



B

7 December 2021

Agenda Enter your subtitle here

> 01 Project workflow

02 Requirements and architecture of the workflow demonstrator

03 EB tresos workflow 04 AUTOSAR Builder (AB) workflow

05

07

EB tresos & AUTOSAR Builder (AB) workflow 06 EB tresos AutoCore OS & RTE

Generate source code and compile





B

Project workflow



Requirements and architecture of the workflow demonstrator

Requirements and architecture of the workflow demonstrator

Requirements

• What shall the software do?

1. Output & state handling

- The software shall output an increasing counter if it is in state RUN.
- The software shall reset if it is switched to state Reset.

2. Input

- The software shall use keyboard inputs for controlling.
- Only chosen characters and numbers must be used.
- Prevent long key presses for multiple controlling.

3. Error handling

- It shall be possible to detect errors during development.

4. Reset

- Must be possible to do a soft reset.
- Must be possible to do a hard reset.

5. Storage

- The software must have the possibility to read ROM values.
- The software must have the possibility to write to NvM during runtime.
- The software must write to non-volatile-memory during soft reset (write-all).

6. Startup

- The software shall use the AUTOSAR way of startup.
- The software must read the last stored entry from non-volatile memory during startup.
- The software components must change the state machine from Init->Run.

Tool requirements of the workflow demonstrator

Versions

Dassault AUTOSAR Builder version

• Version 2021x

Disclaimer

This published tool workflow corresponds to best practice and might not be the unique way to use the tools.

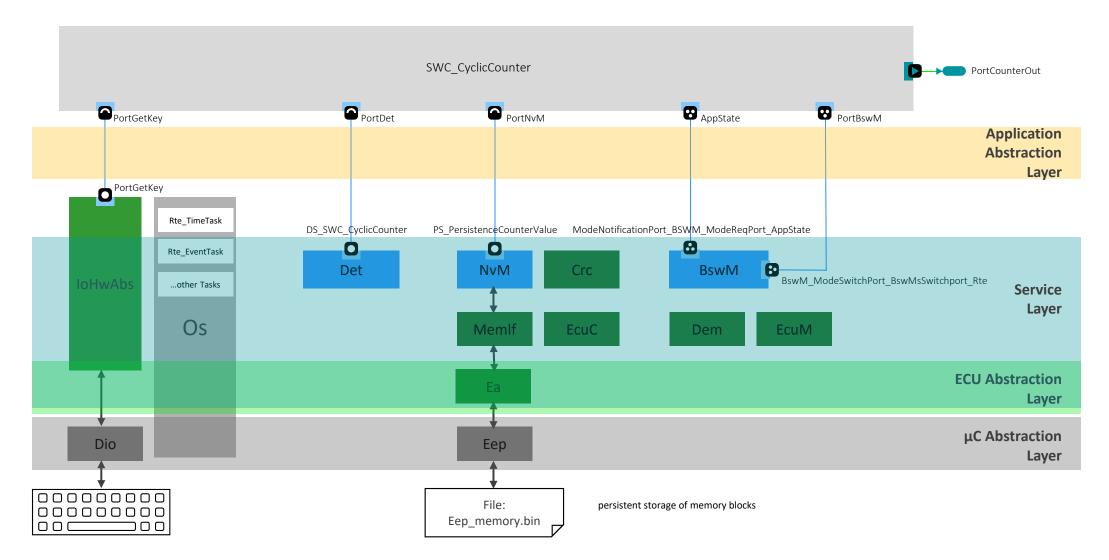
EB tresos Studio version

• Version 28.1.0

EB tresos AutoCore Generic version

• Version 8.8.3

Project architecture





B

EB tresos workflow

Create new EB tresos Studio project

1. Start EB tresos Studio

- Run "Start_Tresos_new.bat" and click



2. Create Project

- − Open the "Workflows" tab "Window \rightarrow Show view \rightarrow Workflows".
- Follow the "Create your project" instructions:
 - 1. Create new project (from Template)
 - 2. In the text box "Select root directory", browse to the project template folder *basicTemplate*.

Hint: This folder is located in the EB tresos AutoCore derivative-specific subdirectory in your EB tresos Studio installation.

- 3. Select the check box "**Copy projects into workspace**" (Otherwise you will modify the template itself).
- 4. Click Finish.

Create new project (from Template)

🔢 Import		_		×
Import Projects Select a directory to sear	ch for existing Eclipse projects.			7
Select root directory:	C:\EB\eb_tresos_demo\tools\tresos\templates	~	Browse	
\bigcirc Select archive file:		\sim	Browse	
Projects:				
basicTemplate (C	:\EB\eb_tresos_demo\tools\tresos\templates\AutoCore\8.8.2-WIN32	X86	Select A	AII
			Deselect	All
			Refres	h
K		>		
Options		-		
Search for nested pro	jects			
Copy projects into w	· · ·			
	eady exist in the workspace			
Working sets			New	
Add project to work	ing sets			
Working sets:			Select	
?	Finish		Cance	I

Rename, load & modules

1. Rename

- Mark the project in the "Project Exlorer" and press F2 (right-click \rightarrow rename)
- Rename your project to "MyDemo_AB_Tresos".

2. Load

- Double-click the project in the "Project Explorer".

3. Modules

- The Basic Template contains several plugins that are necessary for creating our project:
 - Base: Standard header file for EB treosos AutoCore
 - *EcuM: Mode management (without OS support)*
 - BswM: Mode management (with OS support)
 - Det: Development Error Tracer
 - MemMap: Map code and data to specific memory sections
 - OS: Operating System
 - RTE: Runtime Environment (Application \rightarrow Basis Software Abstraction)

Project Explorer 🛛
✓ Image: Value of the value
V 🥔 ECU (WINDOWS, WIN32X86)
> 👺 Atomics (V 1.0. 13, AS4.0.3)
> 👺 Base (V5.0.30, AS4.0.3)
> 👺 BswM (V1.15.5, AS4.0.3)
> 👺 Compiler (V1.0.6, AS4.0.3)
> 👺 Dem (V6.4.0, AS4.0.3)
> 👺 Det (V6.5.5, AS4.0.3)
> 👺 EcuM (V5.15.6, AS4.0.3)
> 👺 Make (V4.0.26, AS4.0.3)
> 👺 Mcu (V4.0.3, AS4.0.3)
> 👺 MemMap (V1.3.8, AS4.0.3)
> 👺 Os (V6.0.302, AS4.0.3)
> 👺 PbcfgM (V1.2.22, AS4.0.3)
> 👺 Platforms (V4.0.0, AS4.0.3)
> 👺 Rte (V6.4.11, AS4.0.3)
> 🗁 config
> 🗁 output
> 🗁 source
> 🗁 system
> 📂 util

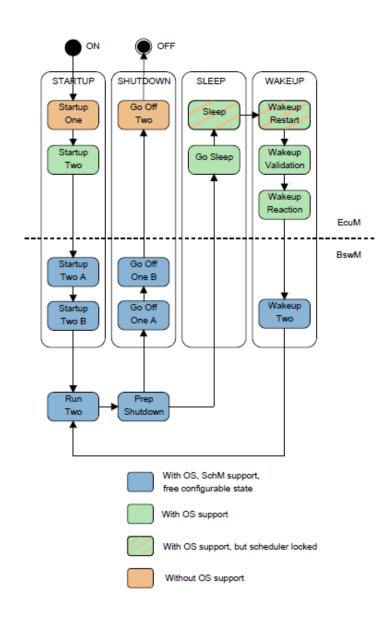
EcuM and BswM background¹

EcuM and BswM are responsible for:

- 1. Startup behavior of an ECU
- 2. State management of an ECU
- 3. Shutdown behavior of an ECU
- EB tresos AutoCore provides the "flexible ECU management" variant of the **EcuM** module.
- This variant of **EcuM** module handles the early STARTUP phases, the late SHUTDOWN phases and the SLEEP phases.
- The whole OS scheduling is locked by the EcuM taking the OS resource RES_SCHEDULER.
- All other phases and states are completely configurable and handled by the **BswM** module as depicted.
- The states of the **BswM** module are completely configurable and can be adapted to system needs.

Important: On event **RUN_TWO** the RTE is started and the SWC can be initialized.

