sc
    <AR-PACKAGES>
        <AR-PACKAGE>
            <SHORT-NAME>Components</SHORT-NAME>
            <ELEMENTS>
                <APPLICATION-SW-COMPONENT-TYPE UUID="cc975eff-f77d-5432-6f23-edc3954570c2">
                    <SHORT-NAME>TurnSignalLamp</SHORT-NAME>
                    <PORTS>
                        <R-PORT-PROTOTYPE UUID="81851942-bbf0-5704-0b2e-b58be9e05451">
                            <SHORT-NAME>RP_HW_TSL</SHORT-NAME>
                            <REQUIRED-COM-SPECS>
                                <NONQUEUED-RECEIVER-COM-SPEC>
                                    <DATA-ELEMENT-REF DEST="VARIABLE-DATA-PROTOTYPE">/Interfaces/Srif_HW_Inp
                                    <HANDLE-OUT-OF-RANGE>NONE</HANDLE-OUT-OF-RANGE>
                                    <USES-END-TO-END-PROTECTION>false</USES-END-TO-END-PROTECTION>
                                    <ALIVE-TIMEOUT>0</ALIVE-TIMEOUT>
                                    <ENABLE-UPDATE>false</ENABLE-UPDATE>
                                    <HANDLE-NEVER-RECEIVED>false</HANDLE-NEVER-RECEIVED>
                                    <HANDLE-TIMEOUT-TYPE>NONE</HANDLE-TIMEOUT-TYPE>
                                    <INIT-VALUE>
                                        <NUMERICAL-VALUE-SPECIFICATION>
                                            <SHORT-LABEL>DefaultInitValue_UInt8</SHORT-LABEL>
                                            <VALUE>0</VALUE>
                                        </NUMERICAL-VALUE-SPECIFICATION>
                                    </INIT-VALUE>
                                </NONQUEUED-RECEIVER-COM-SPEC>
                                <NONQUEUED-RECEIVER-COM-SPEC>
                                    <DATA-ELEMENT-REF DEST="VARIABLE-DATA-PROTOTYPE">/Interfaces/Srif_HW_Inp
                                    <HANDLE-OUT-OF-RANGE>NONE</HANDLE-OUT-OF-RANGE>
                                    <USES-END-TO-END-PROTECTION>false</USES-END-TO-END-PROTECTION>
                                    <ALIVE-TIMEOUT>0</ALIVE-TIMEOUT>
                                    <ENABLE-UPDATE>false</ENABLE-UPDATE>
                                    <HANDLE-NEVER-RECEIVED>false</HANDLE-NEVER-RECEIVED>
                                    <HANDLE-TIMEOUT-TYPE>NONE</HANDLE-TIMEOUT-TYPE>
```

Define AUTOSAR SW architecture



Development of AUTOSAR SW architecture with AUTOSAR Blockset



Property Inspector

Port

Architecture | Info

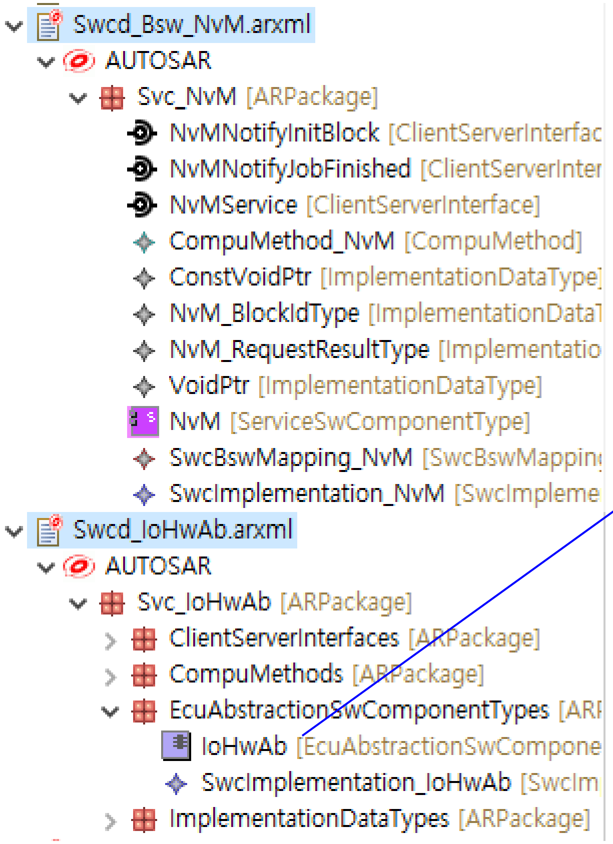
NAME	VALUE
Main	
Name	SP_HW_Input
Interface	Srif_HW_Input_TSL
Kind	Sender
Stereotype	Add..

Round-trip workflow

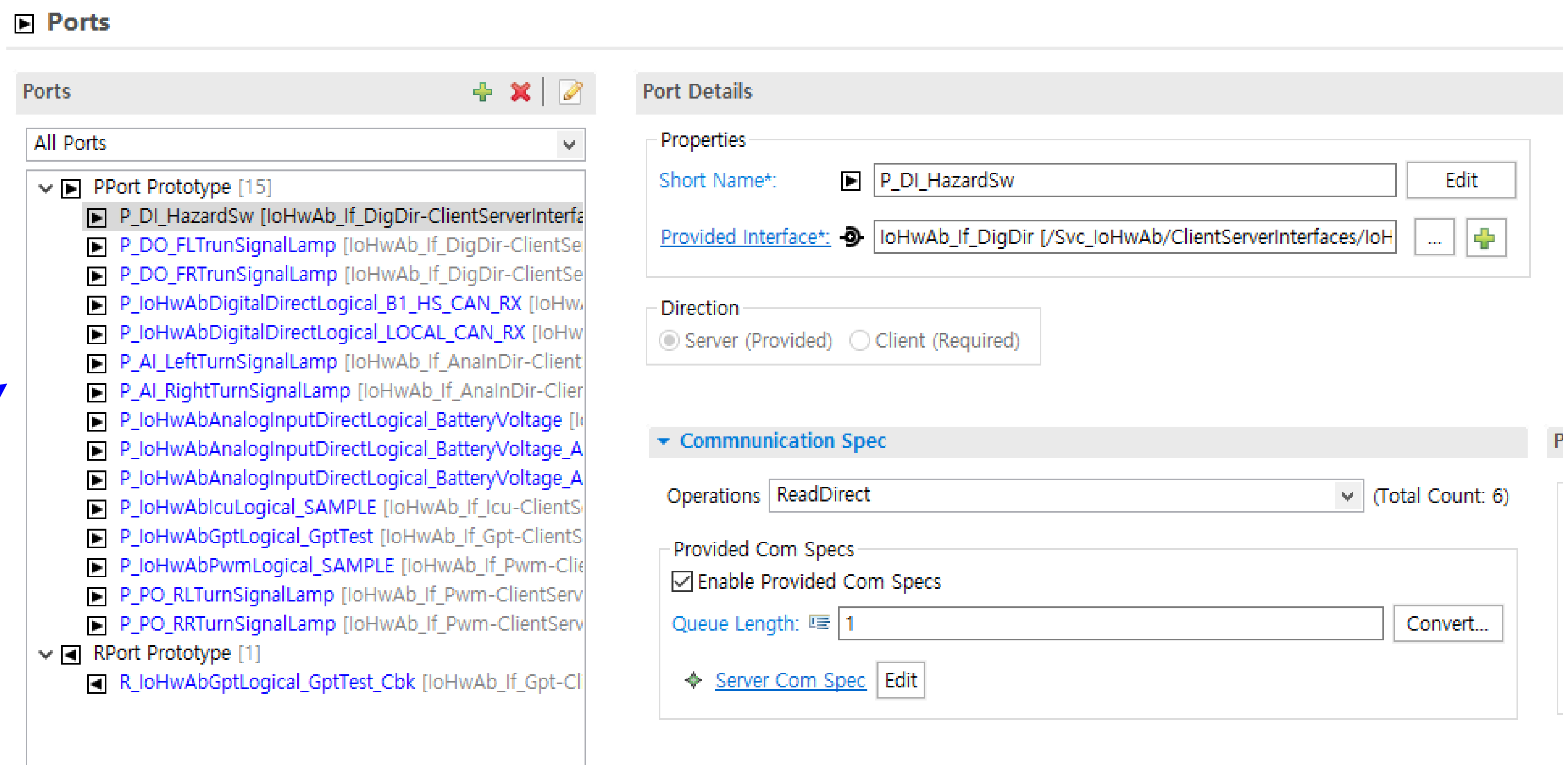
for developing Non-functional SWCs

Round-trip workflow between MATLAB and mobilgene C studio

- Configure BSW module in mobilgene to satisfy the system requirements



Swcd_Bsw_NvM.arxml
AUTOSAR
Svc_NvM [ARPackage]
NvMNotifyInitBlock [ClientServerInterfac
NvMNotifyJobFinished [ClientServerInter
NvMService [ClientServerInterface]
CompuMethod_NvM [CompuMethod]
ConstVoidPtr [ImplementationDataType]
NvM_BlockIdType [ImplementationDataT
NvM_RequestResultType [Implementatio
VoidPtr [ImplementationDataType]
NvM [ServiceSwComponentType]
SwcBswMapping_NvM [SwcBswMappin
SwcImplementation_NvM [SwcImpleme
Swcd_IoHwAb.arxml
AUTOSAR
Svc_IoHwAb [ARPackage]
ClientServerInterfaces [ARPackage]
CompuMethods [ARPackage]
EcuAbstractionSwComponentTypes [ARI
IoHwAb [EcuAbstractionSwCompone
SwcImplementation_IoHwAb [SwcIm
ImplementationDataTypes [ARPackage]



Ports

Ports

All Ports

PPort Prototype [15]
P_DI_HazardSw [IoHwAb_If_DigDir-ClientServerInterfa
P_DO_FLTrunSignalLamp [IoHwAb_If_DigDir-ClientSe
P_DO_FRTrunSignalLamp [IoHwAb_If_DigDir-ClientSe
P_IoHwAbDigitalDirectLogical_B1_HS_CAN_RX [IoHw
P_IoHwAbDigitalDirectLogical_LOCAL_CAN_RX [IoHw
P_AI_LeftTurnSignalLamp [IoHwAb_If_AnalnDir-Client
P_AI_RightTurnSignalLamp [IoHwAb_If_AnalnDir-Clier
P_IoHwAbAnalogInputDirectLogical_BatteryVoltage [I
P_IoHwAbAnalogInputDirectLogical_BatteryVoltage_A
P_IoHwAbAnalogInputDirectLogical_BatteryVoltage_A
P_IoHwAbIcuLogical_SAMPLE [IoHwAb_If_Icu-ClientS
P_IoHwAbGptLogical_GptTest [IoHwAb_If_Gpt-ClientS
P_IoHwAbPwmLogical_SAMPLE [IoHwAb_If_Pwm-Clie
P_PO_RLTurnSignalLamp [IoHwAb_If_Pwm-ClientServ
P_PO_RRTurnSignalLamp [IoHwAb_If_Pwm-ClientServ
RPort Prototype [1]
R_IoHwAbGptLogical_GptTest_Cbk [IoHwAb_If_Gpt-Cl

Port Details

Properties

Short Name*: P_DI_HazardSw

Provided Interface*: IoHwAb_If_DigDir [/Svc_IoHwAb/ClientServerInterfaces/IoH

Direction

● Server (Provided) ○ Client (Required)

Commnunication Spec

Operations ReadDirect (Total Count: 6)

Provided Com Specs

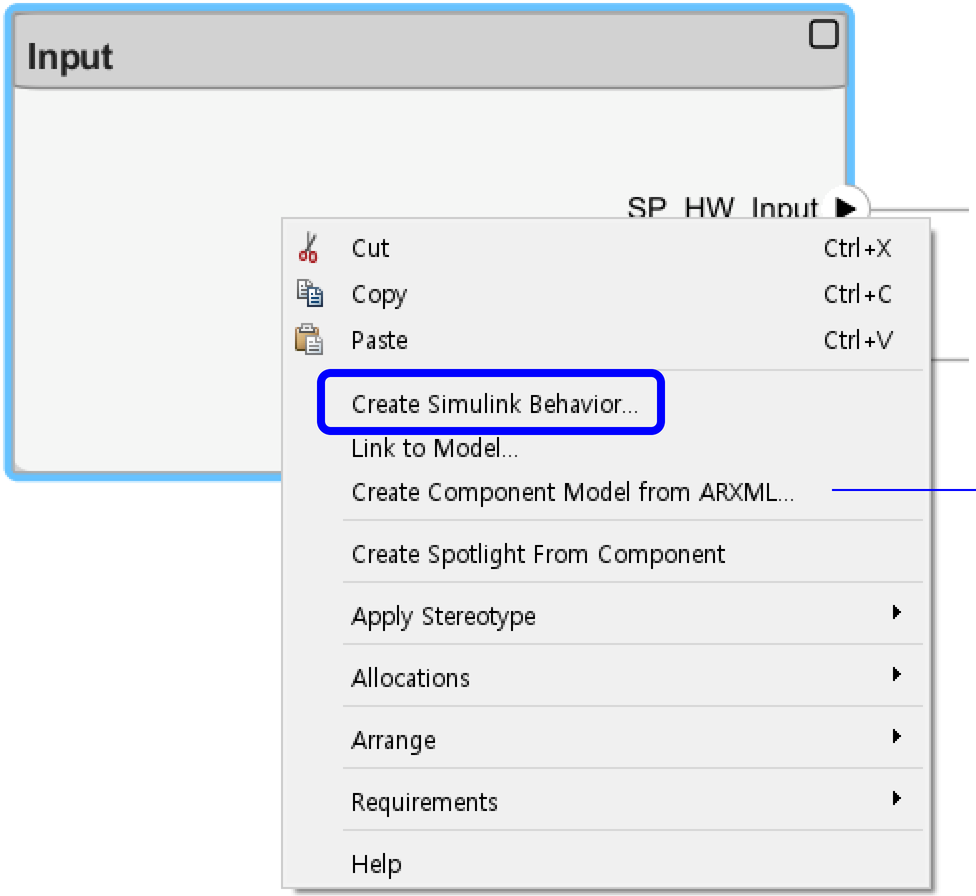
☑ Enable Provided Com Specs

Queue Length: 1

Server Com Spec

Round-trip workflow between MATLAB and mobilgene C studio

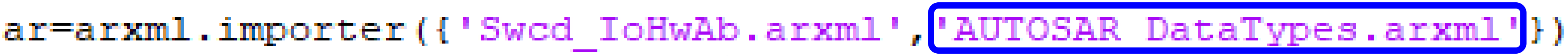
- Import arxml for service block from mobilgene to MATLAB



is not recommended if it contains any SWC

Round-trip workflow between MATLAB and mobilgene C studio

- Input SWC only needs the client-server interface from Swcd_IoHwAb.arxml
- Use MATLAB API to import arxml and update AUTOSAR properties of model



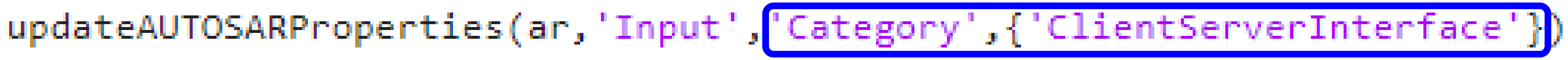
```
ar=arxml.importer({'Swcd_IoHwAb.arxml','AUTOSAR_DataTypes.arxml'})
```

AUTOSAR_DataTypes.arxml

Gets error since it cannot resolve the BaseType

Round-trip workflow between MATLAB and mobilgene C studio

- Input SWC only needs the client-server interface from Swcd_IoHwAb.arxml
- Use MATLAB API to import arxml and update AUTOSAR properties of model

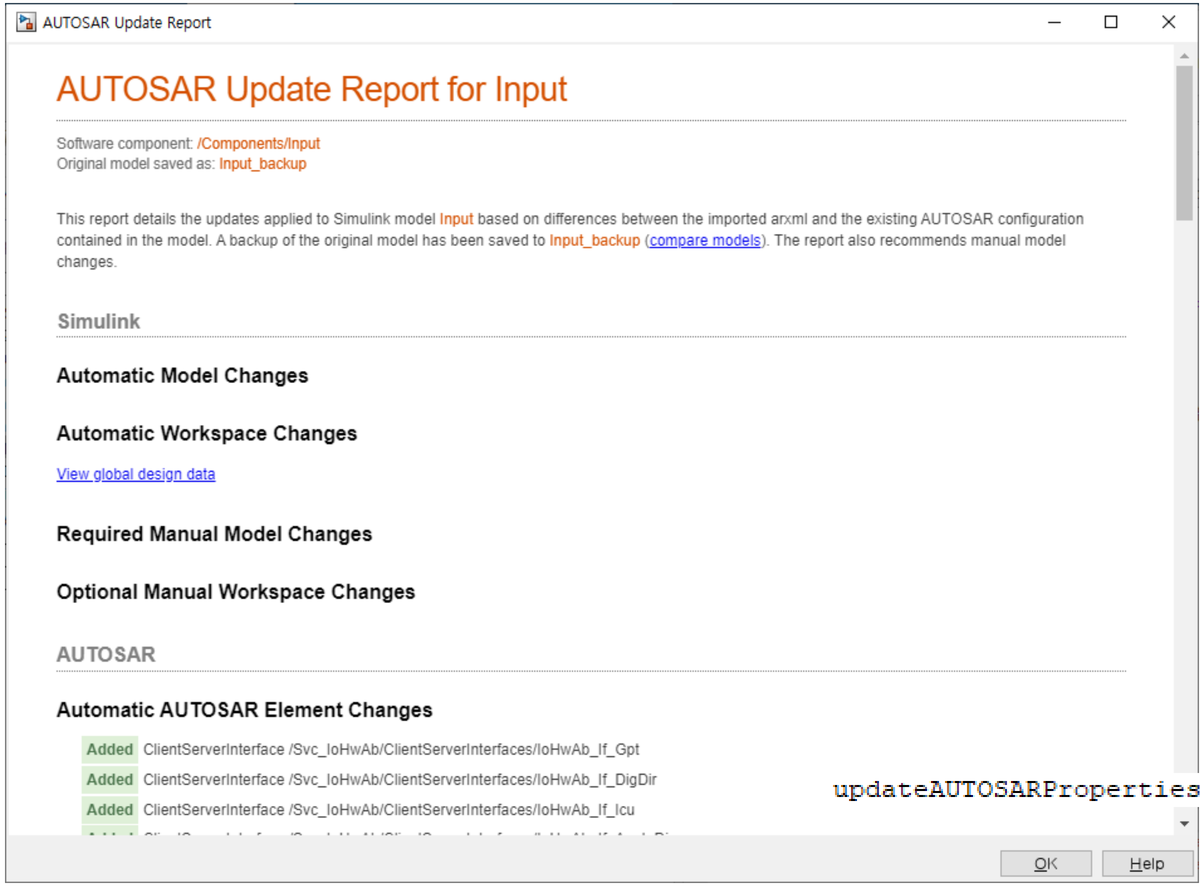


