

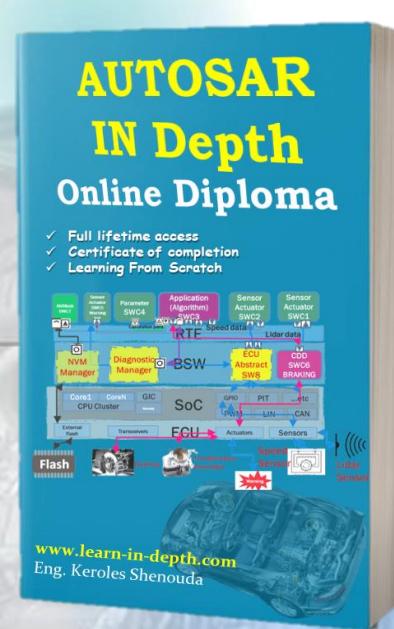
AUTOSAR In Depth

Online Diploma

- ✓ Full lifetime access
- ✓ Access on Android mobile and PC (Windows)
- ✓ Certificate of completion
- ✓ Learning From Scratch

Unit (AUTOSAR SW Layer). Session **AUTOSAR ASW Layer concepts**

- ▶ Use Cases of AUTOSAR Applications
- ▶ AUTOSAR SW "System Design Process"
- ▶ Application Layer
 - ▶ Software components (SW-C)
 - ▶ Virtual Functional Bus (VFB)
 - ▶ Run Time Environment (RTE)
- ▶ Intra- and Inter-ECU Communication
- ▶ VFB generate different RTE based on the INTER/INTRA ECU/Core Communication
 - ▶ IOC (Inter OS Communication)
- ▶ AUTOSAR SWC Template
- ▶ AUTOSAR SWC (Implementation Sequence)



1

#LEARN_IN_DEPTH

#Be_professional_in_embedded_system

eng. Keroles Shenouda

<https://www.facebook.com/groups/embedded.system.KS/>



<https://www.learn-in-depth.com/>
<https://www.facebook.com/groups/embedded.system.KS/>



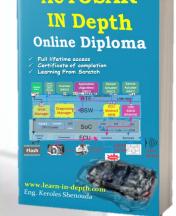
AUTOSAR
IN Depth

Online Diploma

Full lifetime access
Certificate of completion
Learning From Scratch



www.learn-in-depth.com
Eng. Keroles Shenouda



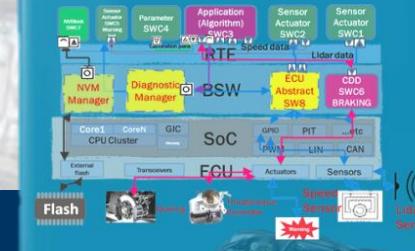
#LEARN_IN_DEPTH
#Be_professional_in_embedded_system

eng. Keroles Shenouda

<https://www.facebook.com/groups/embedded.system.KS/>

AUTOSAR IN Depth Online Diploma

- ✓ Full lifetime access
- ✓ Certificate of completion
- ✓ Learning From Scratch



www.learn-in-depth.com
Eng. Keroles Shenouda

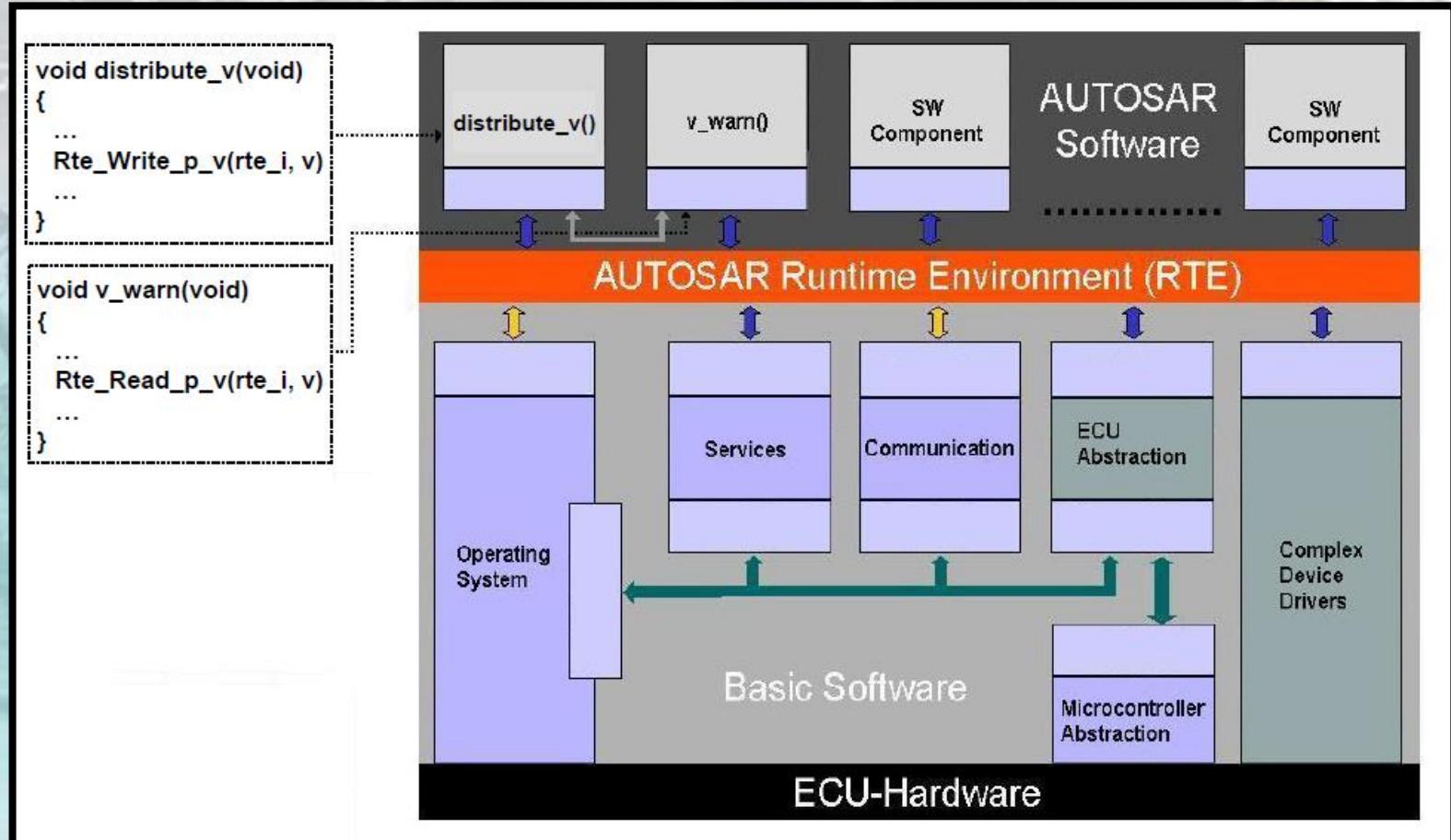
LEARN-IN-DEPTH
Be professional in
embedded system