1 See AutoCore_Generic_Mode_Management_documentation.pdf for detailed information



Generate BSW & SWC description

1. Generate SWCD

- Run "Project → Build Project → generate_swcd"
- The output\generated\swcd folder contains all relevant basic software description (*_Bswmd.arxml) files of the configured modules

2. Generate BSW

- − Run "Project → Generate Project"
- The output \generated \ folder contains all necessary files to build the project.
 - src/include: Source and header files
 - make: makefiles
 - **orti**: ORTI stands for "OSEK Run Time Interface" and has been designed to facilitate an interface between the internal operation of an OSEK operating system and a debugger.
 - templates: template code for BSW and SWC software
 - xgen: variant handling

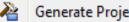
File Edit Search Project Window He	lelp RTE
📸 👻 🔛 🐨 🚺 🖬 🔂 🖓 📢 👘 🖓 Generate All Proje	ects 🗢 ▾ ➪ ▾ 🛛 ϫ
Project Explorer 🖌 Verify Project	
✓ ✓ ✓ MyDemo_At Generate Project	_
V CU (WIN L Build Project	> NG_PostfixAndMove
> 💁 Atom 🎲 Unattended Wizar	rds > Template_withPostRename
> 🕮 Base	contract
>	export_asr_config
>	generate_asr32_swcd
> 👺 Det (V6.5.5, AS4.0.3)	generate_asr40_swcd
> 🚇 EcuM (V5.15.6, AS4.0.3)	generate_project_data
>	generate_swcd
Mcu (v4.0.5, A54.0.5) MemMap (V1.3.8, AS4.0.3)	verify_asr32_swcd
> 🔮 Os (V6.0.302, AS4.0.3)	verify_asr40_swcd
> 🚇 PbcfgM (V1.2.22, AS4.0.3)	verify_swcd
> 🚇 Platforms (V4.0.0, AS4.0.3)	
> 👺 Rte (V6.4.11, AS4.0.3)	
> 🗁 config 🗸 🗁 output	
✓ i generated	
🗸 🗁 swcd	
Atomics_Bswmd.arxm	1
Base_Bswmd.arxml	
BswM_Bswmd.arxml	
Det_Bswmd.arxml	
EcuM_Bswmd.arxml	
PbcfgM_Bswmd.arxml	
✓ ➢ swcd_colibration	
Dem_Bswmd.arxml	
> 🗁 source > 🗁 system	
, System	

EB tresos Studio workflow

Until now you have accomplished the following:

1. You created a project in EB tresos Studio which is based on a pre-configured template.

2. You generated the configuration-dependent source code for each BSW module which is part of the project.



Generate Project

The generated files are located in the folder [project]\output\generated.

3. You generated the basic software description files (*_bswmd.arxml).

generate_swcd

The generated files are located in the folder [project]\output\generated\swcd.

Note: The files are not required at this point of time but will become important when creating our software component with AUTOSAR Builder.





AUTOSAR Builder (AB) workflow

AUTOSAR Builder workflow

Create new AB project

1. Start AUTOSAR Builder

- Run "Start_AB.bat"

2. Create Project

Click "File → New → AUTOSAR Project"
 GB ab_workspace - Authoring Environment - AUTOSAR Builder 2021x - C:\EB\eb_tresos_demo\workspaces\
 File Edit Navigate Search Project ASim Plug-let Tools Run AB Variant Window Help

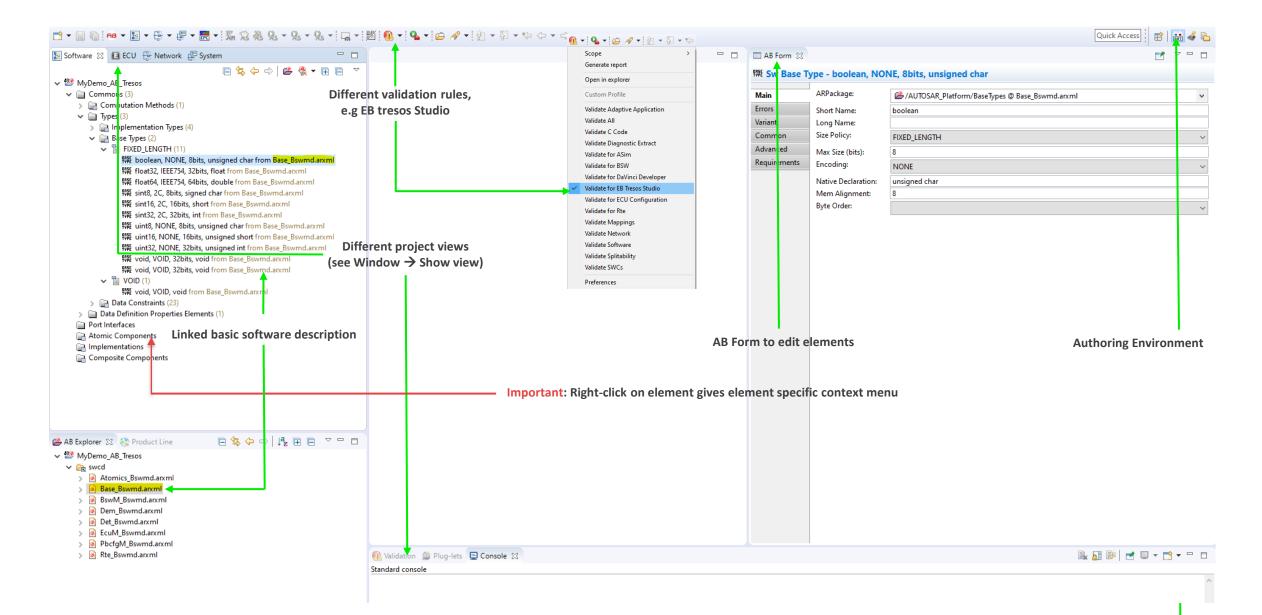
		-			· ·
	New	Alt+Shift+N >	RAT	AUTOSAR Project	
	Open File		A	Simulation Environment	
\sim	Open Projects from File System			Project	

- Enter Project name: "MyDemo_AB_Tresos"
- Import existing EB tresos swcd folder (uncheck "Copy folder in project")
 - This will **link** to the tresos workspace instead of copying the files.
- Click Finish.

Create new AUTOSAR project

RB New AUTOSAR Project					×
AUTOSAR Project Create a new AUTOSAR proje	ct.				0
Project name: MyDemo_AB	_Tresos				
Use default location					
Location: C:\EB\eb_tresos_d	lemo\workspaces\ab	_workspace\MyDemo_AB_Tresos		Browse	e
AUTOSAR Release options Workspace default Release			Configure Wor	kspace Sett	tinas
Alternate Release		AUTOSAR 00048 (http://autosar.org/schema/r4.0)	contigure troi	<u>Ropure sea</u>	~
Import User Library (Conf Classic Platform Import ECU Configurati Import Application Inte Import Standard and Pli Adaptive Platform Import Adaptive Standa	on Parameters rfaces atform Types	{ Builder preferences)			
AUTOSAR Resources O Create empty project O Create new file, named:	default.arxml				
O Import existing file:	Copy file in proj	ect		Browse	
Import existing folder:	C:\EB\eb_tresos_d	emo\workspaces\tresos_workspace\MyDemo_AB_Tresos\output\generated\swcd roject		Browse	t
?			Finish	Cano	cel

AUTOSAR Builder workflow



Create new SWC

- Lock (read-only) EB tresos swcd folder
 To make our life easier, we lock the swcd folder from our EB tresos workspace:
 - − Right-click on AB Explorer \rightarrow swcd \rightarrow Properties
 - Check "Read-only"
 - Hint: Do not lock files from EB tresos swcd folder called, because these files are configuration-dependent.

🖊 💯 MyDemo AB Tresos

> Commons (5)

> i Port Interfaces (

> 🔚 Atomic Compor

> Implementation

> 📄 Composite Com

New

Go Into

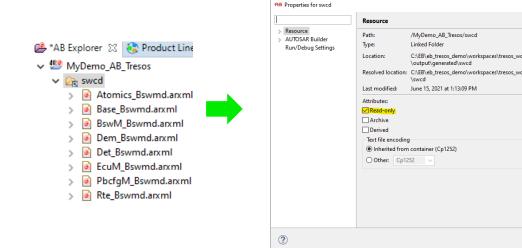
AB Browsers

AB Tables

- XXX_swc_interface.arxml
- XXX_swc_internal.arxml

2. Create new AUTOSAR File

- Right-click on Project (or AB Explorer) → new → AUTOSAR File.
- File name: SWC_CyclicCounter.arxml
- ARPackage: Demo
- The new file will be shown in the AB Explorer.
- Due to the following steps, the new file <u>should</u> be in the parent folder MyDemo_AB_Tresos.