```
updateAUTOSARProperties(ar,'Input','Category',{'ClientServerInterface'})
```

Gets error since it contains SWC

Round-trip workflow between MATLAB and mobilgene C studio

- Client-server interface regarding to IoHwAb is added in AUTOSAR Dictionary



AUTOSAR Update Report for Input

Software component: /Components/Input
Original model saved as: Input_backup

This report details the updates applied to Simulink model Input based on differences between the imported arxml and the existing AUTOSAR configuration contained in the model. A backup of the original model has been saved to Input_backup (compare models). The report also recommends manual model changes.

Simulink

Automatic Model Changes

Automatic Workspace Changes

View global design data

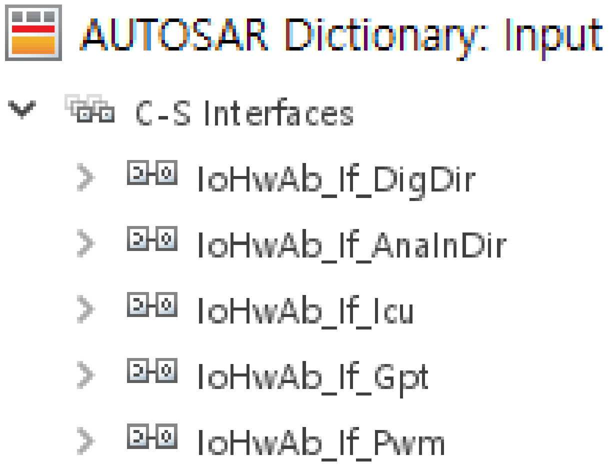
Required Manual Model Changes

Optional Manual Workspace Changes

AUTOSAR

Automatic AUTOSAR Element Changes

Added ClientServerInterface /Svc_IoHwAb/ClientServerInterfaces/IoHwAb_If_Gpt
Added ClientServerInterface /Svc_IoHwAb/ClientServerInterfaces/IoHwAb_If_DigDir
Added ClientServerInterface /Svc_IoHwAb/ClientServerInterfaces/IoHwAb_If_Icu



AUTOSAR Dictionary: Input

C-S Interfaces
- IoHwAb_If_DigDir
- IoHwAb_If_AnalnDir
- IoHwAb_If_Icu
- IoHwAb_If_Gpt
- IoHwAb_If_Pwm

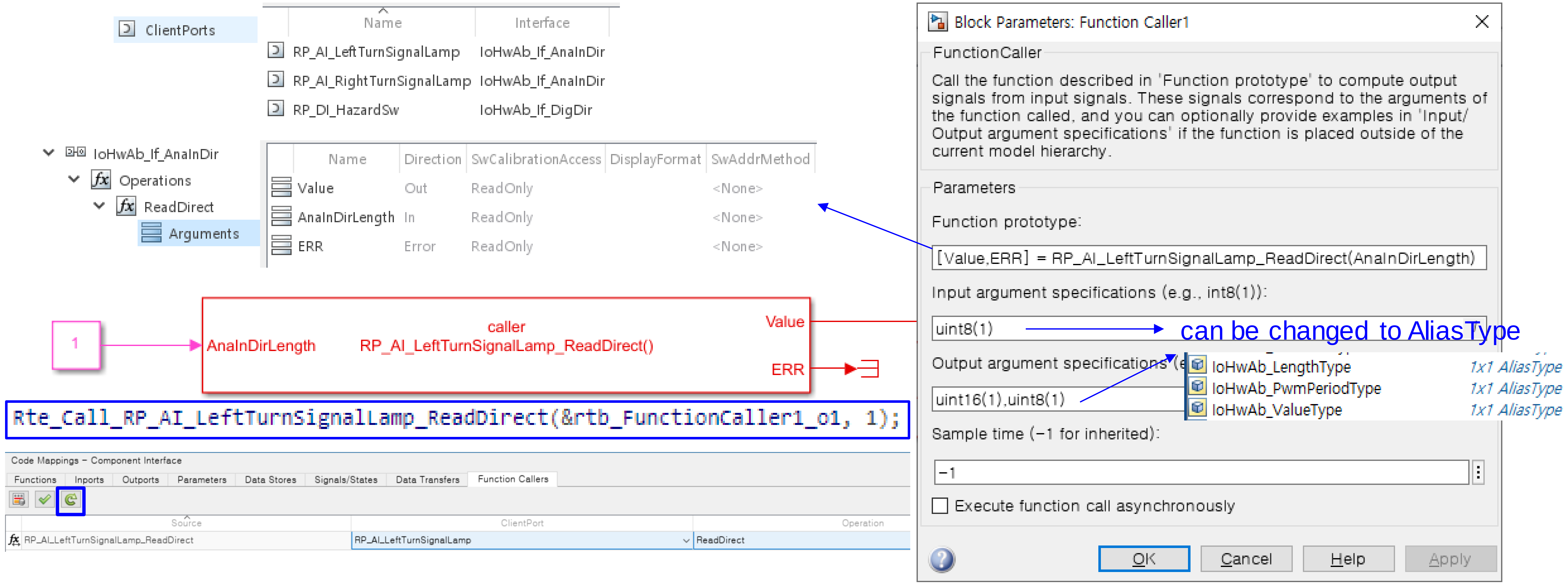
You can ignore this report by adding options

```
updateAUTOSARProperties(ar,'Input','Category',{'ClientServerInterface'},'createReport',false)
```

can be useful to automation script

Round-trip workflow between MATLAB and mobilgene C studio

- Create client port using function caller block and map the properties



ClientPorts

Name	Interface
RP_AI_LeftTurnSignalLamp	IoHwAb_If_AnaInDir
RP_AI_RightTurnSignalLamp	IoHwAb_If_AnaInDir
RP_DI_HazardSw	IoHwAb_If_DigDir

IoHwAb_If_AnaInDir
Operations
ReadDirect
Arguments

Name	Direction	SwCalibrationAccess	DisplayFormat	SwAddrMethod
Value	Out	ReadOnly		<None>
AnaInDirLength	In	ReadOnly		<None>
ERR	Error	ReadOnly		<None>

```
Rte_Call_RP_AI_LeftTurnSignalLamp_ReadDirect(&rtb_FunctionCaller1_o1, 1);
```

Code Mappings - Component Interface

Source	ClientPort	Operation
RP_AI_LeftTurnSignalLamp_ReadDirect	RP_AI_LeftTurnSignalLamp	ReadDirect

Block Parameters: Function Caller1

FunctionCaller

Call the function described in 'Function prototype' to compute output signals from input signals. These signals correspond to the arguments of the function called, and you can optionally provide examples in 'Input/ Output argument specifications' if the function is placed outside of the current model hierarchy.

Parameters

Function prototype:
[Value,ERR] = RP_AI_LeftTurnSignalLamp_ReadDirect(AnaInDirLength)

Input argument specifications (e.g., int8(1)):
uint8(1)

can be changed to AliasType

Output argument specifications (e
uint16(1),uint8(1)

IoHwAb_LengthType	1x1 AliasType
IoHwAb_PwmPeriodType	1x1 AliasType
IoHwAb_ValueType	1x1 AliasType

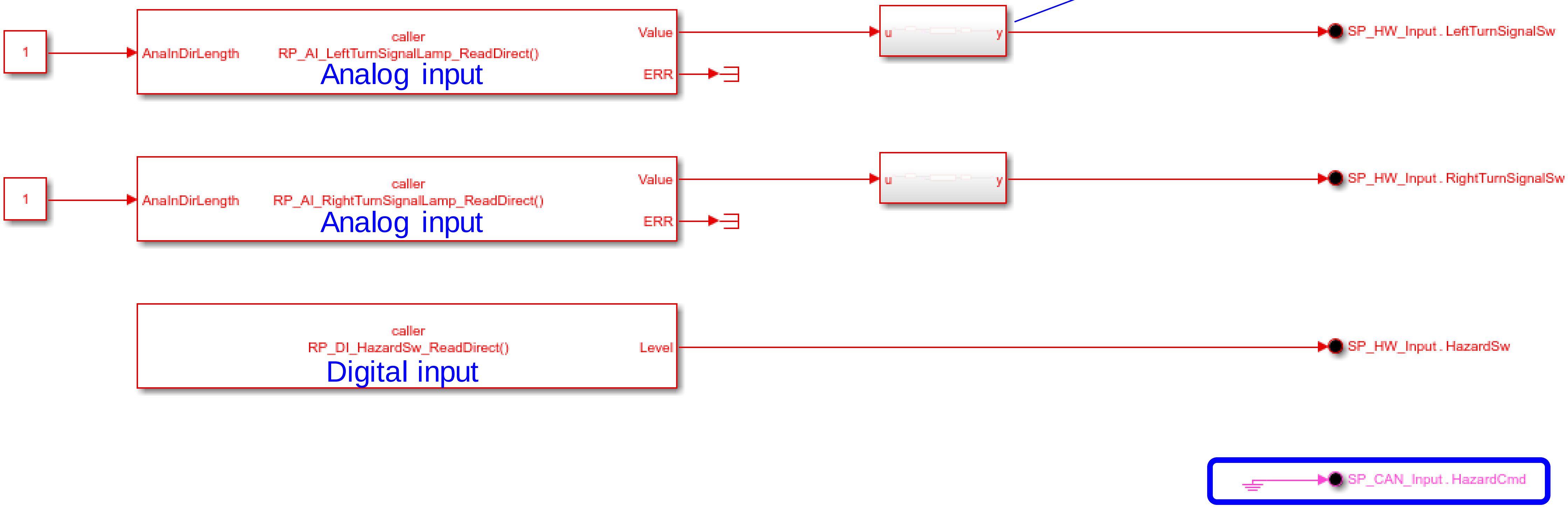
Sample time (-1 for inherited):
-1

□ Execute function call asynchronously

Round-trip workflow between MATLAB and mobilgene C studio

- Input SWC model using IoHwAb services

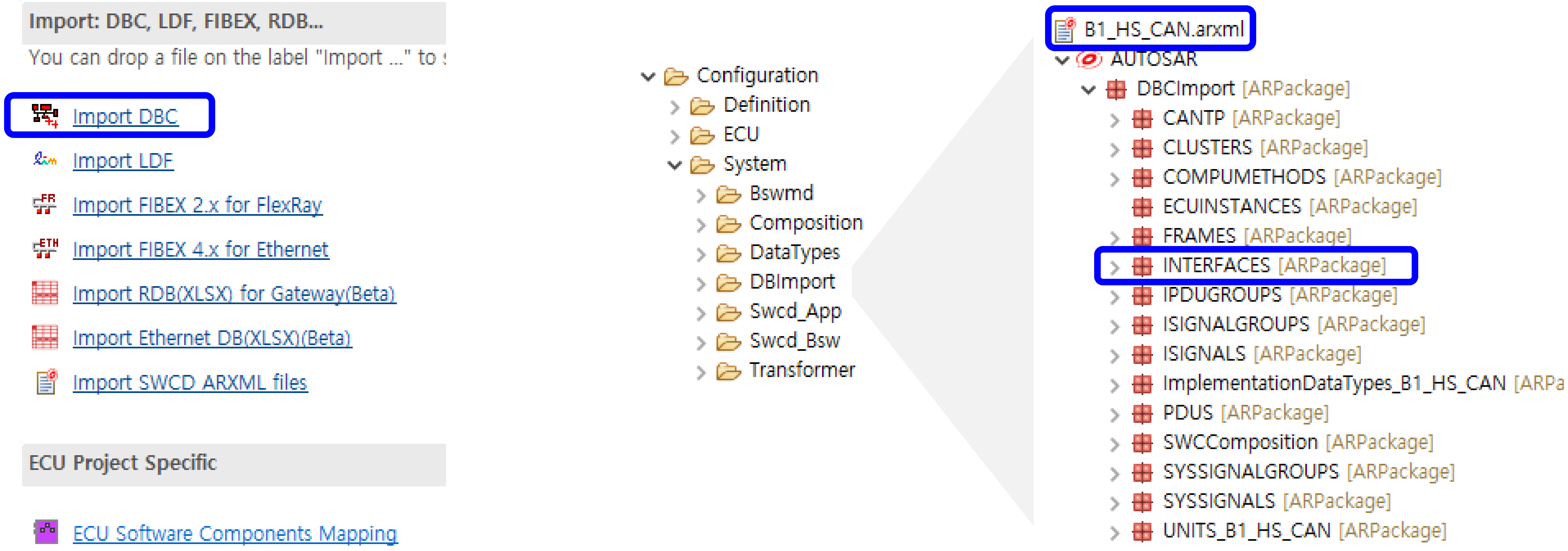
Convert ADC value to ON/OFF


1
AnaInDirLength
caller
RP_AI_LeftTurnSignalLamp_ReadDirect()
Analog input
Value
ERR
u
y
SP_HW_Input . LeftTurnSignalSw
1
AnaInDirLength
caller
RP_AI_RightTurnSignalLamp_ReadDirect()
Analog input
Value
ERR
u
y
SP_HW_Input . RightTurnSignalSw
caller
RP_DI_HazardSw_ReadDirect()
Digital input
Level
SP_HW_Input . HazardSw
SP_CAN_Input . HazardCmd

How to get signals from CAN message?

Round-trip workflow between MATLAB and mobilgene C studio

- To get signals from CAN message, import the arxml file including DBC information from mobilgene



Import: DBC, LDF, FIBEX, RDB...

You can drop a file on the label "Import ..." to :

- Import DBC
- Import LDF
- Import FIBEX 2.x for FlexRay
- Import FIBEX 4.x for Ethernet
- Import RDB(XLSX) for Gateway(Beta)
- Import Ethernet DB(XLSX)(Beta)
- Import SWCD ARXML files

ECU Project Specific

- ECU Software Components Mapping

```
Configuration
  Definition
  ECU
  System
    Bswmd
    Composition
    DataTypes
    DBImport
    Swcd_App
    Swcd_Bsw
    Transformer
```

```
B1_HS_CAN.arxml
  AUTOSAR
    DBCImport [ARPackage]
      CANTP [ARPackage]
      CLUSTERS [ARPackage]
      COMPUMETHODS [ARPackage]
      ECUINSTANCES [ARPackage]
      FRAMES [ARPackage]
      INTERFACES [ARPackage]
      IPDUGROUPS [ARPackage]
      ISIGNALGROUPS [ARPackage]
      ISIGNALS [ARPackage]
      ImplementationDataTypes_B1_HS_CAN [ARPa
      PDUS [ARPackage]
      SWCComposition [ARPackage]
      SYSSIGNALGROUPS [ARPackage]
      SYSSIGNALS [ARPackage]
      UNITS_B1_HS_CAN [ARPackage]
```

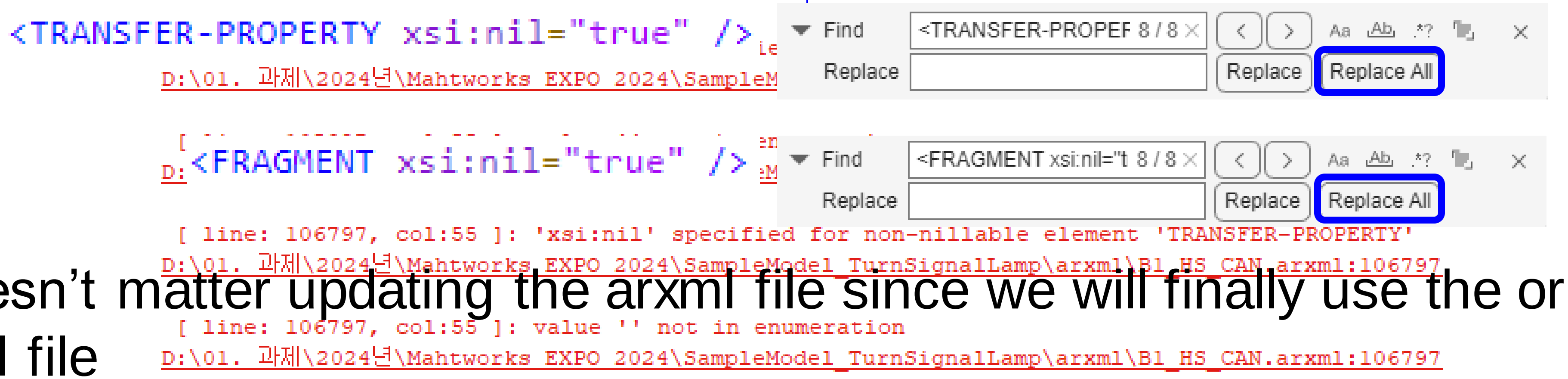
Round-trip workflow between MATLAB and mobilgene C studio

- Some additional works should be done before importing the arxml file
- When we import B1_HS_CAN.arxml as the same way as IoHwAb,