<https://www.learn-in-depth.com/>
<https://www.facebook.com/groups/embedded.system.KS/>

Use Cases of AUTOSAR Applications

Use Case ‘Pedal Management’ view for one ECU

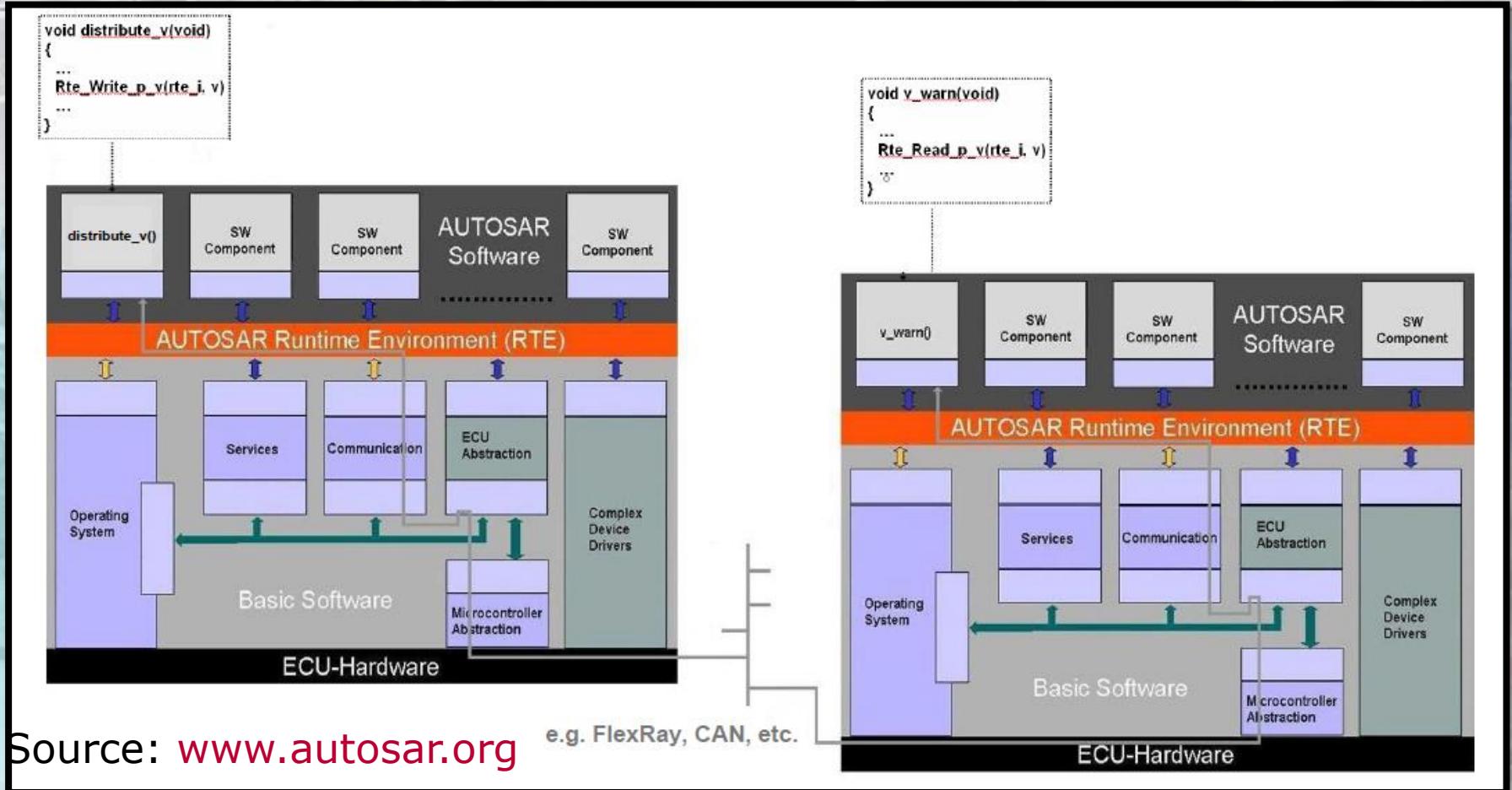


Source: www.autosar.org