🔗 ARPackage

> AUTOSAR Project

📫 File

AUTOSAR File

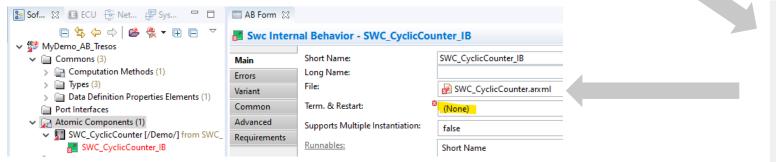
🔗 AB Explorer 🐹 🛛 🍋 Product Line

> SWC_CyclicCounter.arxm

	AUTOSAR File		_	\diamond
	Create a new AUTOSAR file.			
	Enter or select the parent folder:			
	MyDemo_AB_Tresos			
	 ☆ ↔ ★ MyDemo_AB_Tresos 			
	File name: SWC_CyclicCounteraryml			
•	File name: SWC_CyclicCounteraryml			
•				
•	Advanced >>			
•	Advanced >> ARPackage creation			

Create new SWC

- 3. Create new SWC
 - Right-click on "Atomic Components".
 - New \rightarrow Application Component Types
 - Short Name: SWC_CyclicCounter
 - The ports can be added afterwards as well.



Atomic Components

Go Into

Instantiation Wizard

AB Browsers

AB Tables

AB Tools

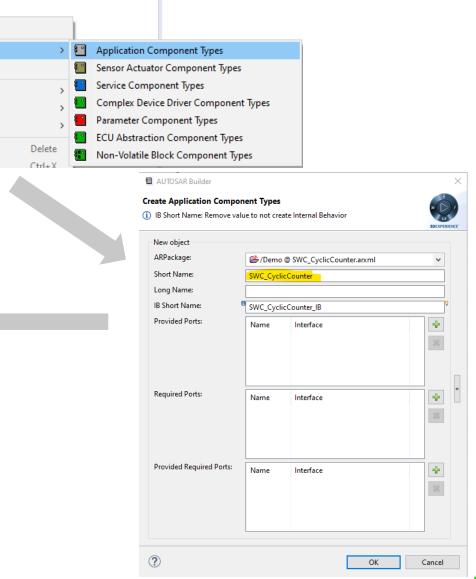
X Delete

& Cut

New

📄 Implemen

Composit

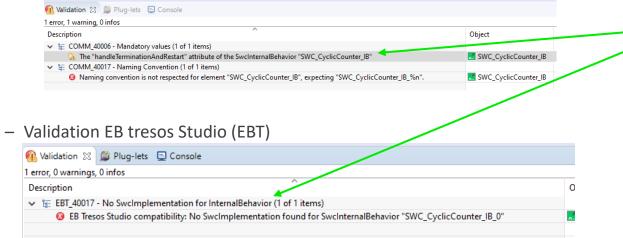


Validation example

- 1. After creation we get some errors & warnings that we have to fix.
- 2. Validate Subtree, not entire project

	7		'		,		 SWC_Cycl SWC (<u></u>	Go Into			Advanced
•	<mark>-</mark> 🎭 - 😥 🛷 - 😫 - 🏹 - 🧐	¢	• 🖒 •				Implementation Composite Co	÷ 100	New Change Type of SWC	>		Simulink Requirements
	Scope >		Project				mpleDemo_AB_		AB Browsers	>		
	Generate report	\checkmark	Subtree						AB Tables AB Tools	>		
	Open in explorer		Selection			OR			AB Validation		Project	Ctrl+Shift+v
	Curtom Brofile		Selection and Relat	ted					Diagrams Element to Replicate		k Related O	·
								Q	RTE Generator - Contract Phase (requires RTEC license)		Subtree	

- 3. Validation is based on the validation rules currently used: | \mathbf{M} \mathbf{v}
 - Validation SWCs



- Double-click guides you to the validation issue
- Change Short name of Internal Behavior SWC_CyclicCounter_IB → SWC_CyclicCounter_IB_0
- 2. Term. & Restart: NO-SUPPORT

SWC implementation

1. Add new SWC implementation

- a. Right-click "Implementations \rightarrow new \rightarrow SWC Implementation"
- b. Important: Check the correct ARPackage: SWC_CyclicCounter.arxml
- c. Add Short Name: Impl_SWC_CyclicCounter
- d. Internal Behavior: SWC_CyclicCounter_IB_0
- e. Add Code Descriptors: SRC
- f. Add Engineering Object: EngObj
 - File Type: SWSRC

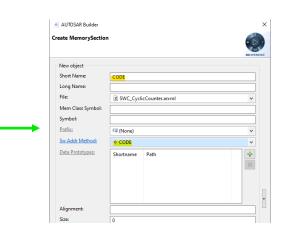
2. Add new ResourceConsumptions

- a. Right-click on "ResourceConsumption \rightarrow new \rightarrow Memory Section"
- b. Short Name: CODE
- c. Sw Addr Methode: CODE

3. Validation

If you validate the subtree again, there should be no EB tresos errors anymore.

	AUTOSAR Builder						×							
	Create Swc Implementatio	on					JODOGENENCE							
	ARPackage:	🖆 /Demo @ S	WC_CyclicCo	unter.arxml			~							
	Short Name:	Impl_SWC_Cyd	licCounter											
	Long Name:													
	Internal Behavior:	SWC_Cyclic	:Counter_IB_0	(/Demo/SWC_C	Cyclic(Counter)	*							
	Sw Version:	1.0.0				+ AUTOSAR B	Ruilder			×]			
	Programming Language:	C												
	Code Generator:				_ '	Create Code [Descriptor	+ AUTOSAR	Builder		I			×
	Vendor ID:	0						Create Autos	ar Engineeri	ing Object			Ŧ	
	Required RTE Vendor ID:					Short Name:	SRC		ar Engineen	ing object			» (D	9
-	Compilers:	Short Name	Name	Version	Ver	Long Name:							SDEXPERI	ENCE
xml						File:	X SWC_CyclicCounter.ar		EngObj					
						Category:		File:						6
	Resource Consumption:	ResourceCons	umption [/Dei	mo/Impl_SWC_0	Cycl	Descriptors:	EngObj	File Type:	SWSRC - fo	r source code				\sim
	Per Instance Memory size:	Alignment	Size	Per Instance M	1em			Runnables:	Element	File				÷
							e							+
	Code Descriptors:	SRC [/Demo/I	mpl_SWC_Cyd	licCounter/]	_									
						?		?				ОК	Cancel	
							×	L			1			
	~													
	?					ОК	Cancel							





EB tresos & AUTOSAR Builder (AB) workflow

Create SWCD Importer

1. Create new System Description Importer in EB tresos

- Right-click on project \rightarrow "Im- and Exporters"
- Name: SWCD_Importer
- Importer/Exporter: System Description Importer
- Mode: Import Only
- Click Next

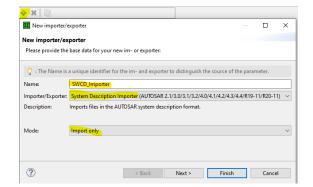
2. Add Input Files

- Click **New...** and add the following paths
 - output\generated\swcd*
 - ..\..\ab_workspace\MyDemo_AB_Tresos*.arxml
- Click Finish
- Run the importer
 - Click Run Importer
 - There should be no error when importing.
 - − This importer will be used for AB \rightarrow EB tresos Workflow.

Create, manage and run im- and exporters

Create, manage and run im- and exporters

Create an importer to import data into your project, or create an exporter to export data to external files.



Create, manage and run im- and exporters				×
Create, manage and run im- and exporters Create an importer to import data into your project, or crea	te an exporter to export data to external files.			-
+ X 🗎	Name: SWCD_Importer			
⊄ SWCD_Importer (System Description Importer (AUTOSAR	2.1/3: General /All Model: System Model Import] Variants ^O The priority of the input files is related to the order they are added. The first file has the highest priority, thus it is the la Input files: Output/generated/swcd/*	st file impor	ted. Add New Edit Remove	
	Meta model version: Ø Auto detect for each file		Down	~
	Validation Validate against X0AL schema AUTOSAR schema varianti standard Import customization:			~
< Run Importer Run Exporter	None	<u>a</u> pply	Canc	
num exporter	OK E	2PPP7	Calic	

BSW ports workflow (DET example)

EB tresos Studio

1. Create new BSW Port

We want to create a new RTE port in a BSW module that could be used by the SWC.