```
ar=arxml.importer({'B1_HS_CAN.arxml','AUTOSAR_DataTypes.arxml'})
```

Gets error

Ctrl+F



```
<TRANSFER-PROPERTY xsi:nil="true" />
D:\01. 과제\2024년\Mahtworks EXPO 2024\SampleM

<FRAGMENT xsi:nil="true" />

 [ line: 106797, col:55 ]: 'xsi:nil' specified for non-nillable element 'TRANSFER-PROPERTY'
D:\01. 과제\2024년\Mahtworks EXPO 2024\SampleModel_TurnSignalLamp\arxml\B1_HS_CAN.arxml:106797

 [ line: 106797, col:55 ]: value '' not in enumeration
D:\01. 과제\2024년\Mahtworks EXPO 2024\SampleModel_TurnSignalLamp\arxml\B1_HS_CAN.arxml:106797
```

Find: <TRANSFER-PROPEF 8 / 8
Replace:

Find: <FRAGMENT xsi:nil="t 8 / 8
Replace:

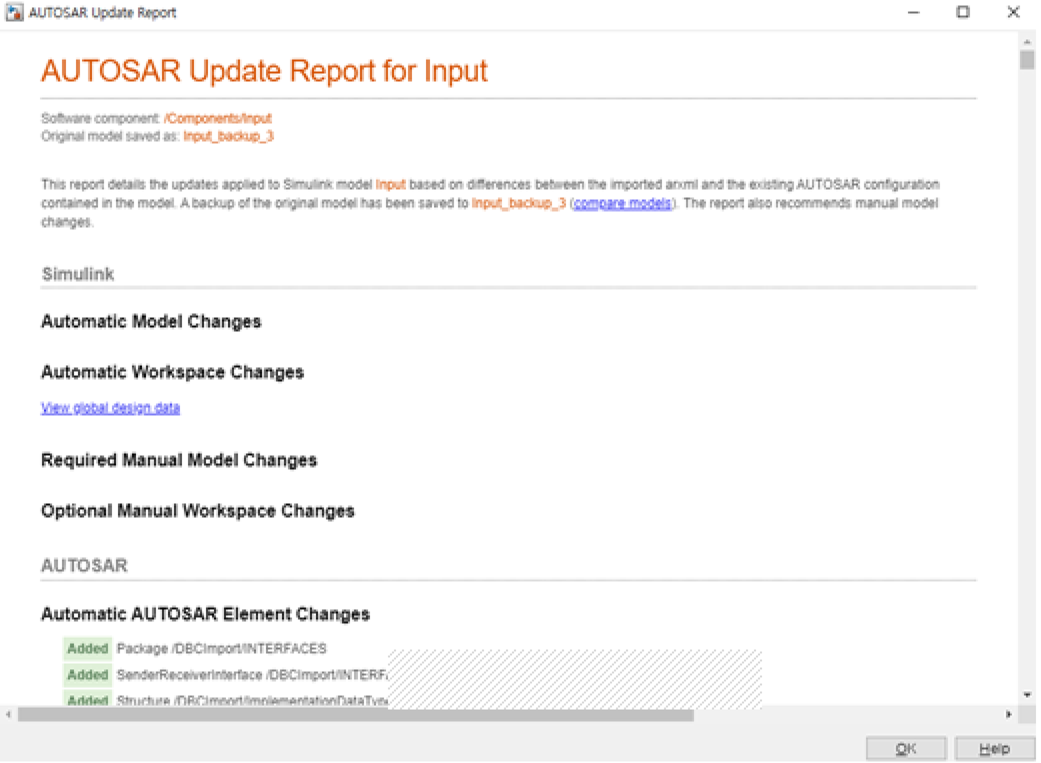
- It doesn't matter updating the arxml file since we will finally use the original arxml file

Conflict Issue

Round-trip workflow between MATLAB and mobilgene C studio

- Import the sender-receiver interface regarding to DBC by using API

```
updateAUTOSARProperties(ar,'Input','Category',{'SenderReceiverInterface'})
```



AUTOSAR Update Report for Input

Software component: /Components/Input
Original model saved as: Input_backup_3

This report details the updates applied to Simulink model Input based on differences between the imported arxml and the existing AUTOSAR configuration contained in the model. A backup of the original model has been saved to Input_backup_3 (compare models). The report also recommends manual model changes.

Simulink

Automatic Model Changes

Automatic Workspace Changes

View global design data

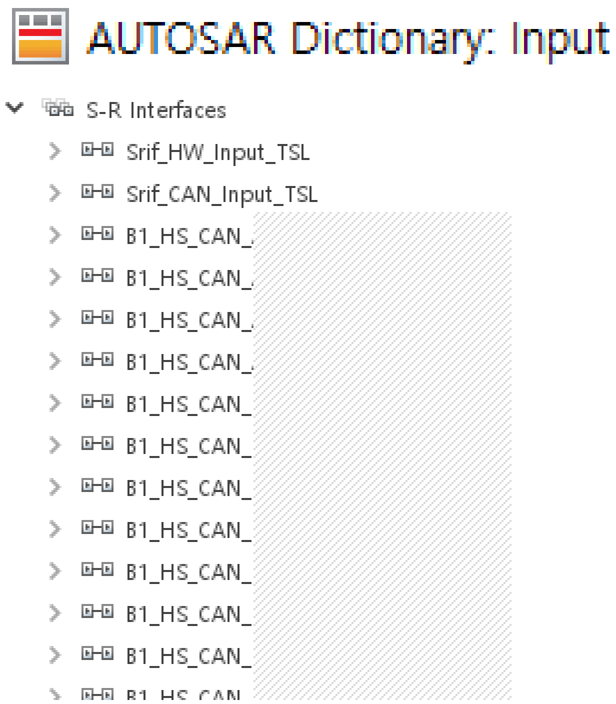
Required Manual Model Changes

Optional Manual Workspace Changes

AUTOSAR

Automatic AUTOSAR Element Changes

Added Package /DBCImport/INTERFACES
Added SenderReceiverInterface /DBCImport/INTERF [redacted]
Added Structure /DBCImport/ImplementationDataTyp



AUTOSAR Dictionary: Input

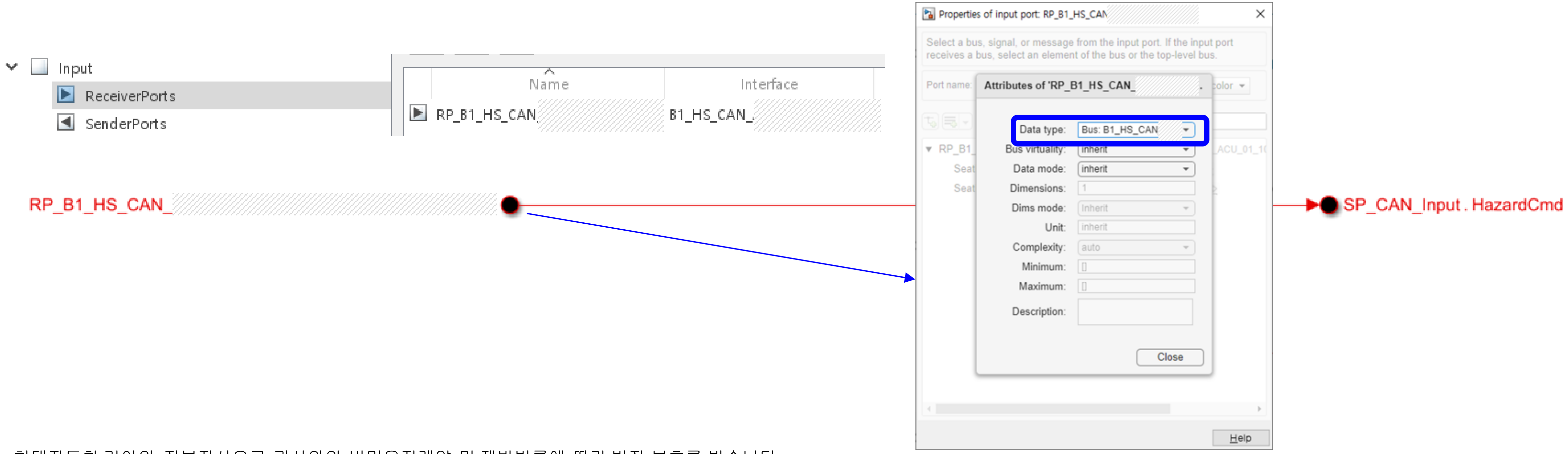
S-R Interfaces
Srif_HW_Input_TSL
Srif_CAN_Input_TSL
B1_HS_CAN_ [redacted]
B1_HS_CAN_
B1_HS_CAN_
B1_HS_CAN_
B1_HS_CAN_
B1_HS_CAN_
B1_HS_CAN_
B1_HS_CAN_
B1_HS_CAN_
B1_HS_CAN_
B1_HS_CAN_
B1_HS_CAN_

Round-trip workflow between MATLAB and mobilgene C studio

- Define the SimulinkBus data type for S-R interface in Simulink Data Dictionary (developing automation MATLAB script is recommended)

B1_HS_CAN_[redacted]

- Add In Bus Element and set it's properties for mapping to Receiver Port



RP_B1_HS_CAN[redacted] B1_HS_CAN_[redacted]

Properties of input port: RP_B1_HS_CAN[redacted]

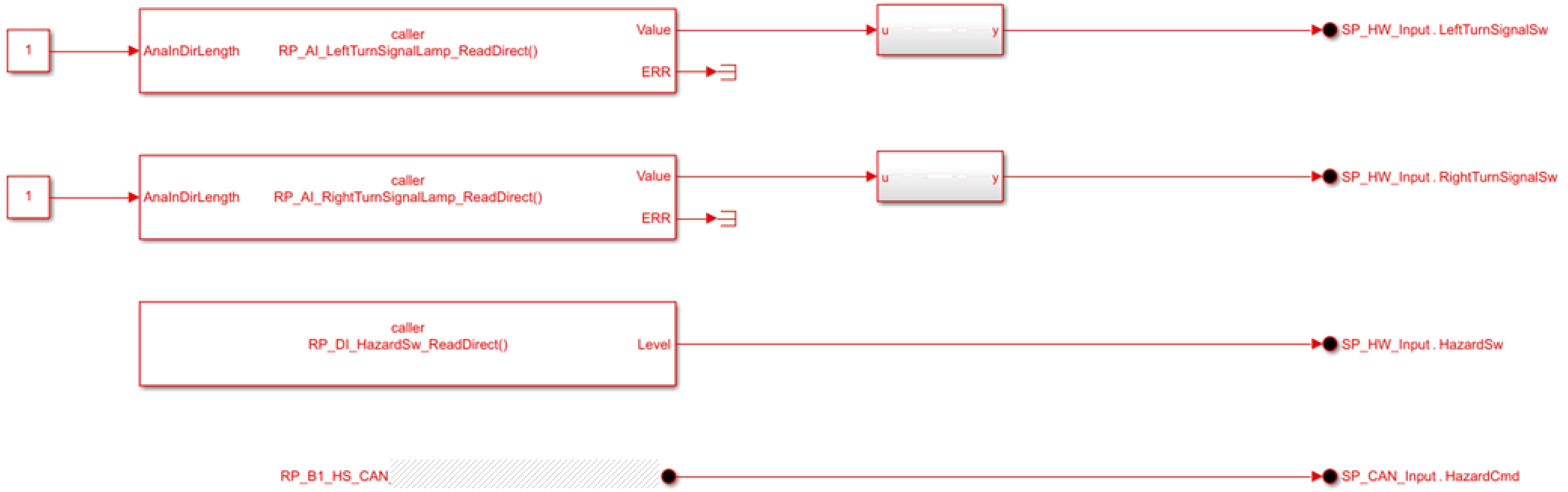
Select a bus, signal, or message from the input port. If the input port receives a bus, select an element of the bus or the top-level bus.

Attributes of 'RP_B1_HS_CAN_[redacted].

Data type: Bus: B1_HS_CAN[redacted]
Bus virtuality: inherit
Data mode: inherit
Dimensions: 1
Dims mode: Inherit
Unit: inherit
Complexity: auto
Minimum: []
Maximum: []
Description:

Round-trip workflow between MATLAB and mobilgene C studio

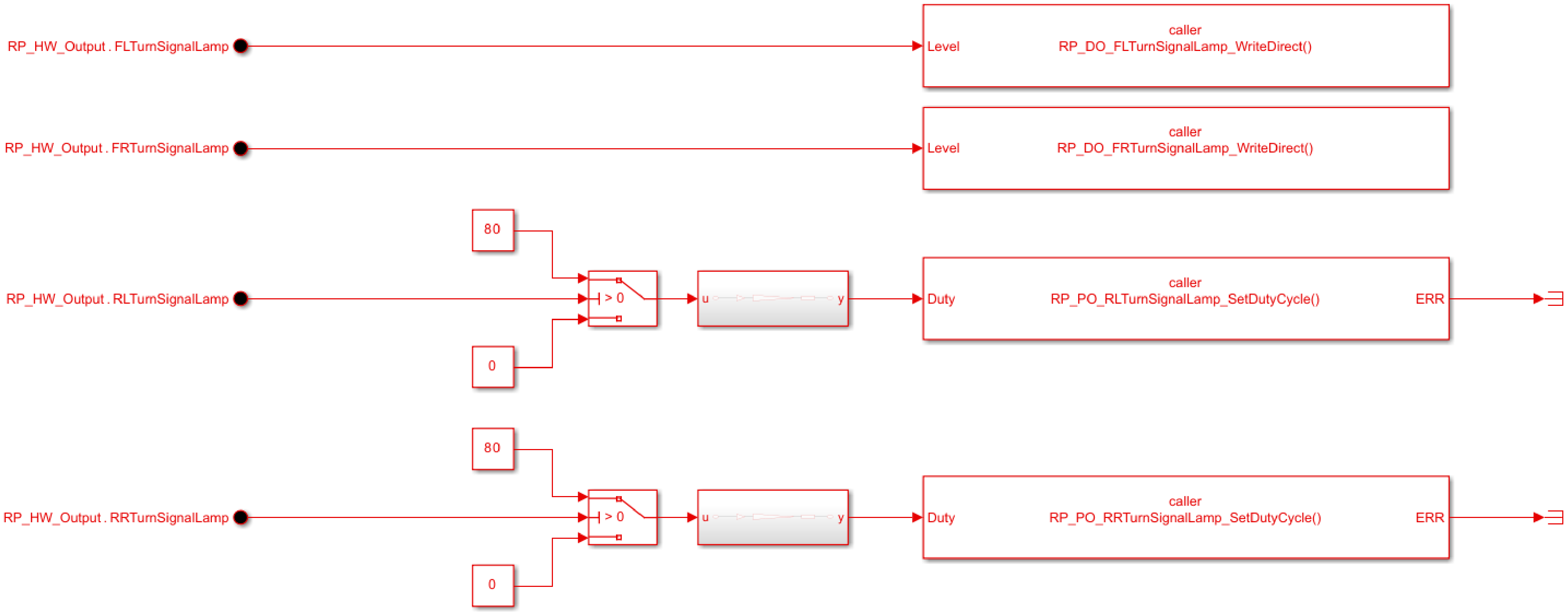
- Final input SWC model


1
AnalnDirLength
caller
RP_AI_LeftTurnSignalLamp_ReadDirect()
Value
ERR
u
y
SP_HW_Input . LeftTurnSignalSw
1
AnalnDirLength
caller
RP_AI_RightTurnSignalLamp_ReadDirect()
Value
ERR
u
y
SP_HW_Input . RightTurnSignalSw
caller
RP_DI_HazardSw_ReadDirect()
Level
SP_HW_Input . HazardSw
RP_B1_HS_CAN
SP_CAN_Input . HazardCmd

- The CAN message is read every step in this case, but reading communication messages will be implemented as DataReceivedEvent and DataReceiveErrorEvent(for timeout) in real applications

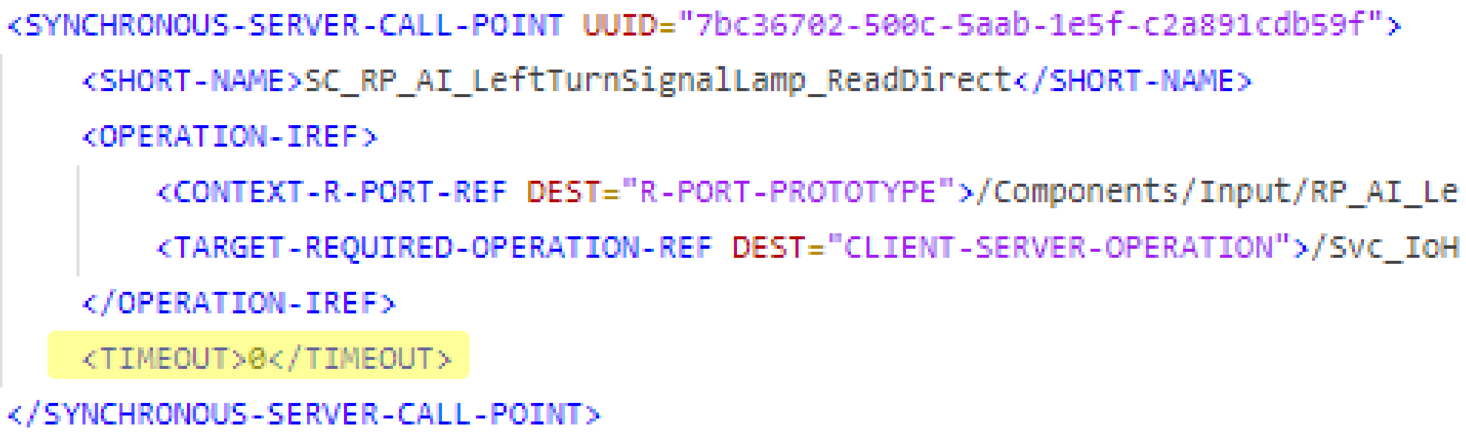
Round-trip workflow between MATLAB and mobilgene C studio

- Output SWC model


RP_HW_Output . FLTurnSignalLamp
Level
caller
RP_DO_FLTurnSignalLamp_WriteDirect()
RP_HW_Output . FRTurnSignalLamp
Level
caller
RP_DO_FRTurnSignalLamp_WriteDirect()
80
RP_HW_Output . RLTurnSignalLamp
> 0
0
u
y
Duty
caller
RP_PO_RLTurnSignalLamp_SetDutyCycle()
ERR
80
RP_HW_Output . RRTurnSignalLamp
> 0
0
u
y
Duty
caller
RP_PO_RRTurnSignalLamp_SetDutyCycle()
ERR

Round-trip workflow between MATLAB and mobilgene C studio

- If a SWC use server call point, the timeout value is set to 1us by default in MATLAB
- mobilgene SWP 4.0.3 does not support non-zero timeout value

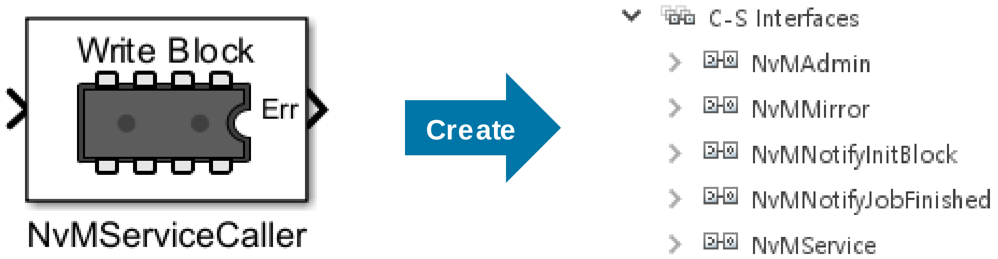