Project Explorer 🔀	
MyDemo_AB_Tresos	Det
 Ø ECU (WINDOWS, WIN32X86) 	
> 🐏 Atomics (V1.0.13, AS4.0.3)	
> 🐏 Base (V5.0.30, AS4.0.3)	Name 🗁 Det
> Parameter States (V1.15.5, AS4.0.3)	
> 👺 Compiler (V1.0.6, AS4.0.3)	General Det Accessing Software Components Det Notifications EB Published Information Published Information
> 👺 Dem (V6.4.0, AS4.0.3)	
Bet (V6.5.5, AS4.0.3)	
 W Det (V6.5.5, AS4.0.3) Det 	Englishe Bita Harana' Day and a
	Enable Rte Usage* 👔 🔽 🎸 🗸
Oet	
Det > Pet EcuM (V5.15.6, AS4.0.3)	Enable Rte Usage* 👔 🛃 🎸 🗸
Det EcuM (V5.15.6, AS4.0.3) Wake (V4.0.26, AS4.0.3) Make (V4.0.2, AS4.0.3) Mcu (V4.0.3, AS4.0.3) MemMap (V1.3.8, AS4.0.3)	Det Accessing Software Components*
 Det EcuM (V5.15.6, AS4.0.3) Make (V4.0.26, AS4.0.3) Make (V4.0.3, AS4.0.3) 	Det Accessing Software Components*
Det EcuM (V5.15.6, AS4.0.3) @: Make (V4.0.26, AS4.0.3) @: Make (V4.0.3, AS4.0.3) @: Mcu (V4.0.3, AS4.0.3) @: MemMap (V1.3.8, AS4.0.3)	Det Accessing Software Components*
 Det Set EcuM (V5.15.6, AS4.0.3) Moke (V4.026, AS4.0.3) Mcu (V4.0.3, AS4.0.3) Mcu (V4.0.3, AS4.0.3) So (v6.0.302, AS4.0.3) 	Det Accessing Software Components*

2. Generate SWCD data

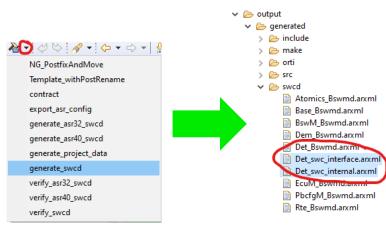
- Generate the SWCD data based on the new configuration.
- Click "generate_swcd"
- 3. There will be two additional .arxml files afterwards
 - Det_swc_interface: Definition of the port interface
 - Det_swc_internal: Implementation, internal behavior, etc.

AUTOSAR Builder

1. AB project changes on the fly

 New DET Port Interfaces, Atomic Component and Implementation will occur **automatically**.





Connect ports in AUTOSAR Builder

1. Create AUTOSAR file

- − File → New → AUTOSAR File
- File name: ToplevelComposition.arxml
- ARPackage Name: Demo
- 2. Create new Composition Type
 - Composite Components \rightarrow new \rightarrow Composition Type
 - ARPackage: ToplevelComposition.arxml
 - Short Name: Toplevel
 - Click OK

Hint: Double-click on Toplevel \rightarrow Opens Diagram view

- 3. Drag and drop SWC_CylicCounter and Development Error Tracer from Atomic Components to Diagram.
- 4. Connect both components by dragging the connector from **Det** to **SWC_CylicCounter**.

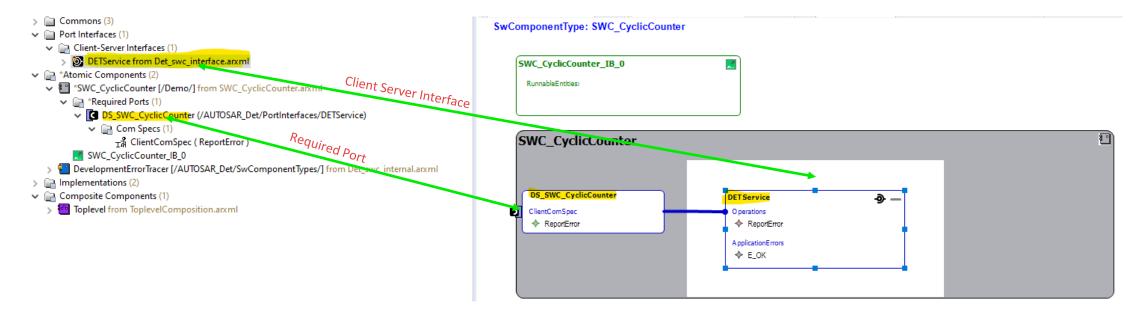


	osition Component Type		
ARPackage:	🖄 /Demo @ ToplevelComposition.arxr		
Short Name:	Toplevel		
Long Name:			
		OK Ca	ncel

Connect ports in AUTOSAR Builder

5. Visualize SWC Content

- Double-click SWC_CyclicCounter in Diagram View.
- The required ports are automatically generated for the SWC.



AUTOSAR Mode Management

In mode management there are two parties involved: **Mode managers** and **mode users**.

Mode managers:

Responsible for switching modes and are the only instances able to change the value of the global variable. A mode manager is either an SWC, which provides a **ModeRequestPort** or a Basic Software Module (e.g. BswM), which either provides also a **ModeRequestPort** in its Software Component Description or a **ModeDeclarationGroup** in its Basic Software Module Description.

application mode manager mode request mode switch port port mode request mode switch port port mode machine Instance System Services System Services

Mode users:

Will be informed of **Mode switches** via well-defined mechanisms and have the possibility to read the currently active mode at any time. If a Mode user wants to change into a different mode, they can request a Mode switch from the corresponding Mode manager.

1. Create BswM RTE port

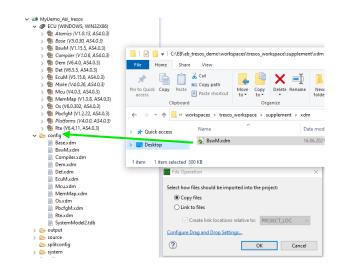
Because we want call our runnables based on modes defined by the BswM, we need to create a port to inform the SWC about mode changes.

- a. Copy (drag & drop) the **BswM.xdm** from **supplement/xdm** folder to your EB tresos project config folder and overwrite the current one.
 - The .xdm file contains the necessary configuration for creating BswM ports.

2. Compare BswM config

If you are interested in the config changes within the copied BswM, you can use the EB tresos Studio built-in compare tool

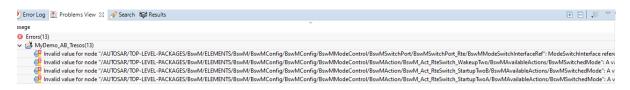
- a. Rename the BswM.xdm.bak \rightarrow Bswm_old.xdm.
- b. Mark both BswM files (Press Ctrl button).
- c. Press button "Only show unprocessed changes".
- d. The changes tree will show all the additional configurations.



> PbcfgM (V1.2.22, AS4.0.3)	MyDemo_AB_Ti	esos/config/BswM_old.xdm <	- MyDemo_AB_Tresos/config/BswM.xdm	•		
 Platforms (V4.0.0, AS4.0.3) Rte (V6.4.11, AS4.0.3) 	Merged model			MyDemo_AB_Tresos/config		MyDemo_AB_Tresos/config
W Re (V0.4.11, A34.0.5) Config	Mergeumouer					
		V BswMConfig				
Base.xdm		IMPORTER_INFO			* *	
Base.xdm.bak		✓ Can BswMConfig			•	
BswM_old.xdm		IMPORTER_IN			4 ×	Imp_AS2
E DSWIMACUTT	· /	🗸 🙀 BswMArbitra			*	
Compiler.xdm Open		IMPORTE	R_INFO		*	Imp_AS2
Compiler.xdm.		V 🔚 BswMLog	gicalExpression		4 X	
📄 Dem.xdm 🛛 📔 Copy	Ctrl+C	BswN	LogEx_StartupTwoB		4 X	
Dem.xdm.bak n Paste	Ctrl+V		1_LogEx_StartupTwoB +		**	
Det.xdm	Delete	@, IN	IPORTER_INFO		4 X	Imp_AS2
Det.xdm.bak		🔜 B:	swMLogicalOperator		4 X	BSWM_AND
EcuM.xdm Move	[🗸 📊 Bs	swMArgumentRef		4 X	
EcuM.xdm.bak Rename		~ @	h	ASPath:/BswM/BswM/BswM	*	ASPath:/BswM/BswM/BswM
📄 Mcu.xdm 🔊 Import			B IMPORTER_INFO		4 X	Imp_AS2
Mcu.xdm.bak		🕞 BswN	1_LogEx_Shutdown		*	
📄 MemMap.xdm 🏊 Export		V 🖪 BswMMc	deCondition		**	
MemMap.xdm Refresh	F5	🗸 🕞 Bswlv	1_Cond_WakeupTwo		•	
S.xdm			swMConditionValue	BswMBswMode	•	BswMBswMode
📄 Os.xdm.bak 🛛 Team	>		BswMBswMode		•	
PbcfgM.xdm Compare With	>	Each Other	BswMBswRequestedMode		•	99
PbcfgM.xdm.b Replace With	>	Local History	1_Cond_SwcCyclicRun		•	
Rte.xdm	· · · ·	Locar matory	Cond SwcCyclicShutdown		4 X	

3. Create Mode Declarations Groups

After we copy the BswM.xdm, we will see some EB tresos errors in the "Problems View".



This is because the BswM has no idea of the **Modes and Port Interfaces.** We need to give that information to the BswM.

• Copy (Drag & Drop) the content of the supplement/arxml folder to your AUTOSAR Builder project (prepared ModeDeclarationGroups)

Note: We copy that directly there because we import everything from the AUTOSAR Builder project in our EB tresos project and we need these groups for further processing in AUTOSAR Builder.



- BswMMode.arxml: Contains the modes defined by BswM
- BswMAppMode.arxml: Contains the modes of the application SWC_CyclicCounter.

- 4. Inspect the Mode Declaration in AB
 - The Mode Declarations can be viewed in AB.
 - The modes correspond to the ones defined in EB tresos BswM configuration.

5. Import ModeDeclarationGroups & Port Interfaces in EB tresos

– Run "SWCD_Importer" (Right-click on project → Im- and Exporters..)

Note: If you see import warnings, you can ignore them as they are due to a missing mapping.

- The BswM errors in the "Problems View" should disappear.
- Now the BswM has all the information he needs to generate its SWCD.

6. Generate SWCD in EB tresos

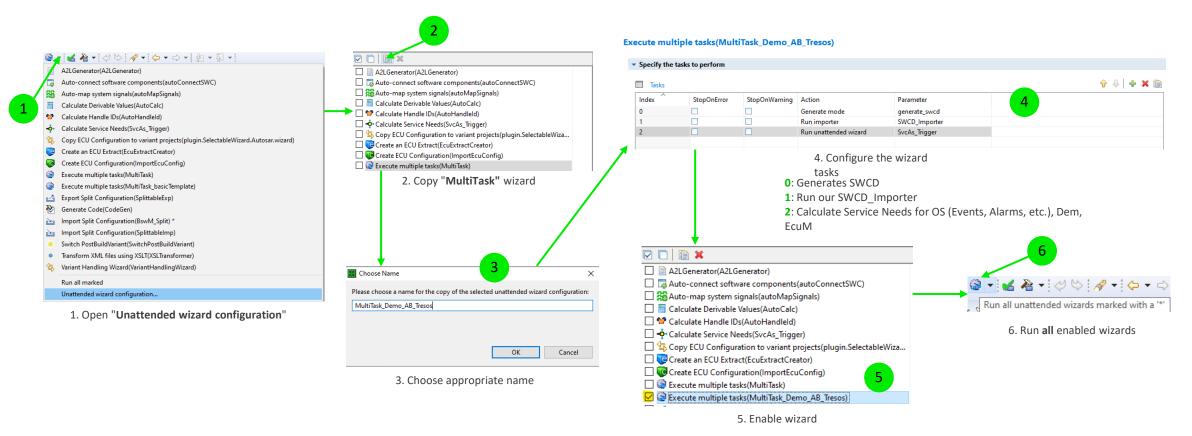
- Run "generate_swcd".
- In your AB project a new "Atomic Component" (BswM) will appear.
- Now we can use the BswM, for example, to start our runnables or to inform the BswM about mode changes (e.g. shutdown).



- ✓ Atomic Components (3)
 - SWC_CyclicCounter [/Demo/] from SWC_CyclicCounter.arxml
 - > 📄 Required Ports (1)
 - SWC_CyclicCounter_IB_0
 - BswM [/AUTOSAR_BswM/SwComponentTypes/] from BswM.arxml
 - TevelopmentErrorTracer [/AUTOSAR_Det/SwComponentTypes/] from Det_swc_internal.arxml

Create EB tresos MultiTask wizard

As we have to do several steps such as **importing**, **generate_swcd**, etc., we create an MultiTask wizard in EB tresos that can automatically execute all manually steps sequentially. We will update this wizard with an additional task.

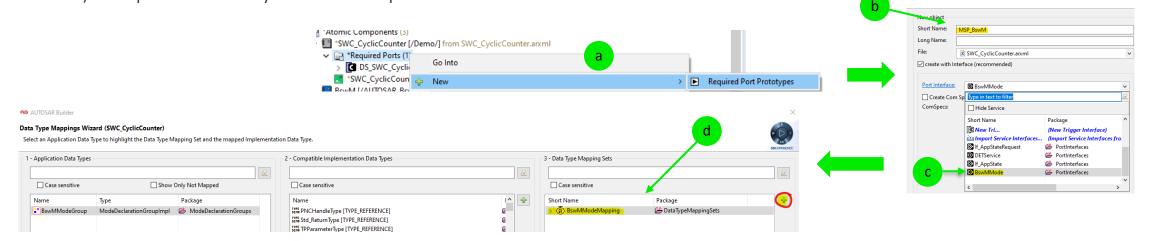


7. Create SWC ModeSwitchPort (MSP)

First, we need to create a required port (R-Port) that BswM can inform our SWC about mode changes. We can either do that manually or by connecting the BswM MSP to our SWC (see "BSW Port Workflow Example"). We want to do that manually to see what we can do if we want to create new R- or P-Ports.

- a. Right-click on "Required Ports" and add a new Port Prototype.
- b. Short Name: MSP_BswM
- c. Port Interface: Choose BswMMode Port Interface
- d. Add the Data Type Mapping

The **Data Type Mapping** is necessary, because we need to know which Data Type (e.g. uint8, uint16, structs, etc.) we expect to receive by the connected port.

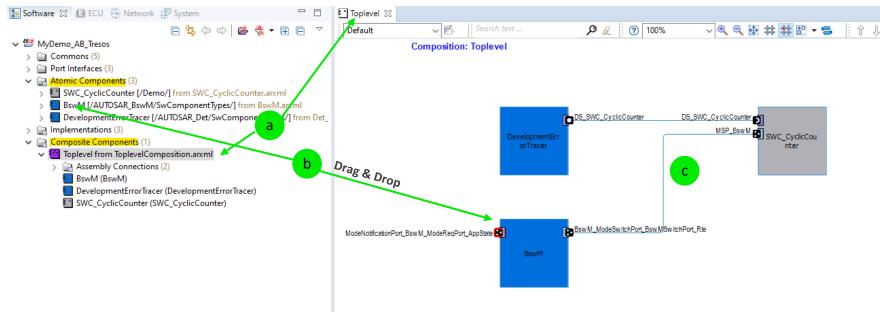


AUTOSAR Builder

8. Connect SWC ModeSwitchPort (MSP)

Now, we want to connect the MSP of the BswM with the already created MSP_BswM of our SWC.

- a. Open **Top-level Diagram View** from Composite Components (double click).
- b. Drag & Drop BswM from Atomic Components to top-level view.
- c. Connect BswM MSP with MSP_BswM of SWC.



Create runnables

Now we want to add some **runnables** to our SWC. As you know, runnables describe functions that can be activated by an RTE event.