```
<SYNCHRONOUS-SERVER-CALL-POINT UUID="7bc36702-500c-5aab-1e5f-c2a891cdb59f">
    <SHORT-NAME>SC_RP_AI_LeftTurnSignalLamp_ReadDirect</SHORT-NAME>
    <OPERATION-IREF>
        <CONTEXT-R-PORT-REF DEST="R-PORT-PROTOTYPE">/Components/Input/RP_AI_Le
        <TARGET-REQUIRED-OPERATION-REF DEST="CLIENT-SERVER-OPERATION">/Svc_IoH
    </OPERATION-IREF>
    <TIMEOUT>0</TIMEOUT>
</SYNCHRONOUS-SERVER-CALL-POINT>
```

Conflict Issue

Round-trip workflow between MATLAB and mobilgene C studio

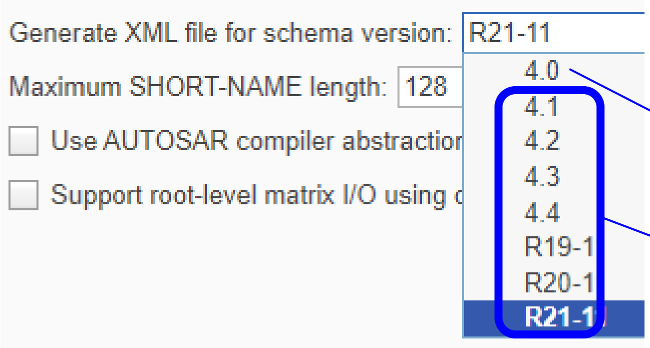
- NvM SWC uses C-S interface from BSW NvM service block
- If you add NvMServiceCaller block in your model, C-S interface regarding to NvM will be created in AUTOSAR Dictionary



- C-S Interfaces
  - NvMAdmin
  - NvMMirror
  - NvMNotifyInitBlock
  - NvMNotifyJobFinished
  - NvMService

Round-trip workflow between MATLAB and mobilgene C studio

- Differences of AUTOSAR schema between MATLAB and mobilgene



Generate XML file for schema version: R21-11

Maximum SHORT-NAME length: 128

□ Use AUTOSAR compiler abstractio

□ Support root-level matrix I/O using c

MATLAB		mobilgene SWP 4.0.3	
Argument of WriteBlock	Argument of JobFinished	Argument of WriteBlock	Argument of JobFinished
SrcPtr	ServiceId	Source	ServiceId
SrcPtr	BlockRequest		

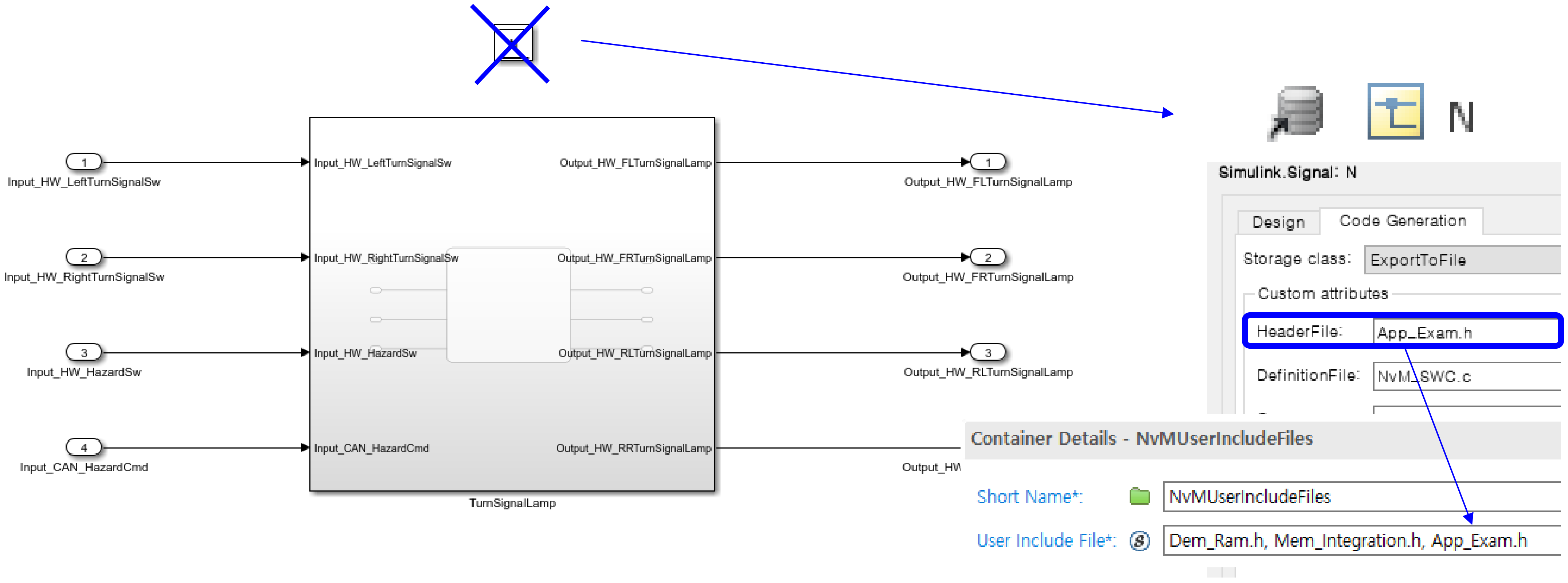
Same with mobilgene SWP 4.4.0

- Importing C-S interface from Swcd_Bsw_NvM.arxml without configuring the schema version option

Conflict Issue

Round-trip workflow between MATLAB and mobilgene C studio

- Remove the Data Store Memory block in TurnSignalLamp SWC
- Define the global variable N(Ram Block) in .sldd as Simulnk.Signal data type


1
Input_HW_LeftTurnSignalSw
2
Input_HW_RightTurnSignalSw
3
Input_HW_HazardSw
4
Input_CAN_HazardCmd
Input_HW_LeftTurnSignalSw
Input_HW_RightTurnSignalSw
Input_HW_HazardSw
Input_CAN_HazardCmd
Output_HW_FLTurnSignalLamp
Output_HW_FRTurnSignalLamp
Output_HW_RLTurnSignalLamp
Output_HW_RRTurnSignalLamp
TurnSignalLamp
1
Output_HW_FLTurnSignalLamp
2
Output_HW_FRTurnSignalLamp
3
Output_HW_RLTurnSignalLamp
Output_HW
N
Design
Code Generation
Custom attributes

Simulink.Signal: N

Storage class: ExportToFile

HeaderFile: App_Exam.h

DefinitionFile: NvM_SWC.c

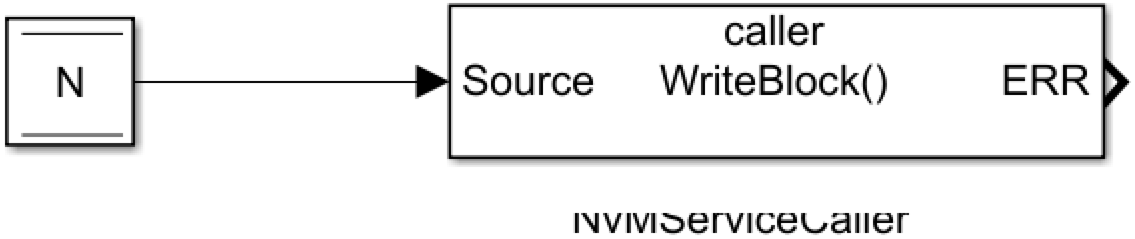
Container Details - NvMUserIncludeFiles

Short Name*: NvMUserIncludeFiles

User Include File*: Dem_Ram.h, Mem_Integration.h, App_Exam.h

Round-trip workflow between MATLAB and mobilgene C studio

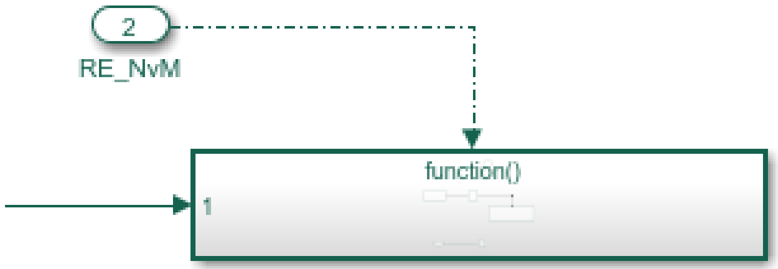
- Use Function Caller block instead of NvMServiceCaller



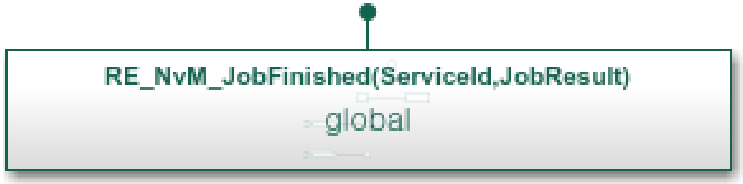
본 문서는 현대자동차·기아의 정보자산으로 귀사와의 비밀유지계약 및 제반법률에 따라 법적 보호를 받습니다.

Round-trip workflow between MATLAB and mobilgene C studio

- NvM SWC calls WriteBlock when it receives a specific eventual CAN message
- Writing can be done only if the status of the flag variable JobF in the JobFinished function is OK
- We will skip the developing progress of the NvM SWC



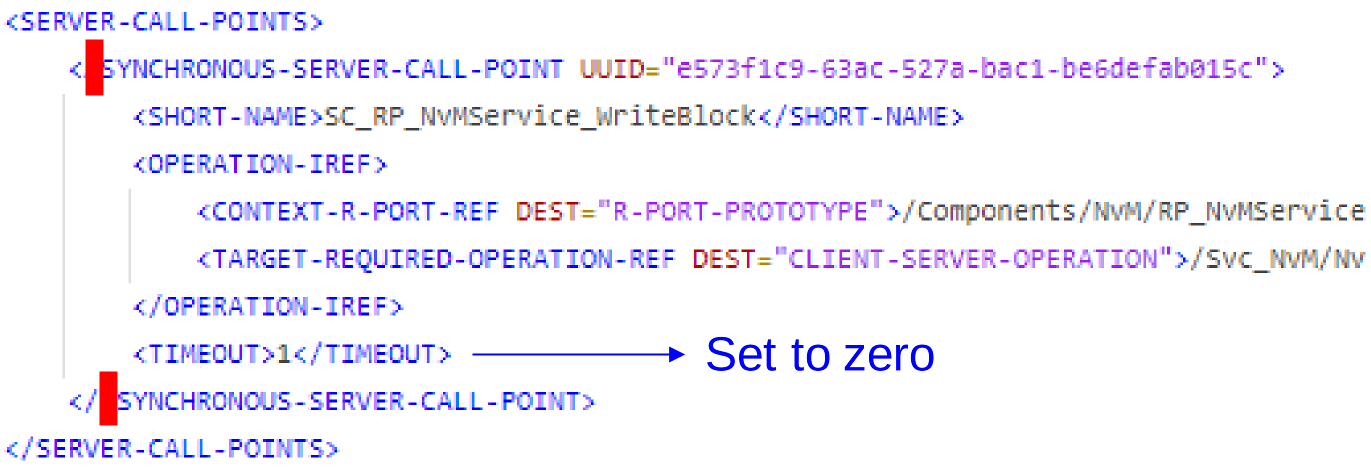
DataReceivedEvent



OperationInvokedEvent

Round-trip workflow between MATLAB and mobilgene C studio

- Only asynchronous server call point is generated for NvM service component in MATLAB
- mobilgene SWP 4.0.3 does not support asynchronous server call point
- Replace ASYNCH to SYNCH by using fileread, strrep MATLAB APIs



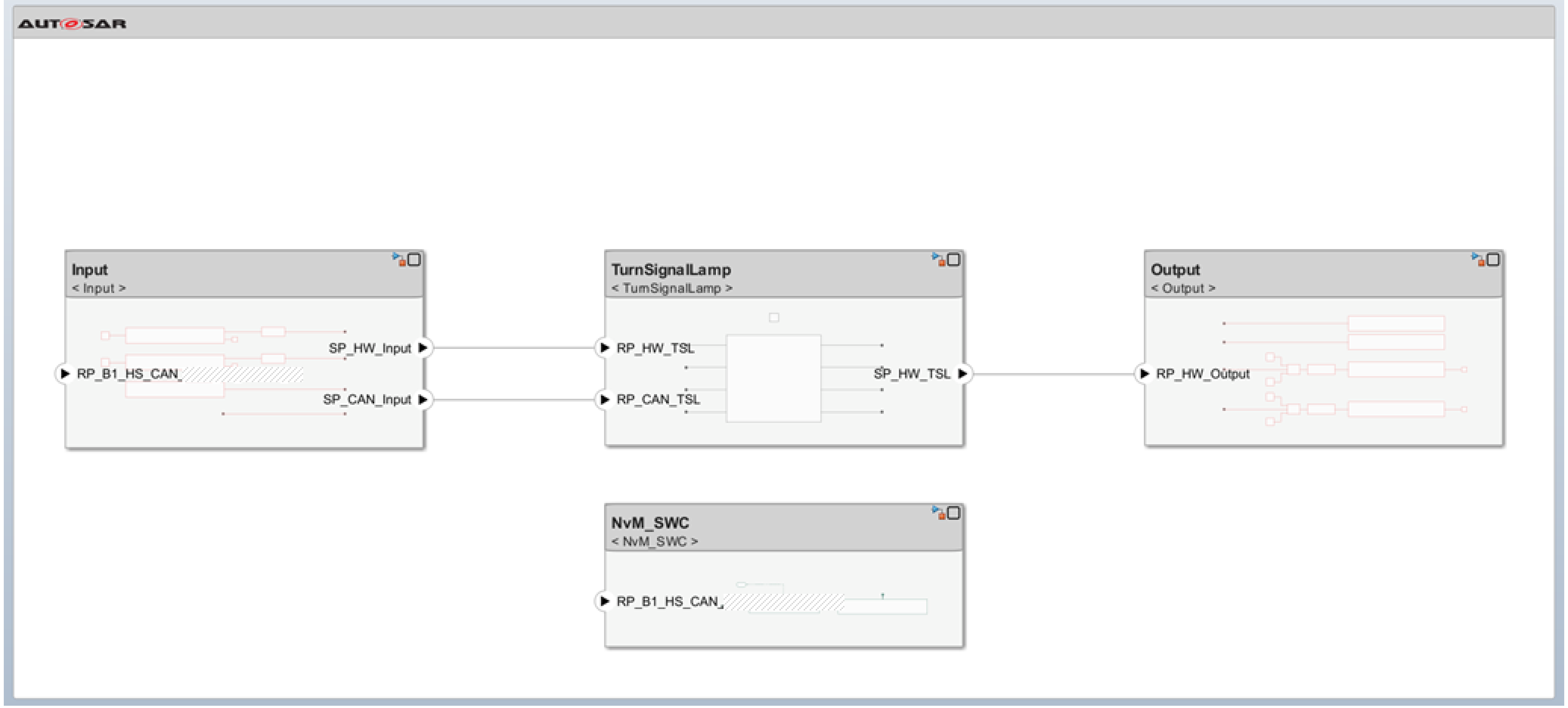
```
<SERVER-CALL-POINTS>
    <SYNCHRONOUS-SERVER-CALL-POINT UUID="e573f1c9-63ac-527a-bac1-be6defab015c">
        <SHORT-NAME>SC_RP_NvMService_WriteBlock</SHORT-NAME>
        <OPERATION-IREF>
            <CONTEXT-R-PORT-REF DEST="R-PORT-PROTOTYPE">/Components/NvM/RP_NvMService
            <TARGET-REQUIRED-OPERATION-REF DEST="CLIENT-SERVER-OPERATION">/Svc_NvM/Nv
        </OPERATION-IREF>
        <TIMEOUT>1</TIMEOUT>
    </SYNCHRONOUS-SERVER-CALL-POINT>
</SERVER-CALL-POINTS>
```

Set to zero

Conflict Issue

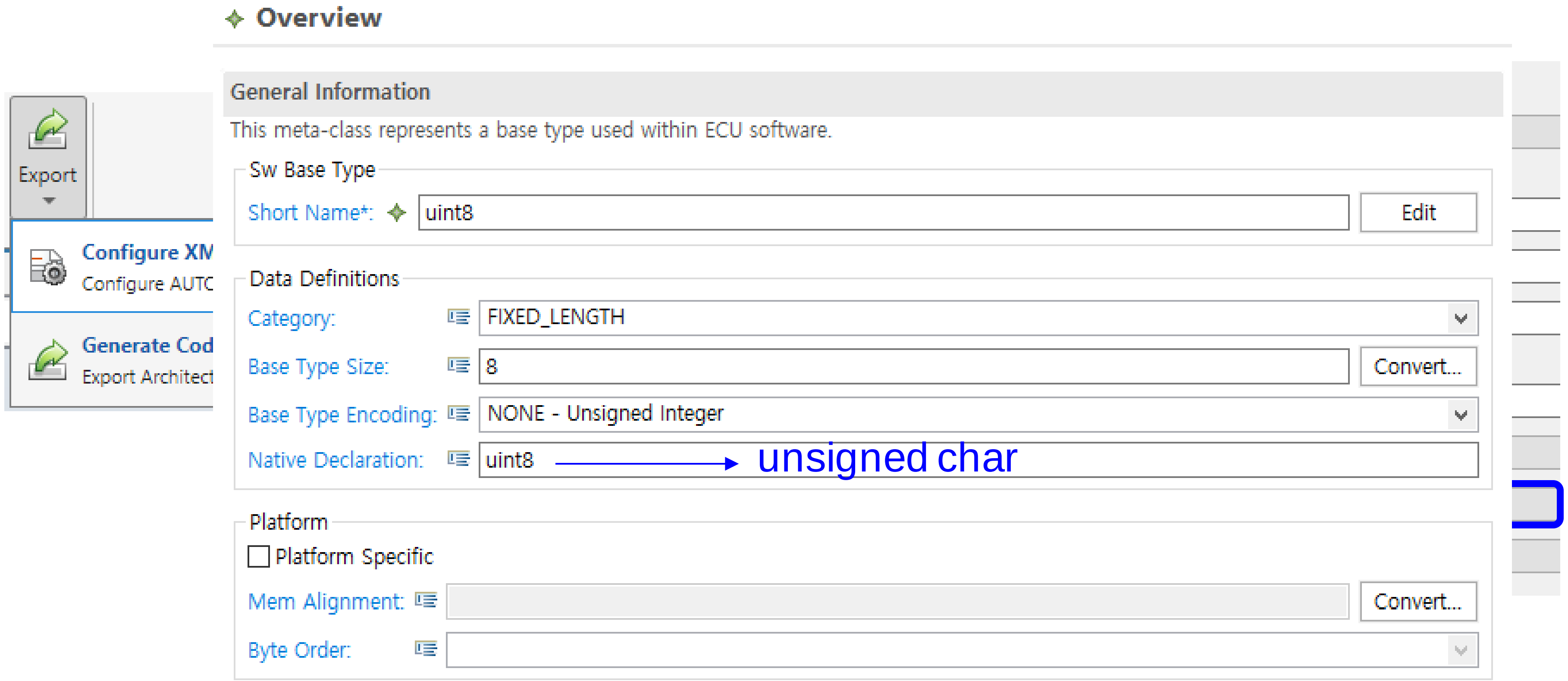
Round-trip workflow between MATLAB and mobilgene C studio

- Final AUTOSAR SW architecture model



Round-trip workflow between MATLAB and mobilgene C studio

- Change native declaration SwBaseTypes because mobilgene can import only Platform Type



✦ Overview

General Information

This meta-class represents a base type used within ECU software.

Sw Base Type

Short Name*: uint8

Data Definitions

Category: FIXED_LENGTH

Base Type Size: 8

Base Type Encoding: NONE - Unsigned Integer

Native Declaration: uint8 → unsigned char

Platform

□ Platform Specific

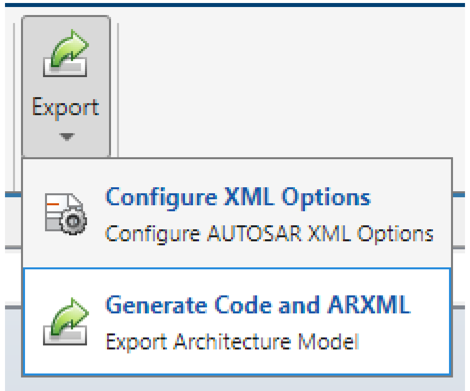
Mem Alignment:

Byte Order:

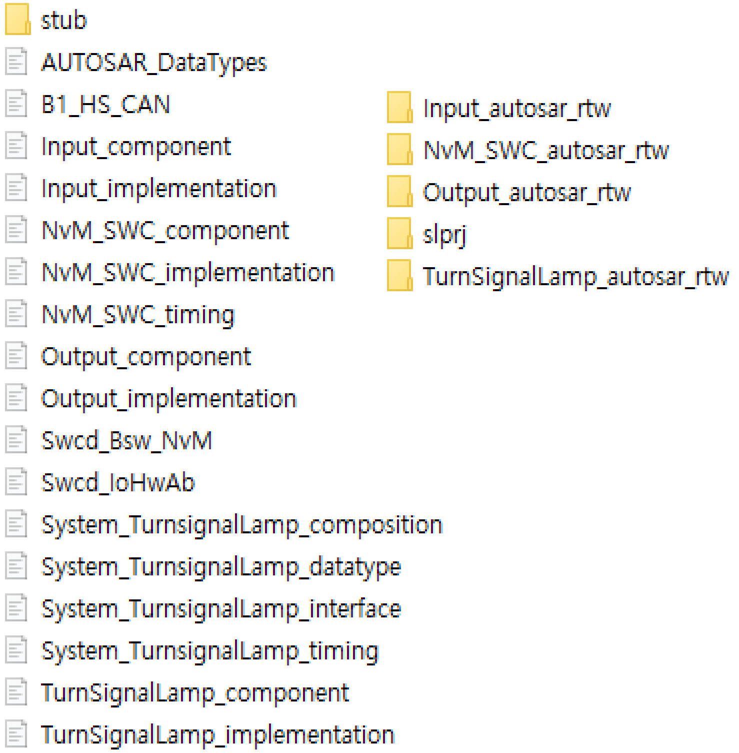
Conflict Issue

Round-trip workflow between MATLAB and mobilgene C studio

- Generate code and ARXML
- Copy arxml files and source files to mobilgene project



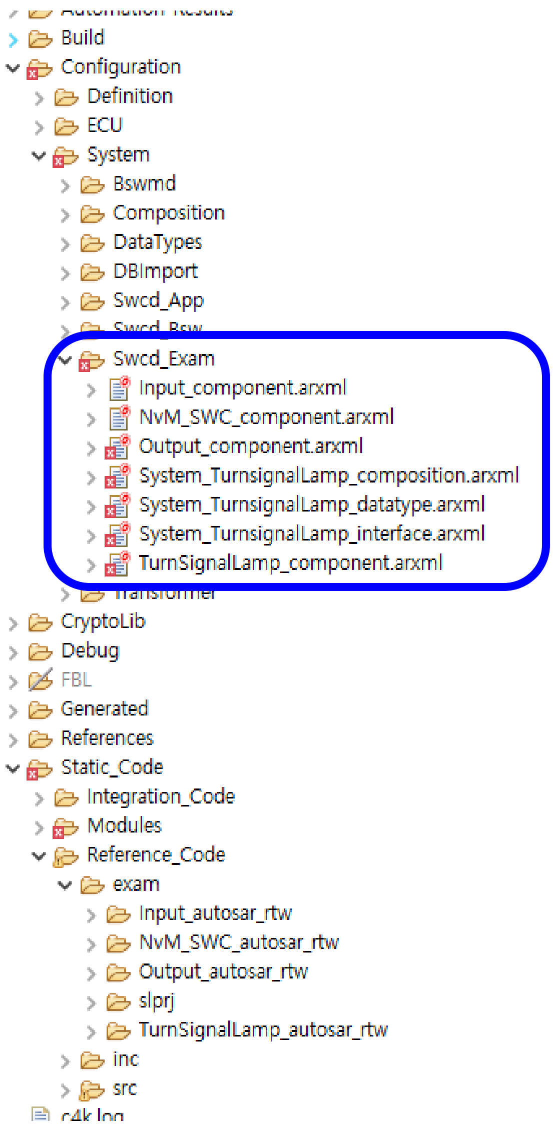
Create



stub
AUTOSAR_DataTypes
B1_HS_CAN
Input_component
Input_implementation
NvM_SWC_component
NvM_SWC_implementation
NvM_SWC_timing
Output_component
Output_implementation
Swcd_Bsw_NvM
Swcd_IoHwAb
System_TurnsignalLamp_composition
System_TurnsignalLamp_datatype
System_TurnsignalLamp_interface
System_TurnsignalLamp_timing
TurnSignalLamp_component
TurnSignalLamp_implementation

Input_autosar_rtw
NvM_SWC_autosar_rtw
Output_autosar_rtw
slprj
TurnSignalLamp_autosar_rtw

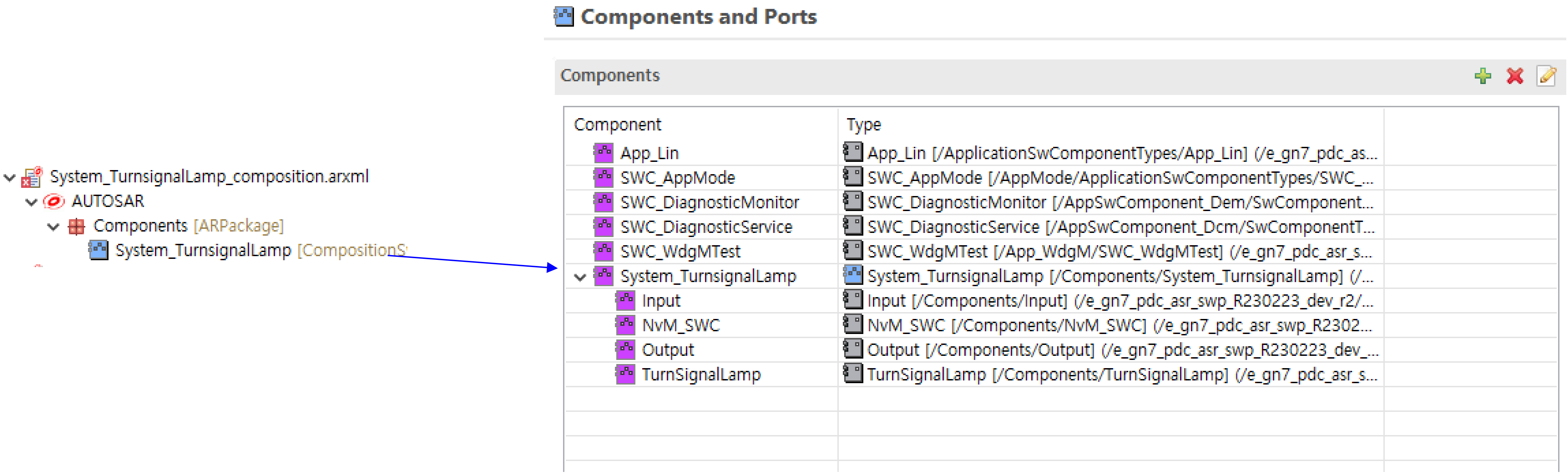
Copy



```
Automation Results
Build
Configuration
  Definition
  ECU
  System
    Bswmd
    Composition
    DataTypes
    DBImport
    Swcd_App
    Swcd_Bsw
    Swcd_Exam
      Input_component.arxml
      NvM_SWC_component.arxml
      Output_component.arxml
      System_TurnsignalLamp_composition.arxml
      System_TurnsignalLamp_datatype.arxml
      System_TurnsignalLamp_interface.arxml
      TurnSignalLamp_component.arxml
    Transformer
CryptoLib
Debug
FBL
Generated
References
Static_Code
  Integration_Code
  Modules
  Reference_Code
    exam
      Input_autosar_rtw
      NvM_SWC_autosar_rtw
      Output_autosar_rtw
      slprj
      TurnSignalLamp_autosar_rtw
    inc
    src
c4k.log
```

Integration SW in mobilgene

- Add the composition generated from MATLAB into CSWC_RootComposition



- System_TurnsignalLamp_composition.arxml
  - AUTOSAR
    - Components [ARPackage]
      - System_TurnsignalLamp [CompositionS

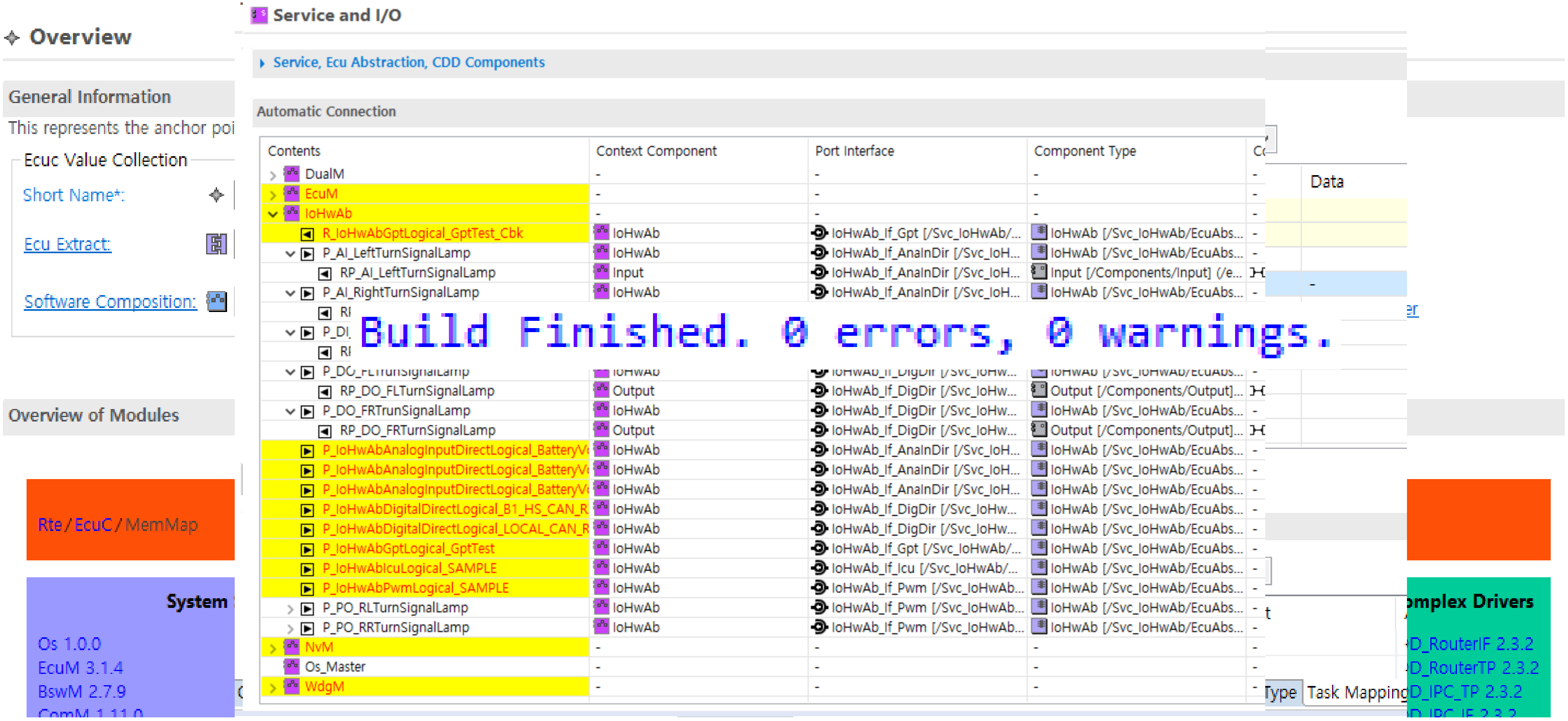
Components and Ports

Components

Component	Type
App_Lin	App_Lin [/ApplicationSwComponentTypes/App_Lin] (/e_gn7_pdc_as...
SWC_AppMode	SWC_AppMode [/AppMode/ApplicationSwComponentTypes/SWC_...
SWC_DiagnosticMonitor	SWC_DiagnosticMonitor [/AppSwComponent_Dem/SwComponent...
SWC_DiagnosticService	SWC_DiagnosticService [/AppSwComponent_Dcm/SwComponentT...
SWC_WdgMTest	SWC_WdgMTest [/App_WdgM/SWC_WdgMTest] (/e_gn7_pdc_asr_s...
System_TurnsignalLamp	System_TurnsignalLamp [/Components/System_TurnsignalLamp] (/...
Input	Input [/Components/Input] (/e_gn7_pdc_asr_swp_R230223_dev_r2/...
NvM_SWC	NvM_SWC [/Components/NvM_SWC] (/e_gn7_pdc_asr_swp_R2302...
Output	Output [/Components/Output] (/e_gn7_pdc_asr_swp_R230223_dev_...
TurnSignalLamp	TurnSignalLamp [/Components/TurnSignalLamp] (/e_gn7_pdc_asr_s...

Integration SW in mobilgene

- Just proceed with the mobilgene development process



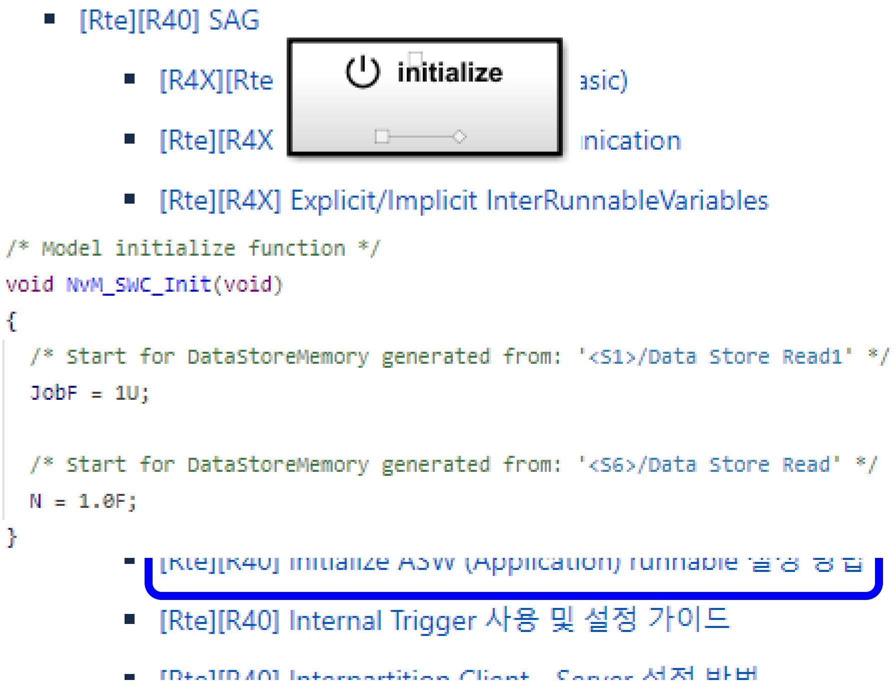
Automatic Connection

Contents	Context Component	Port Interface	Component Type	Cc
DualM	-	-	-	-
EcuM	-	-	-	-
IoHwAb	-	-	-	-
R_IoHwAbGptLogical_GptTest_Cbk	IoHwAb	IoHwAb_If_Gpt [/Svc_IoHwAb/...	IoHwAb [/Svc_IoHwAb/EcuAbs...	-
P_AI_LeftTurnSignalLamp	IoHwAb	IoHwAb_If_AnalnDir [/Svc_IoH...	IoHwAb [/Svc_IoHwAb/EcuAbs...	-
RP_AI_LeftTurnSignalLamp	Input	IoHwAb_If_AnalnDir [/Svc_IoH...	Input [/Components/Input] (/e...	⌶
P_AI_RightTurnSignalLamp	IoHwAb	IoHwAb_If_AnalnDir [/Svc_IoH...	IoHwAb [/Svc_IoHwAb/EcuAbs...	-
RI				
P_DI				
RI				
P_DO_FLTurnSignalLamp	IoHwAb	IoHwAb_If_DigDir [/Svc_IoHw...	IoHwAb [/Svc_IoHwAb/EcuAbs...	-
RP_DO_FLTurnSignalLamp	Output	IoHwAb_If_DigDir [/Svc_IoHw...	Output [/Components/Output]...	⌶
P_DO_FRTrunSignalLamp	IoHwAb	IoHwAb_If_DigDir [/Svc_IoHw...	IoHwAb [/Svc_IoHwAb/EcuAbs...	-
RP_DO_FRTurnSignalLamp	Output	IoHwAb_If_DigDir [/Svc_IoHw...	Output [/Components/Output]...	⌶
P_IoHwAbAnalogInputDirectLogical_BatteryV	IoHwAb	IoHwAb_If_AnalnDir [/Svc_IoH...	IoHwAb [/Svc_IoHwAb/EcuAbs...	-
P_IoHwAbAnalogInputDirectLogical_BatteryV	IoHwAb	IoHwAb_If_AnalnDir [/Svc_IoH...	IoHwAb [/Svc_IoHwAb/EcuAbs...	-
P_IoHwAbAnalogInputDirectLogical_BatteryV	IoHwAb	IoHwAb_If_AnalnDir [/Svc_IoH...	IoHwAb [/Svc_IoHwAb/EcuAbs...	-
P_IoHwAbDigitalDirectLogical_B1_HS_CAN_R	IoHwAb	IoHwAb_If_DigDir [/Svc_IoHw...	IoHwAb [/Svc_IoHwAb/EcuAbs...	-
P_IoHwAbDigitalDirectLogical_LOCAL_CAN_R	IoHwAb	IoHwAb_If_DigDir [/Svc_IoHw...	IoHwAb [/Svc_IoHwAb/EcuAbs...	-
P_IoHwAbGptLogical_GptTest	IoHwAb	IoHwAb_If_Gpt [/Svc_IoHwAb/...	IoHwAb [/Svc_IoHwAb/EcuAbs...	-
P_IoHwAbIcuLogical_SAMPLE	IoHwAb	IoHwAb_If_Icu [/Svc_IoHwAb/...	IoHwAb [/Svc_IoHwAb/EcuAbs...	-
P_IoHwAbPwmLogical_SAMPLE	IoHwAb	IoHwAb_If_Pwm [/Svc_IoHwAb...	IoHwAb [/Svc_IoHwAb/EcuAbs...	-
P_PO_RLTurnSignalLamp	IoHwAb	IoHwAb_If_Pwm [/Svc_IoHwAb...	IoHwAb [/Svc_IoHwAb/EcuAbs...	-
P_PO_RRTurnSignalLamp	IoHwAb	IoHwAb_If_Pwm [/Svc_IoHwAb...	IoHwAb [/Svc_IoHwAb/EcuAbs...	-
NvM	-	-	-	-
Os_Master	-	-	-	-
WdgM	-	-	-	-