In the following we will deal with the following topics:

- We will add an (explicit) interrunnable variable for "CurrentCounterValue":
 - **Explicit** means the runnable entity can directly access an interrunnable variable. Changes are immediately visible to other runnable entities with explicit access to the interrunnable variable.
 - Communication across all runnables within an SWC is possible.
- We will add the following runnables:
 - Event-triggered ("SWC_CyclicCounter_Init"),
 - Time-triggered ("SWC_CyclicCounter_Cyclic")
- We will add access points:
 - Server-Call Points Access: Report Error via Det Port
 - InterRunnable Variable Access: Read and Write access on variable

Create runnables

- 1. Create Inter Runnable Variable (IRV)
 - I. Open the context menu of Internal Behavior SWC_CyclicCounter_IB_0
 - II. Add New \rightarrow Inter Runnable Variable

2. Create Runnable SWC_CyclicCounter_Init

I. Open the context menu of Internal Behavior SWC_CyclicCounter_IB_0

Create Runnable Entities

2

- II. Add New \rightarrow Runnable Entities
- III. Tab IRV Accesses:
 - a. Parameters:
 - Write Access on IRV
 - b. Events:
 - Event Name: MSE_CyclicCounter_Init
 - Event Type: Mode Switch Event
 - Activation: ON-ENTRY
 - Required port: MSP_BswM
 - Mode Decl.: RUN_TWO
 - See EcuM & BswM background slide for information

Create Inter F	(unnable Var	lable		3000
Short Name:	CurrentCou	nterValue		
Long Name:				
File:	SWC_Cy	clicCounter.arxml		
Type:	R uint8 [VA	LUE]		
Impl Policy:	(None)			
Communicatio	on Approach:	O Implicit Explicit	iit	
Init Value:		[Num] : 0		
Addressing Me	thod:	📧 (None)		
Sw Calibration	Access:	(None)		

1

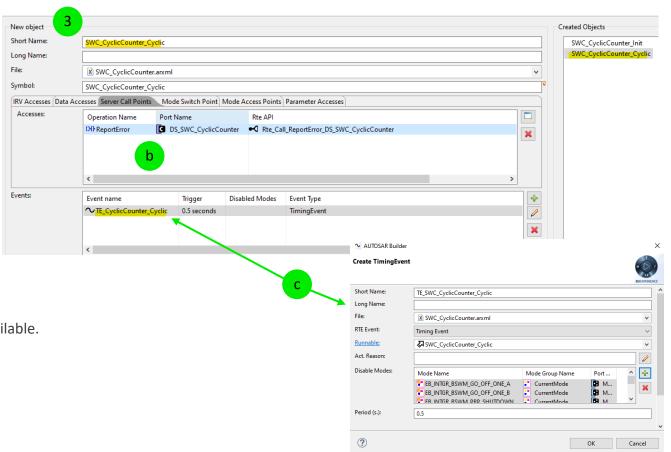
	Z										
New object	_						Created Objects				
Short Name:	SWC_CyclicCounter_Init						SWC_CyclicCounter	Init			
Long Name:							SWC_CyclicCounter	Cyclic			
File:	SWC_CyclicCounter.arx	ml				*					
Symbol:	SWC_CyclicCounter_Init					0					
IRV Accesses Data	Accesses Server Call Points N	Aode Switch F	Point Mode Access Points Parameter Accesses				AUTOSAR Builder				×
Parameters:	Data Name	R/W	Rte API				Create SwcModeSv	vitchEvent			
	W CurrentCounterValue	w	Rte_InvWrite_SWC_CyclicCounter_Init_CurrentCo	unterValue	а	22					IDEPERENCE
							Short Name:	MSE_CyclicCo	unter_Init		
							Long Name:				
	<					>	File:	SWC_Cyclic	:Counter.arxml		~
	×						RTE Event:	Mode Switch E	vent		~
Events:	Event name	Trigger		Disabled Modes	Event Type	+	Runnable:	SWC_Cyclic	:Counter_Init		*
	MSE_CyclicCounter_Init	ON-ENT	RY-EB_INTGR_BSWM_RUN_TWO from MSP_BswM		Sw SwitchEvent	0	Act. Reason:				0
					- b	36	Disable Modes:	Mode Name	Mode Group Name	Port Name	+
	<			_	-						36
						,					
							Activation:	ON-ENTRY			~
							Required port:	MSP_BswM	(BswMMode)		~
							Mode Decl. 1:	EB_INTGR_B	SWM_RUN_TWO (BswN	ModeGroup)	*
ation											
-											
							(?)			OK	Cancel

Create runnables

3. Create runnable SWC_CyclicCounter_Cyclic

- I. Open the context menu of Internal Behavior SWC_CyclicCounter_IB_0
- II. Add New \rightarrow Runnable Entities
- III. Tab IRV Accesses:
 - a. Parameters:
 - Read/write access on IRV
- IV. Tab Server Call Points (SCP):
 - b. Accesses:
 - Synchronous Server Call Point for operation Det ReportError
 - c. Events:
 - Event Name: TE_CyclicCounter_Cylic
 - Event Type: Timing Event
 - Disable Modes: Add all modes except RUN_TWO
 - The runnable will not triggered if one of the other modes will be available.
 - Period: 0.5 (seconds)

Create Runnable Entities



AUTOSAR Builder

Validate SWC

AUTOSAR Builder validation rule for EB tresos Studio 1.

- Open the validator 🧌 🖬 and choose "Validate for EB tresos Studio". a.
- Right-click on your project \rightarrow AB Validation \rightarrow Project (or click the validator button after you marked your project). b.

Some validation errors will be shown in the "Validation" view. С.

🚺 Valid	dation 🔀 🇯 Plug-lets 📮 Console
0 errors	s, 0 warnings, 1 info
Descrip	tion
✓ 181	EBT_40027 - EB Tresos Studio reject attributes present in arxml with no value (10 of 10 items)
1	鴙 Attribute "gShortName" of ExecutableEntityActivationReason "SWC_CyclicCounter_Cyclic.activationReasons[0]" is set while no v
1	😘 Attribute "gShortName" of ExecutableEntityActivationReason "SWC_CyclicCounter_Init.activationReasons[0]" is set while no valu
	😘 Attribute "gShortName" of ExecutableEntityActivationReason "SWC_CyclicCounter_Init.activationReasons[1]" is set while no valu

- Double-click on the error will guide you to the error. d.
- Right-click \rightarrow Quick Fix (Crtl +1) will give you some hints how to fix e. the issue.

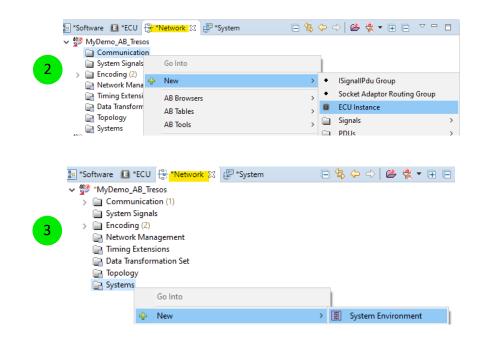
You can use "Find Similar Problems" to fix all problems at once.

Select a fix	0
Select the fix for "Attribute "location" of SElement "SWC_CyclicCounter.containedExtensions[0].elements[13 is set while no value exists. This will lead to an error while importing the model into EB Tresos Studio".	"
Select a fix:	
Unset attribute	
AB Quick Fix	
Select a fix	
Select the fix for "Attribute "location" of SElement "SWC_CyclicCounter.containe Unset attribute	
is set while no value exists. This will lead to an error while importing the model i Problems:	
Select a fix: Resource 22 Location	Select All
🛛 😼 SWC_CyclicCounter.axml /Demo/SWC_CyclicCou	Deselect All
Unset attribute 🛛 🗹 🔓 SWC_CyclicCounter.arxml //Demo/SWC_CyclicCou	Deselect All
	d Similar Problems
☑ 🔓 SWC_CyclicCounter.axml /Demo/SWC_CyclicCou	
□ 🔓 Atomics_Bswmd.arxml /AUTOSAR_Atomics/Bs	
☑ 🔓 SWC_CyclicCounter.axml /Demo/SWC_CyclicCou	
☑ 🔓 SWC_CyclicCounter.axml /Demo/SWC_CyclicCou	
SWC_CyclicCounter.arxml //Demo/SWC_CyclicCou	
SWC_CyclicCounter.anzml /Demo/SWC_CyclicCou	
☑ 🔓 SWC_CyclicCounter.arxml /Demo/SWC_CyclicCou	
Problems: ⑦	Cancel
Resource Select All	Curicer
🗹 😰 SWC_CyclicCounter.arxml /Demo/SWC_CyclicCou Deselect All	
Destect All	
Find Similar Problems	
Find similar Proteins	

SWC to ECU mapping

After we created our SWC and connected it to the BSW prototypes, we need to map the SWC prototypes to a specific ECU.