```
Build Finished. 0 errors, 0 warnings.
```

Integration SW in mobilgene

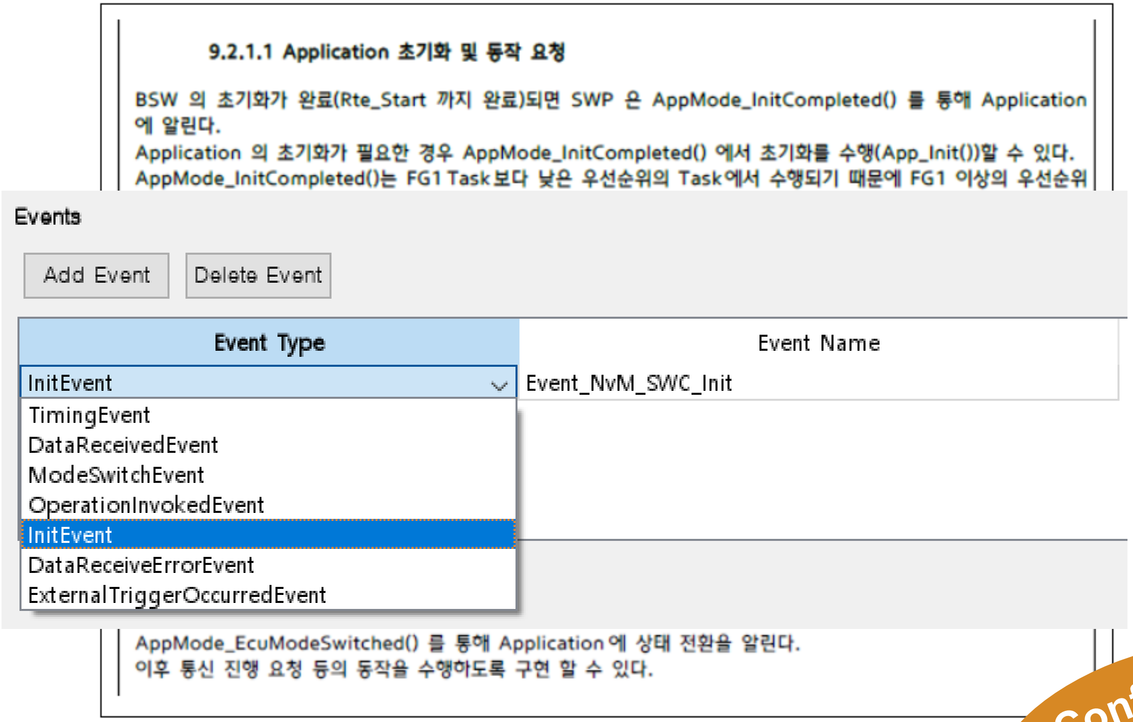
- Init Event for initialization function is not supported in mobilgene SWP 4.0.3
- Refer to the mobilgene FAQ to call initialization function of each application



- [Rte][R40] SAG
  - [R4X][Rte asic)
  - [Rte][R4X nication
  - [Rte][R4X] Explicit/Implicit InterRunnableVariables

```
/* Model initialize function */
void NvM_SWC_Init(void)
{
  /* Start for DataStoreMemory generated from: '<S1>/Data Store Read1' */
  JobF = 1U;

  /* Start for DataStoreMemory generated from: '<S6>/Data Store Read' */
  N = 1.0F;
}
```

  - [Rte][R40] Initialize ASW (Application) runnable 설정 방법
  - [Rte][R40] Internal Trigger 사용 및 설정 가이드
  - [Rte][R40] Interpartition Client - Server 설정 방법



9.2.1.1 Application 초기화 및 동작 요청

BSW 의 초기화가 완료(Rte_Start 까지 완료)되면 SWP 은 AppMode_InitCompleted() 를 통해 Application 에 알린다.
Application 의 초기화가 필요한 경우 AppMode_InitCompleted() 에서 초기화를 수행(App_Init())할 수 있다.
AppMode_InitCompleted()는 FG1 Task보다 낮은 우선순위의 Task에서 수행되기 때문에 FG1 이상의 우선순위

Events

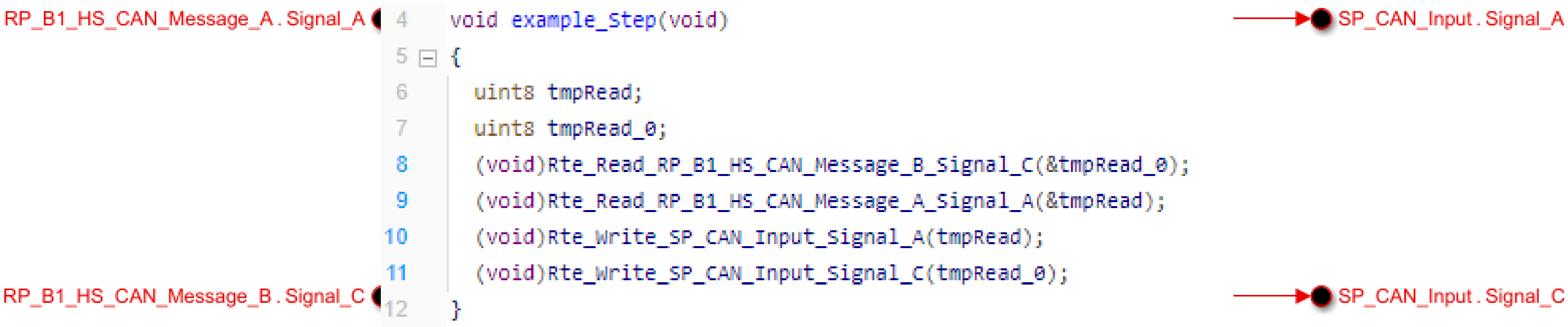
Event Type	Event Name
InitEvent	Event_NvM_SWC_Init

AppMode_EcuModeSwitched() 를 통해 Application 에 상태 전환을 알린다.
이후 통신 진행 요청 등의 동작을 수행하도록 구현 할 수 있다.

Conflict Issue

Useful modeling techniques

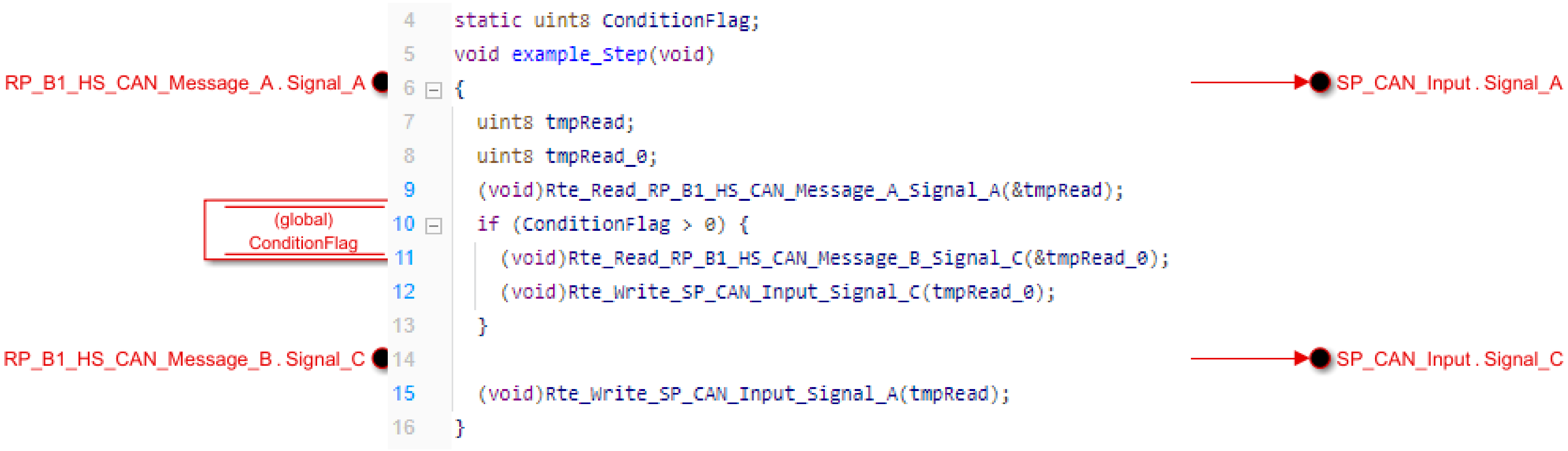
- Conditional RteRead/Write calls for general bus element data type


RP_B1_HS_CAN_Message_A.Signal_A
RP_B1_HS_CAN_Message_B.Signal_C
SP_CAN_Input.Signal_A
SP_CAN_Input.Signal_C

```
void example_Step(void)
{
  uint8 tmpRead;
  uint8 tmpRead_0;
  (void)Rte_Read_RP_B1_HS_CAN_Message_B_Signal_C(&tmpRead_0);
  (void)Rte_Read_RP_B1_HS_CAN_Message_A_Signal_A(&tmpRead);
  (void)Rte_Write_SP_CAN_Input_Signal_A(tmpRead);
  (void)Rte_Write_SP_CAN_Input_Signal_C(tmpRead_0);
}
```

Useful modeling techniques

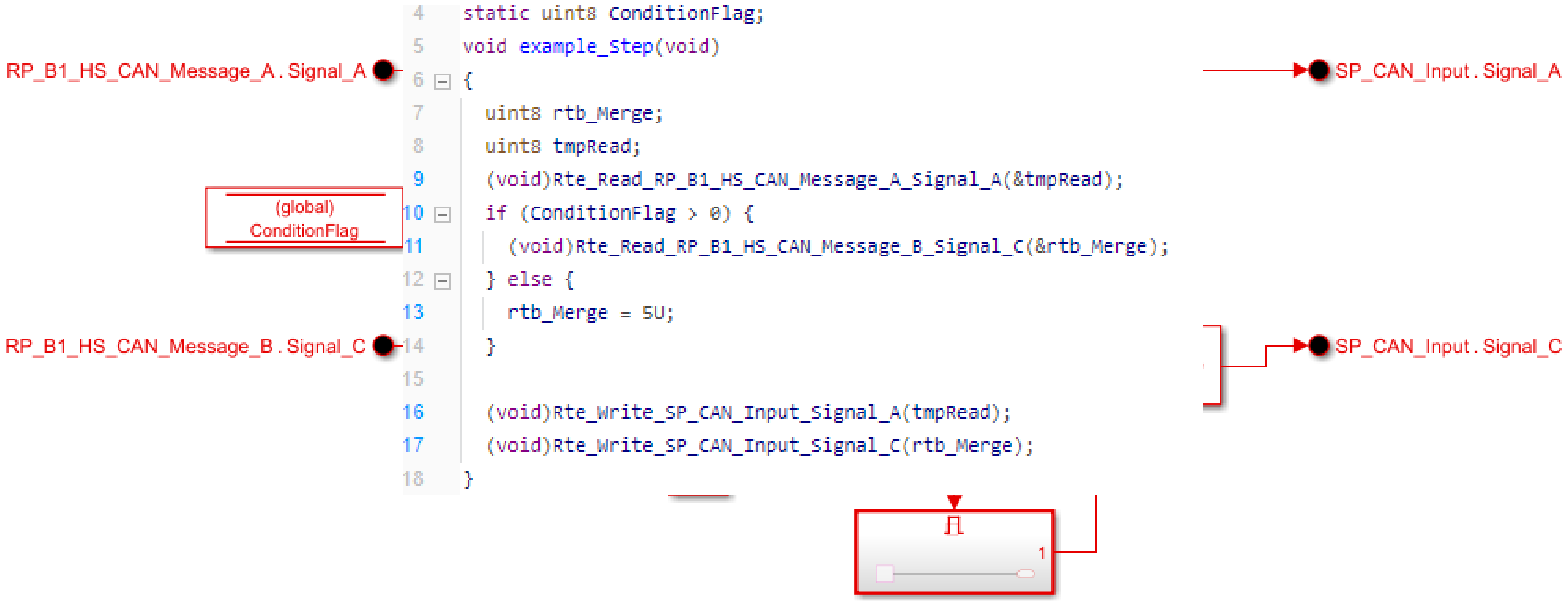
- Conditional RteRead/Write calls for general bus element data type