- 1. Create new AUTOSAR file
 - a. Filename: System.arxml
 - b. ARPackage: Demo
- 2. Create new ECU instance
 - a. Open "Network" view
 - b. ARPackage: **Demo@System.arxml**
 - c. Name: DemoEcu
- 3. Create new System Environment
 - a. ARPackage: Demo@System.arxml
 - b. Name: DemoSystem
 - c. Category: SYSTEM_DESCRIPTION
 - d. RootSwComposition Prototype: Toplevel
 - e. System Version: 1.0.0
 - f. Fibex Element: Choose DemoEcu Instance



SWC to ECU mapping

Now all SWC can be mapped to the ECU instance. Therefore, we can use the integrated AUTOSAR Builder wizard.

Oţ	oen S\	W "Component	to ECU Mapping Wiza	rd"		iagram ECU Mapping Wi					
a.	Double	e-click on DemoSy	stem.								
b.	Mark <u>a</u>	all SWC Componer	nt Prototypes and \pm cor	nnect	t them to the EC	U instan	ce.				
Compone	ent Mappings	5									
Compone	ent Mappings Status		Component Context		Component Type		ECU	Co	mponent to E	Lu Mapp	-+-
Compone		Component Prototype	Component Context	Ø	Component Type *	Ø	ECU *	Co	mponent to e *	ecd Happ Ø	-0-
Compone	Status <filter></filter>	Component Prototype	Component Context	Ø 8	Component Type * BswM [/AUTOSAR_BswM/SwCc	-	ECU *	Ø	mponent to E * DemoEcuMapping	Ø	+ <>
Compone 1 2	Status <filter></filter>	Component Prototype *	ð *		*	🗱 DemoEcu	ECU *	Ø 97	*	Ø 1	◆
1	Status <filter></filter>	Component Prototype * @ BswM [/Demo/Toplevel/]	¢ *	1	* BswM [/AUTOSAR_BswM/SwCo	🗱 DemoEcu	ECU *	⊘ 卵 卵	* DemoEcuMapping	0 7 7	+ + -

- 4. The **DemoEcu** now contains all SWC prototypes we need to configure the RTE.
- 5. Next, we are going to import the entire system into our EB tresos Studio project.



EB tresos AutoCore OS & RTE

© Elektrobit 2021 | Confidential information

Create OS tasks

As we know that every runnable needs to be mapped to a task, we have to create <u>two</u> additional OS tasks in our EB tresos AutoCore OS configuration. The two tasks are needed to map the cyclic and the event-driven runnable that we have created.

Create two additional OS tasks

- Open the OS config in EB tresos Studio "Project Explorer" and go to tab "OsTask".
- Add <u>two</u> additional tasks.

General		shed Information	OcAlarm	OrAnnMode	OcAppli	cation (DeCounter	OrEve	ant (OcSpinlock	Ocle	c (0r	er OcRe	cource	Oct	orscher	duleTable
General	LD FUDI	sneu miormation	OSAIdIIII	OSAppivioue	OSAppin	cation	oscounter	USEVE		osspiniock	USIC			source	USI	ask Osserier	ullelable
	OsTask*																
Ind	êx 🏳	Name				OcTack	Activation			OsTaskPriori	.	ī	0-5+	acksize		OsTaskSched	ula
inu		indiffe			123	USIdSK	Activation	[I	123	USIASKPITUT	LY	1	g 0550	acksize		Usiaskacheu	uie
	0 🗁	Init_Task			121		1	12		12	27 [1024		NON	
	1 🗁	SchMDiagStateTa	sk_20ms		12		1	122			50			512	D	FULL	
	2 🗁	Rte_Event_Task			B		1	12			1	3		1		FULL	
	3 🗁	Rte_Time_Task			B		1	12			2	6		1		FULL	

- 1. Rte_Event_Task:
 - Os TaskPriority: 51
- 2. Rte_Time_Task:
 - Os TaskPriority: 52

In this example all other params could keep their default values. Any needed params such as **OsStackzize** will be calculated automatically.

Update MultiTask wizard

We need to update/extend our previously created MultiTask wizard

Specify the t	asks to perform			
Tasks				
Index ^	StopOnError	StopOnWarning	Action	Parameter
0	\checkmark		Generate mode	generate_swcd
1	\checkmark		Run importer	SWCD_Importer
2	\checkmark		Run unattended wizard	EcuExtractCreator
3	\checkmark		Run unattended wizard	AutoHandleld
4	\checkmark		Run unattended wizard	SvcAs_Trigger

Add EcuExtractCreator and AutoHandleID

- EcuExtractCreator: Creates an ECU extract that is needed to select a system and ECU for RTE configuration.
- AutoHandleID: Calculates the IDs for the different ACG plugins used such as CanIf, Dem, etc. (not necessary here, only for completeness).

Important: Please keep the task sequence as shown above, otherwise it is not guaranteed to have a correct workflow.

Import System into RTE

1. Run the MultiTask wizard

- All necessary files will be generated and imported.

2. Select System and EcuInstance

– You need to select a System and Eculnstance for configuring the RTE.

l can be act creation,
~
~

3. Open RTE Editor

- Change "Rte Generation Output" to Full

🖕 *Project Explorer 🕴	⊟ 🕏	~	ង្ហីថំ "Rte Editor 😢	
 MyDemo_AB_Tresos 		^	Rte Editor <u>Swaming(s) detected</u>	
> 👺 Base (V5.0.30, AS4.0.3)			General Implementation Selec Partitioning Optimizations Event Mapping Data Mapping Exclusive Areas	/leasi
> 🚇 BswM (V1.15.5, AS4.0.3)				
> 👺 Compiler (V1.0.6, AS4.0.3))		Rte Generator Output 👔 Full	
> 💁 Dem (V6.4.0, AS4.0.3)				
>			OSEK OS compatibility mode 🔣 🗌	
> 9 Make (V4.0.26, AS4.0.3)			Os counter /Os/Os/Rte_Counter ~	
> 💁 Mcu (V4.0.3, AS4.0.3)			Os counter /Os/Os/Rte_Counter V	
> 💁 MemMap (V1.3.8, AS4.0.3	5)		Enable DET Error Reporting 🕱 🔲 Report Uninitialized Errors 🕱	
> 🐏 Os (V6.0.302, AS4.0.3)				
> 🚇 PbcfgM (V1.2.22, AS4.0.3)			▼ Rte Generation	
> Platforms (V4.0.0, AS4.0.3)	0			
Rte (V6.4.11, AS4.0.3)			Handling of unconnected require ports	1
Rte (Generic Editor)				1
Rte Editor			Contraction of the second se	1

4. Runnable to task mapping

- Open tab "Event Mapping"

Rte Editor <u>5 warning(s) detected</u>

General	Implementation Selec	Partitioning	Optimizations	Event Mapping	Data Mapping	Exclusive Areas	Measurement &	Calib	NVRAM Allocation	VFB Tracing	Bsw Trigger Conn	ecti Bsw	Mode Mapp	ing [»] 2
🔻 Unm	napped RTE and BSW ev	vents												
	lay RTE and Bsw events w Unmapped RTE and BSV		d mapping only	X										, the second sec
	Event		Event type		Executa	ble entity			nstance				Categ	🖹 Re
	MSE_BswM_ModeReqPC MSE_BswM_ModeReqPC MSE_CyclicCounter_Init	ort_Ap 📄 ort_Ap	ModeSwitchEv ModeSwitchEv ModeSwitchEv ModeSwitchEv	ent Entry ent Entry	RE_BswM	ModeKeqPort 4 ModeReqPort 4 ModeReqPort 4 ModeReqPort 6 ModeReqPort 7 ModeReqPort 7 Mo	AppState) /E	cubrtract/lopLevelC cubrtract/TopLevelC cubrtract/TopLevelC cubrtract/TopLevelC	omposition/ omposition/	BswM BswM		CAT1B CAT1B	
🔻 Мар	ped RTE and BSW even	nts												
Task	/Os/Os/Rte_Eve	ent_Task		~										
	Mapped RTE And BSW e	events												600
	P. 📄 Event					Ev	ent type		Executa	ble entity			📄 Ins	tance

- Map all ModeSwitchEvents (MSE) to Rte_Event_Task (Note: AppState events are not necessary here, but they are part of the entire demo)
- Map RteTimingEvent (TE) to Rte_Time_Task
- Close RTE, save your project, and run the "Calculate Service Needs" wizard.

EB tresos & AUTOSAR Builder workflow

Until now you have accomplished the following:

- 1. You created a new project in AB and know how to include BSW SWCD.
- 2. You created you own SWC in AB and learned how to visualize them.
- 3. You learned how to validate your SWC against EB tresos and other validation rules.
- 4. You created your first connection between your SWC and the ports provided by the DET BSWMD.
- 5. You got some insides about Mode managers and Mode users that are necessary for switching modes in AUTOSAR.
- 6. You got familiar how to add and access necessary SWC subelements, such as:
 - Creating Runnables
 - Creating ModeSwitchPorts (MSP)
 - Creating Inter Runnable Variables (IRV)
 - Creating Server Call Points (SCP)

- You learned how to connect the Basis Software Manager (BswM) to your SWC and how to use ModeDeclarations.
- 12. You learned how to map your SWC to an ECU creating a system environment.
- 13. You got some insights about "How to create EB tresos unattended wizards" and how to use them to import your system into the RTE.
- 14. You got familiar with the process of:
 - Creating OS tasks for "runnable to task mapping"
 - Configuring the RTE and map the runnable events to the pre-configured tasks

Summary: You learned the basic workflow concept between EB tresos Studio and AUTOSAR Builder, and how comfortable it is to build up your own system really fast by connecting BSW with application SWC in an easy and visualized manner.

Iterate through the workflows

Working agile or by increments

Normally, an ECU project is not finished after one execution of the workflow. Instead, several iterations are necessary until a software for an ECU is completely finished.

Depending on the changed input, one can jump back to dedicated workflow steps to iterate through the workflow:

- Changed basic software modules
- Changed application
- Revalidation in AUTOSAR Builder necessary
- Additional runnable in SWC
- Changed OS(Task) configuration
- Changed RTE configuration

- \rightarrow jump back to slide 10
- \rightarrow jump back to slide 17
- \rightarrow jump back to slide 19
- \rightarrow jump back to slide 33
- ightarrow jump back to slide 40
- \rightarrow jump back to slide 42



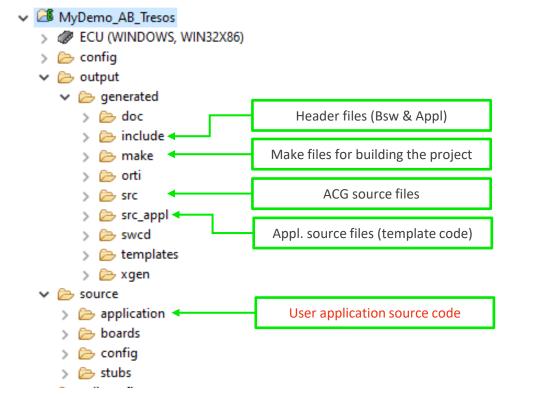
Generate source code and compile

© Elektrobit 2021 | Confidential information

Import System into RTE

1. Generate code

- Run the EB tresos Studio code generation process
 - Note: You can ignore the warnings
- Inspect generated code
 - The generation process will output several files within your project



2. Copy application source templates

Because the template code will be overwritten every time a generation process is started, we need to copy our "SWC_CyclicCounter_Template.c to our "User application source code" folder".

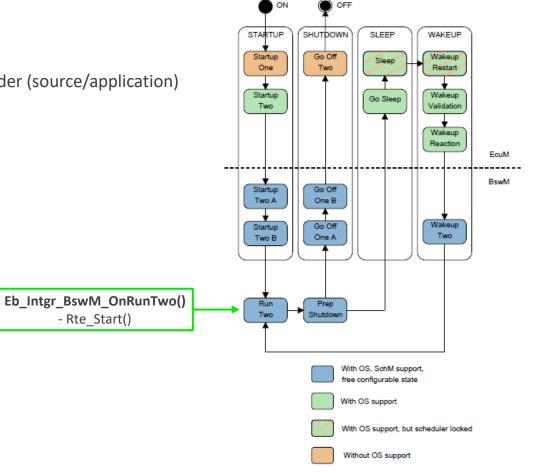
✓ ➢ output	
✓ ≥ generated	
> 🗁 doc	
> 🗁 include	
> 🗁 make	
> 🗁 orti	
> 🗁 src	
🗸 🗁 src_appl	
BswM_Template.c	
DevelopmentErrorTracer_Templ	ate.c
SWC_CyclicCounter_Template.	:
> 🗁 swcd	
> 📂 templates	
> 🗁 xgen	
V 🗁 source	
🗸 🗁 application	Сору
Eb_Intgr_BswM_Modules.h	٧c
Eb_Intgr_BswM_UserCallouts.c	
Eb_Intgr_BswM_UserCallouts.h	
📄 Eb_Intgr_Det.h	
Eb_Intgr_EcuM_Callout_Stubs.c	
Eb_Intgr_InitTask.c	
Eb_Intgr_McuArchClockInit.c	
Eb_Intgr_McuArchClockInit.h	
Eb_Intgr_NvM_Include.h	
Eb_Intgr_NvM_Include.h main.c	
Eb_Intgr_NvM_Include.h	

Update source code

Because the "basicTemplate" only does common **BSW scheduling** (without application) and we are now doing **Full scheduling** (with application), we must start the RTE at the **RUN_TWO** state.

- 3. Enable Rte_Start
 - Edit file "Eb_Intgr_BswM_UserCallouts.c" in "User application source code" folder (source/application)
 - Search for function "Eb_Intgr_BswM_OnRunTwo"
 - Remove following if/endif or change to #if 1 to enable Rte_Start():
 #if 0

```
if ( Rte_Start() != E_OK )
{
    /* Rte start failed */
}
#endif /* 0 */
```



Compile

In this step we just want to compile our project. The "SWC_CyclicCounter_Template.c" does not have any logic – it only contains rudimentary source code for each runnable (C function).

- 1. Update launch_cfg.bat
 - 1. Edit lauch_cfg.bat in <project>/util folder
 - 2. Set the %TRESOS_BASE% variable to your EB tresos installation folder e.g.
 - IF [%TRESOS_BASE%]==[] SET TRESOS_BASE=c:\EB\eb_tresos_demo\tools\tresos\
- 2. Run launch.bat and execute the following make statements
 - 1. make clean
 - 2. make –j

EB make rules	
ED make rules	

Note: On first compilation process, the make environment will automatically extract the GCC compiler.

C:\WINDOWS\system3	cmd.exe				-	- 🗆
Environment vari	oles overview					
TARGET DERIVATE TOOLCHAIN TRESOS_BASE PLUGINS_BASE	= WINDOWS = WIN32X86 = mgcc62 = c:/EB/eb_tresos = c:/EB/eb_tresos	_demo/tools/tresos/ _demo/tools/tresos/plugin	ns			
binaries libraries	C:/EB/eb_tresos_	demo/workspaces/tresos_w demo/workspaces/tresos_w	orkspace/MyDemo_A	B_Tresos/output/b B_Tresos/output/l	lib	
libraries generated files dependency files object files preprocessed fil	C:/EB/eb_tresos_ C:/EB/eb_tresos_ C:/EB/eb_tresos_ C:/EB/eb_tresos_ c:/EB/eb_tresos_ es C:/EB/eb_tresos_		orkspace/MyDemo_A orkspace/MyDemo_A orkspace/MyDemo_A orkspace/MyDemo_A orkspace/MyDemo_A	B_Tresos/output/b B_Tresos/output/1 B_Tresos/output/g B_Tresos/output/d B_Tresos/output/d B_Tresos/output/c	lib generated depend obj	
libraries generated files dependency files object files preprocessed fil EB build rules o Full Scale Targe make clean make clean make gener make _j	C:/EB/eb_tresos_ C:/EB/eb_tresos_ C:/EB/eb_tresos_ C:/EB/eb_tresos_ c:/EB/eb_tresos_ errview ts: (remove o global (remove o ste (run tres (generate cups (check fo src (copy all	demo/workspaces/tresos_w demo/workspaces/tresos_w demo/workspaces/tresos_w demo/workspaces/tresos_w demo/workspaces/tresos_w let except generate folde	orkspace/MyDemo_A orkspace/MyDemo_A orkspace/MyDemo_A orkspace/MyDemo_A orkspace/MyDemo_A orkspace/MyDemo_A ballibs) ballibs) able)	B_Tresos/output/b B_Tresos/output/1 B_Tresos/output/g B_Tresos/output/d B_Tresos/output/d B_Tresos/output/c	lib generated depend obj	



inquiries@elektrobit.com elektrobit.com

© Elektrobit 2021 | Confidential information