```
static uint8 ConditionFlag;
void example_Step(void)
{
  uint8 tmpRead;
  uint8 tmpRead_0;
  (void)Rte_Read_RP_B1_HS_CAN_Message_A_Signal_A(&tmpRead);
  if (ConditionFlag > 0) {
    (void)Rte_Read_RP_B1_HS_CAN_Message_B_Signal_C(&tmpRead_0);
    (void)Rte_Write_SP_CAN_Input_Signal_C(tmpRead_0);
  }

  (void)Rte_Write_SP_CAN_Input_Signal_A(tmpRead);
}
```

Useful modeling techniques

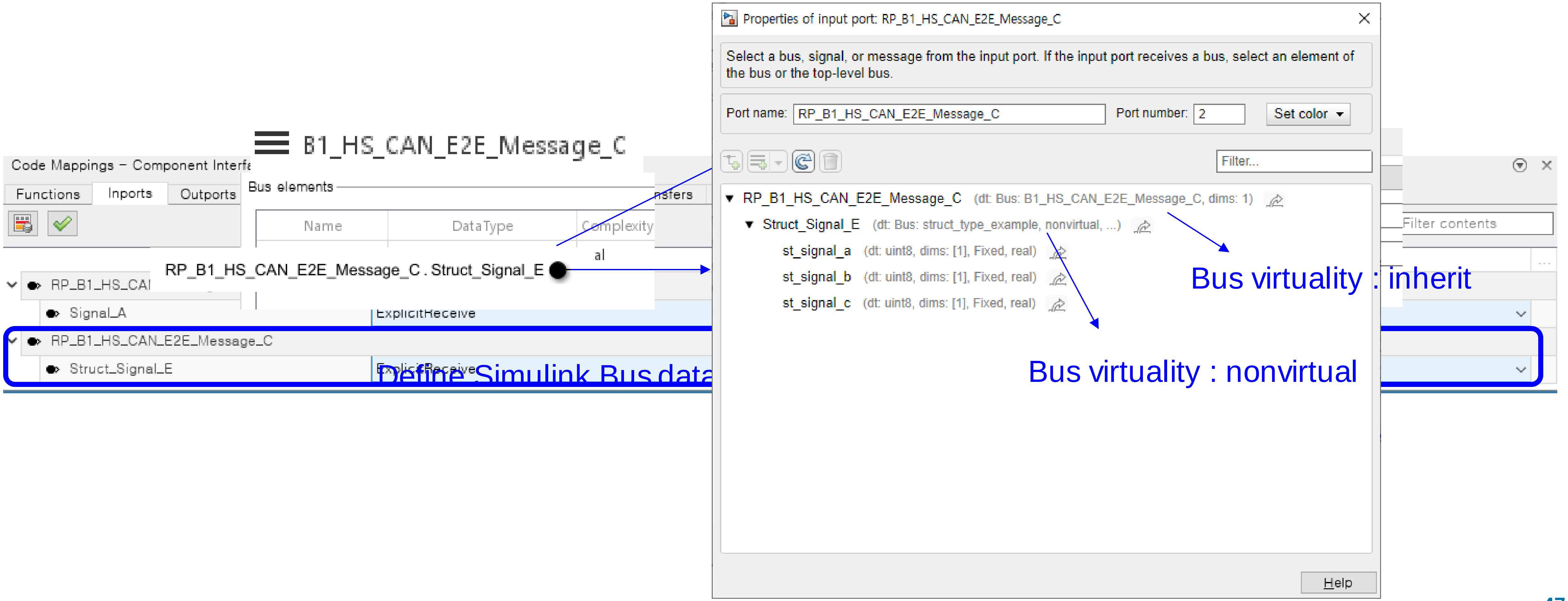
- Conditional RteRead/Write calls for general bus element data type



```
static uint8 ConditionFlag;
void example_Step(void)
{
  uint8 rtb_Merge;
  uint8 tmpRead;
  (void)Rte_Read_RP_B1_HS_CAN_Message_A_Signal_A(&tmpRead);
  if (ConditionFlag > 0) {
    (void)Rte_Read_RP_B1_HS_CAN_Message_B_Signal_C(&rtb_Merge);
  } else {
    rtb_Merge = 5U;
  }

  (void)Rte_Write_SP_CAN_Input_Signal_A(tmpRead);
  (void)Rte_Write_SP_CAN_Input_Signal_C(rtb_Merge);
}
```

Useful modeling techniques

- Conditional RteRead/Write calls for bus element data type struct



Properties of input port: RP_B1_HS_CAN_E2E_Message_C

Select a bus, signal, or message from the input port. If the input port receives a bus, select an element of the bus or the top-level bus.

Port name: RP_B1_HS_CAN_E2E_Message_C Port number: 2 Set color

- RP_B1_HS_CAN_E2E_Message_C (dt: Bus: B1_HS_CAN_E2E_Message_C, dims: 1)
  - Struct_Signal_E (dt: Bus: struct_type_example, nonvirtual, ...)
    - st_signal_a (dt: uint8, dims: [1], Fixed, real)
    - st_signal_b (dt: uint8, dims: [1], Fixed, real)
    - st_signal_c (dt: uint8, dims: [1], Fixed, real)

Bus virtuality : inherit

Bus virtuality : nonvirtual

B1_HS_CAN_E2E_Message_C

Bus elements

Name	DataType	Complexity
		al

RP_B1_HS_CAN_E2E_Message_C . Struct_Signal_E

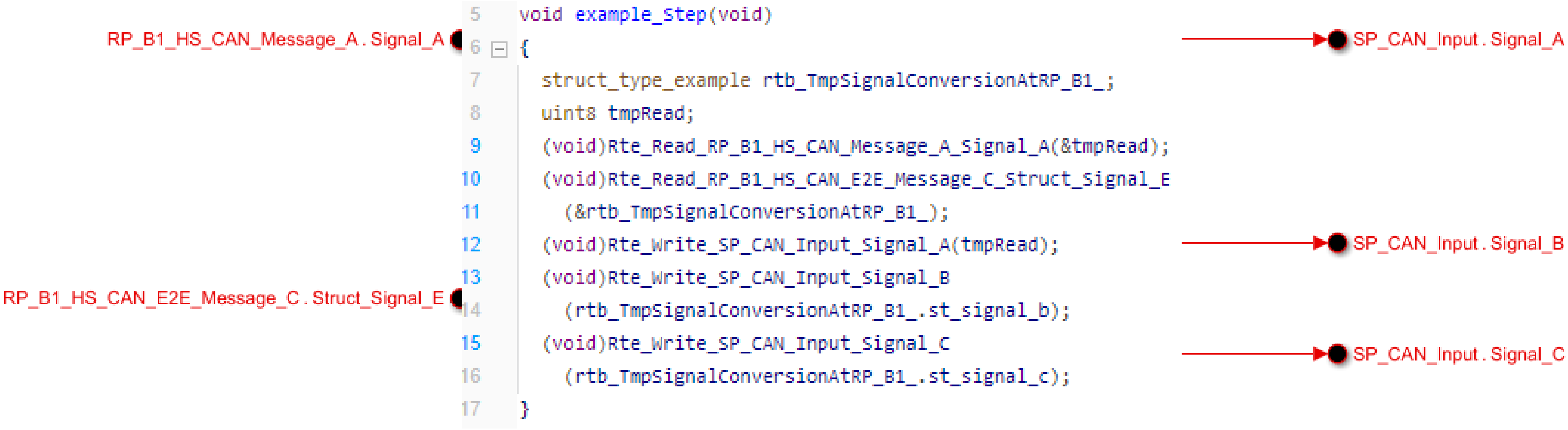
Code Mappings – Component Interfa

RP_B1_HS_CAI	
Signal_A	ExplicitReceive
RP_B1_HS_CAN_E2E_Message_C	
Struct_Signal_E	ExplicitReceive

Define Simulink Bus data

Useful modeling techniques

- Conditional RteRead/Write calls for bus element data type struct

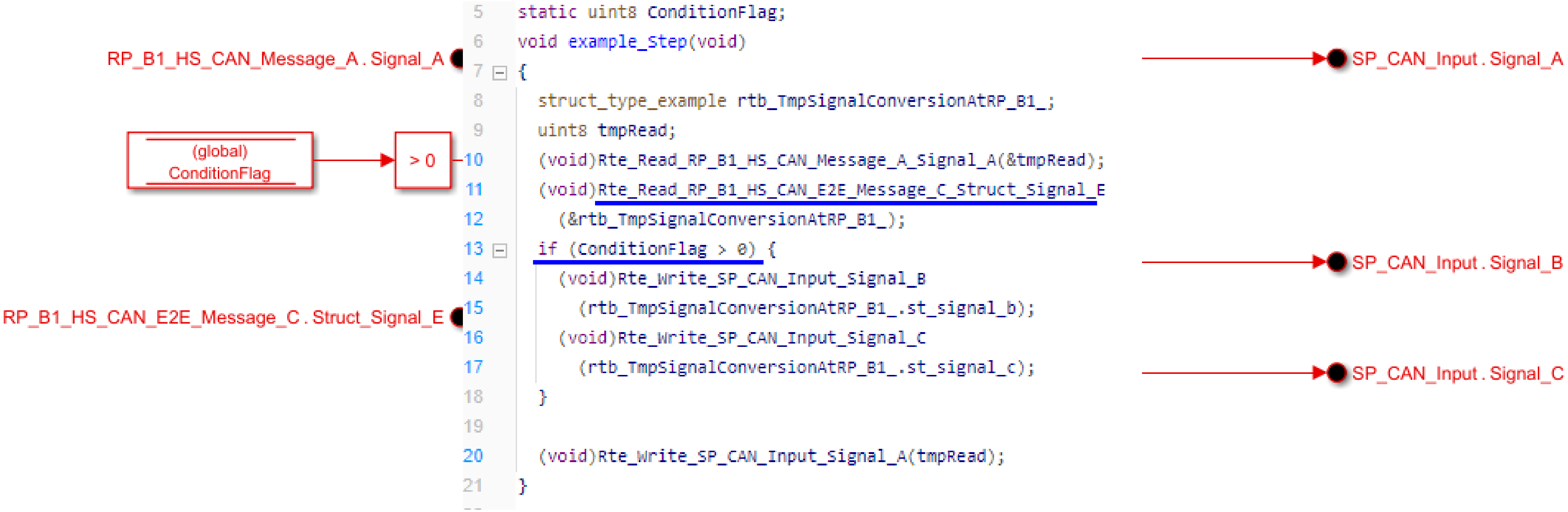


```
void example_Step(void)
{
  struct_type_example rtb_TmpSignalConversionAtRP_B1_;
  uint8 tmpRead;
  (void)Rte_Read_RP_B1_HS_CAN_Message_A_Signal_A(&tmpRead);
  (void)Rte_Read_RP_B1_HS_CAN_E2E_Message_C_Struct_Signal_E
    (&rtb_TmpSignalConversionAtRP_B1_);
  (void)Rte_Write_SP_CAN_Input_Signal_A(tmpRead);
  (void)Rte_Write_SP_CAN_Input_Signal_B
    (rtb_TmpSignalConversionAtRP_B1_.st_signal_b);
  (void)Rte_Write_SP_CAN_Input_Signal_C
    (rtb_TmpSignalConversionAtRP_B1_.st_signal_c);
}
```

RP_B1_HS_CAN_Message_A . Signal_A
RP_B1_HS_CAN_E2E_Message_C . Struct_Signal_E
SP_CAN_Input . Signal_A
SP_CAN_Input . Signal_B
SP_CAN_Input . Signal_C

Useful modeling techniques

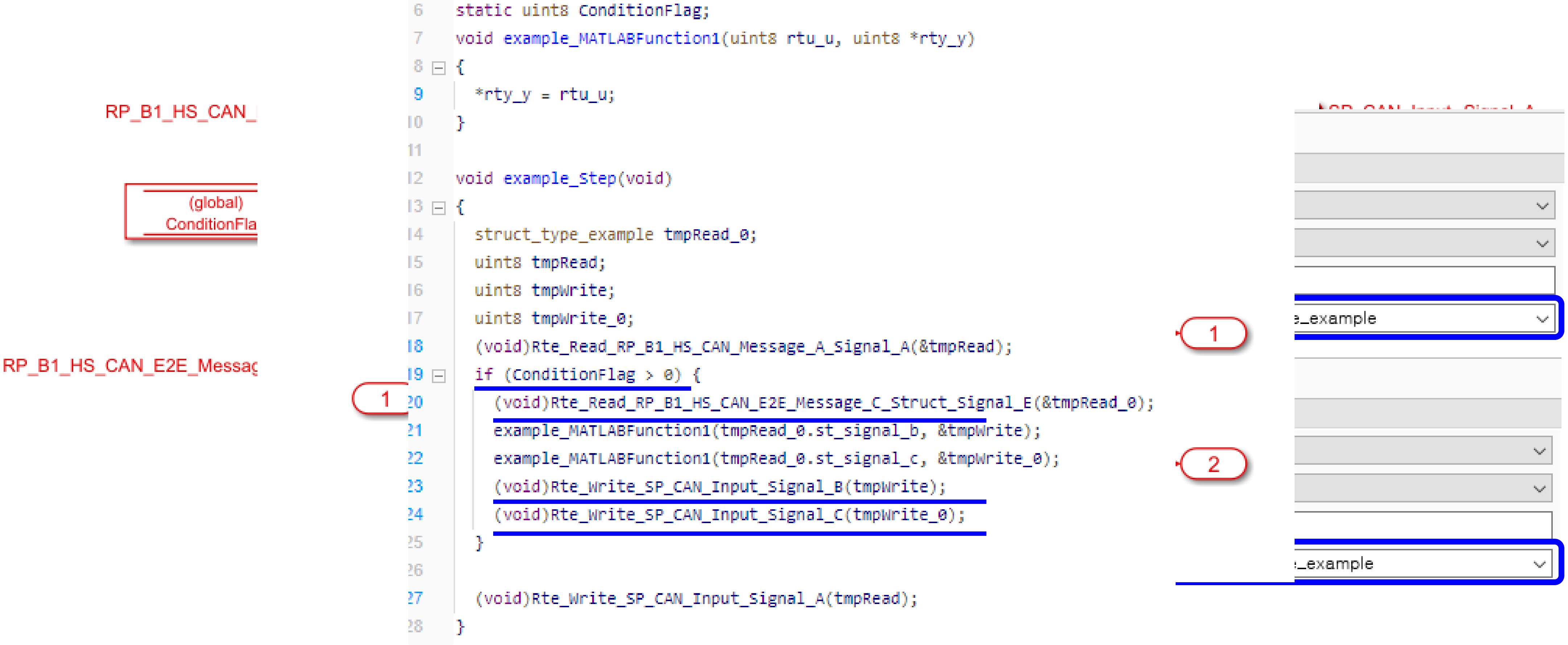
- Conditional RteRead/Write calls for bus element data type struct



```
static uint8 ConditionFlag;
void example_Step(void)
{
  struct_type_example rtb_TmpSignalConversionAtRP_B1_;
  uint8 tmpRead;
  (void)Rte_Read_RP_B1_HS_CAN_Message_A_Signal_A(&tmpRead);
  (void)Rte_Read_RP_B1_HS_CAN_E2E_Message_C_Struct_Signal_E
    (&rtb_TmpSignalConversionAtRP_B1_);
  if (ConditionFlag > 0) {
    (void)Rte_Write_SP_CAN_Input_Signal_B
      (rtb_TmpSignalConversionAtRP_B1_.st_signal_b);
    (void)Rte_Write_SP_CAN_Input_Signal_C
      (rtb_TmpSignalConversionAtRP_B1_.st_signal_c);
  }

  (void)Rte_Write_SP_CAN_Input_Signal_A(tmpRead);
}
```

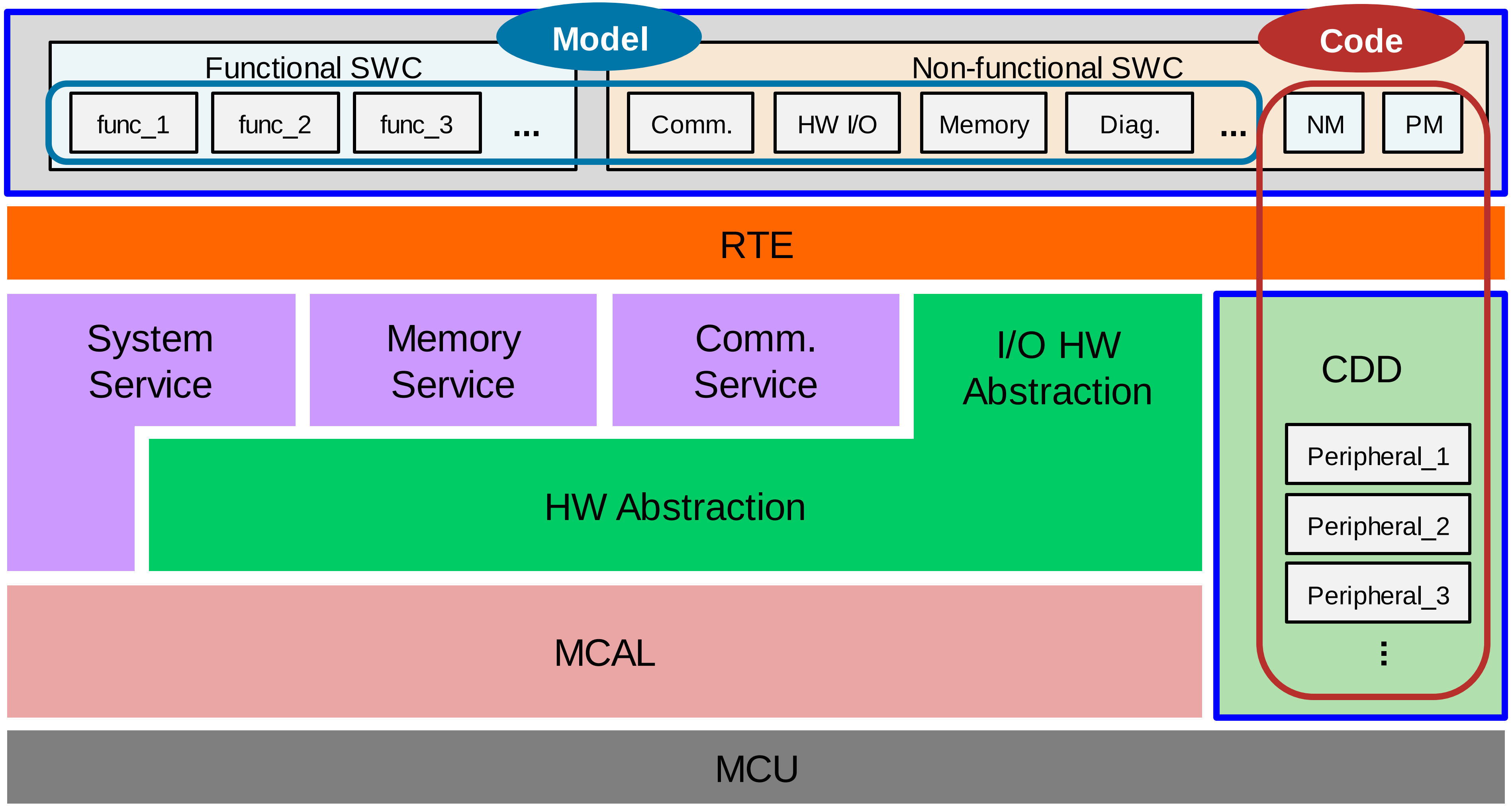
Useful modeling techniques

- Conditional RteRead/Write calls for bus element data type struct



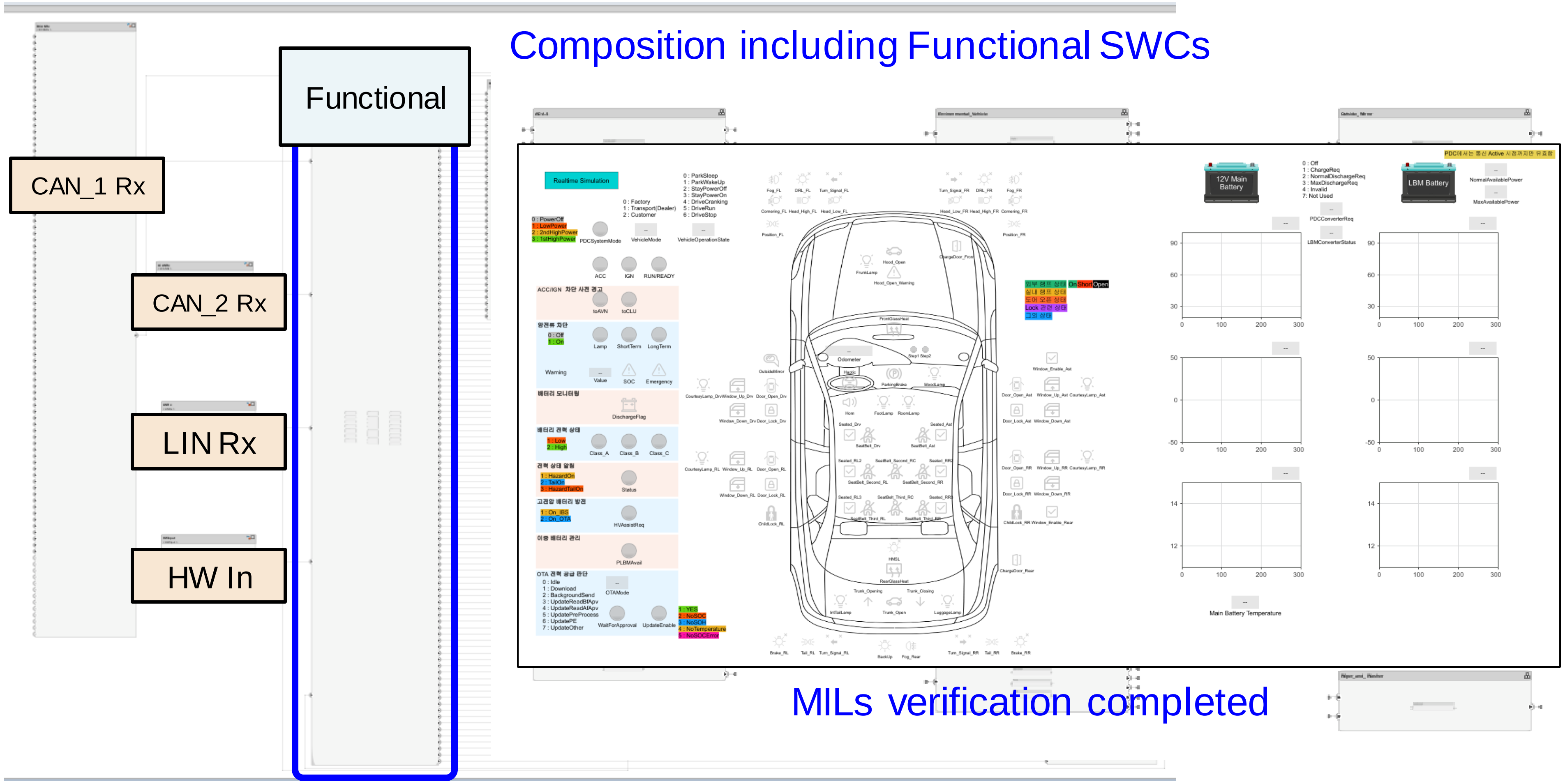
```
static uint8 ConditionFlag;
void example_MATLABFunction1(uint8 rtu_u, uint8 *rty_y)
{
  *rty_y = rtu_u;
}

void example_Step(void)
{
  struct_type_example tmpRead_0;
  uint8 tmpRead;
  uint8 tmpWrite;
  uint8 tmpWrite_0;
  (void)Rte_Read_RP_B1_HS_CAN_Message_A_Signal_A(&tmpRead);
  if (ConditionFlag > 0) {
    (void)Rte_Read_RP_B1_HS_CAN_E2E_Message_C_Struct_Signal_E(&tmpRead_0);
    example_MATLABFunction1(tmpRead_0.st_signal_b, &tmpWrite);
    example_MATLABFunction1(tmpRead_0.st_signal_c, &tmpWrite_0);
    (void)Rte_Write_SP_CAN_Input_Signal_B(tmpWrite);
    (void)Rte_Write_SP_CAN_Input_Signal_C(tmpWrite_0);
  }

  (void)Rte_Write_SP_CAN_Input_Signal_A(tmpRead);
}
```

Our use case


Model
Code
Functional SWC
func_1
func_2
func_3
...
Non-functional SWC
Comm.
HW I/O
Memory
Diag.
...
NM
PM
RTE
System Service
Memory Service
Comm. Service
I/O HW Abstraction
HW Abstraction
CDD
Peripheral_1
Peripheral_2
Peripheral_3
⋮
MCAL
MCU

Our use case

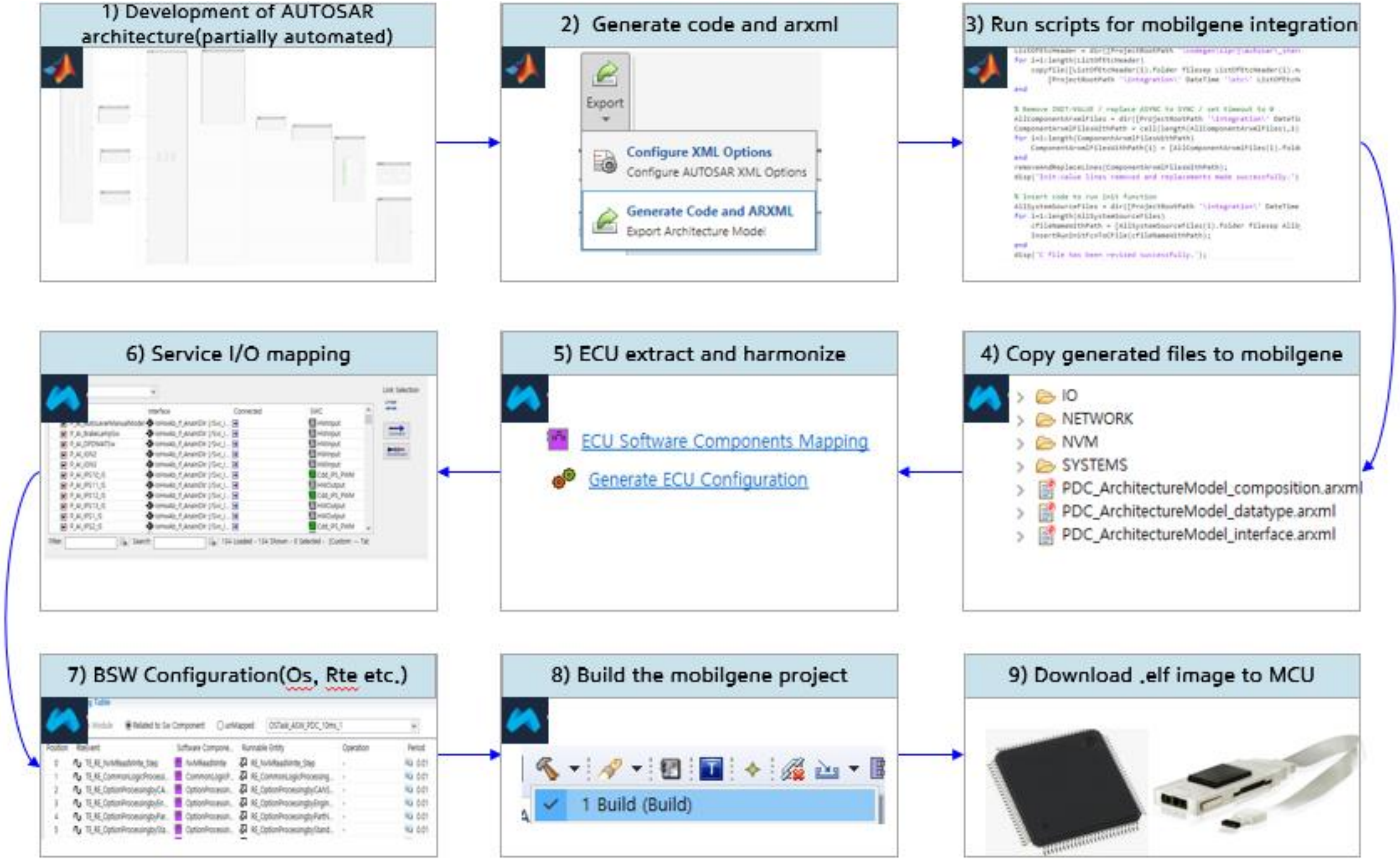
- Modeling application layer with AUTOSAR Blockset



Composition including Functional SWCs

MILs verification completed

Our use case



1) Development of AUTOSAR architecture(partially automated)

2) Generate code and arxml

3) Run scripts for mobilgene integration

```
ListOfEtcHeader = dir([ProjectRootPath '\codegen\lpr\autosar\_shar
for i=1:length(ListOfEtcHeader)
    copyfile([ListOfEtcHeader(i).folder filesep ListOfEtcHeader(i).n
        [ProjectRootPath '\integration\' DateTime '\etc\' ListOfEtcH
end

% Remove INIT-VALUE / replace ASYNC to SYNC / set timeout to 0
AllComponentArxmlFiles = dir([ProjectRootPath '\integration\' DateTi
ComponentArxmlFilesWithPath = cell(length(AllComponentArxmlFiles),1)
for i=1:length(ComponentArxmlFilesWithPath)
    ComponentArxmlFilesWithPath{i} = [AllComponentArxmlFiles(i).fold
end
removeAndReplaceLines(ComponentArxmlFilesWithPath);
disp('Init-value lines removed and replacements made successfully.')

% Insert code to run Init function
AllSystemSourceFiles = dir([ProjectRootPath '\integration\' DateTime
for i=1:length(AllSystemSourceFiles)
    cFileNameWithPath = [AllSystemSourceFiles(i).folder filesep AllS
    InsertRunInitFcnToCFile(cFileNameWithPath);
end
disp('C file has been revised successfully.');
```

4) Copy generated files to mobilgene

IO
NETWORK
NVM
SYSTEMS
PDC_ArchitectureModel_composition.arxml
PDC_ArchitectureModel_datatype.arxml
PDC_ArchitectureModel_interface.arxml

5) ECU extract and harmonize

6) Service I/O mapping

7) BSW Configuration(Os, Rte etc.)

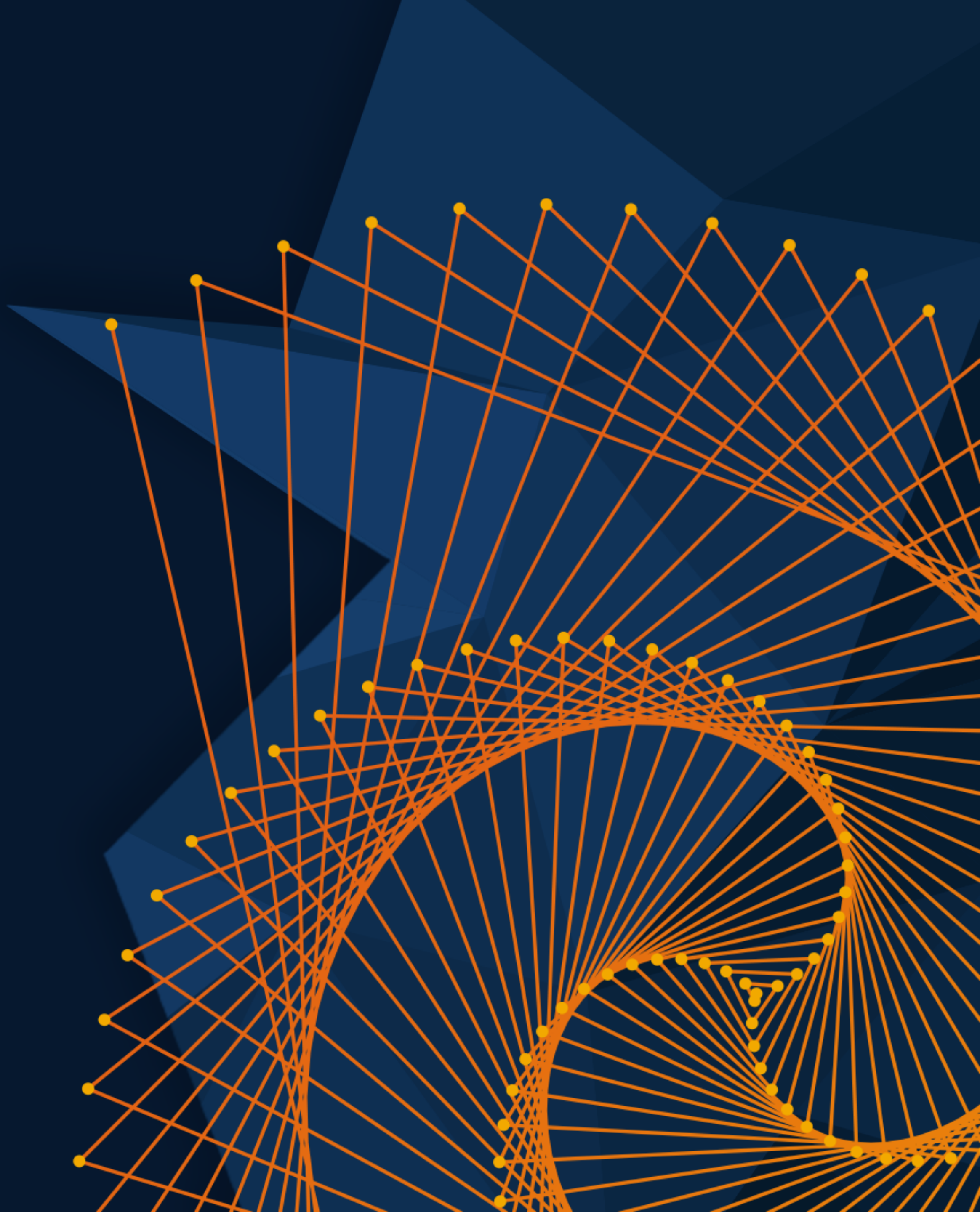
8) Build the mobilgene project

9) Download .elf image to MCU

Our use case

- Testing the SW in HILs and in vehicle level and developing new/updated functions or features
- Planning to do the same works with mobilgene SWP 4.4.0

MATLAB EXPO



MathWorks®