www.learn-in-depth.com/
<https://www.facebook.com/groups/embedded.system.KS/>

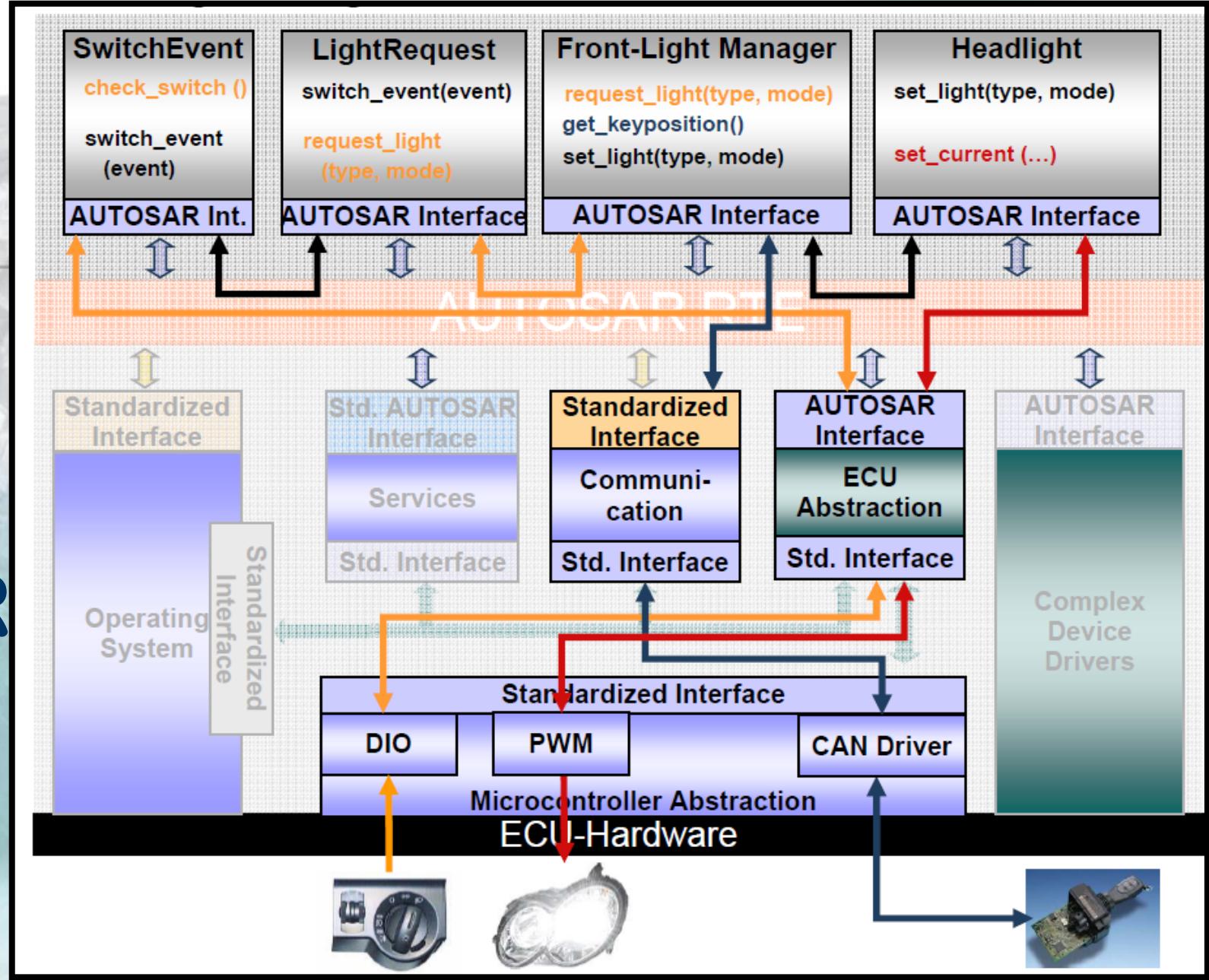


Use Case ‘Pedal Management’ view for two ECUs



<https://www.learn-in-depth.com/>
<https://www.facebook.com/groups/embedded.system.KS/>

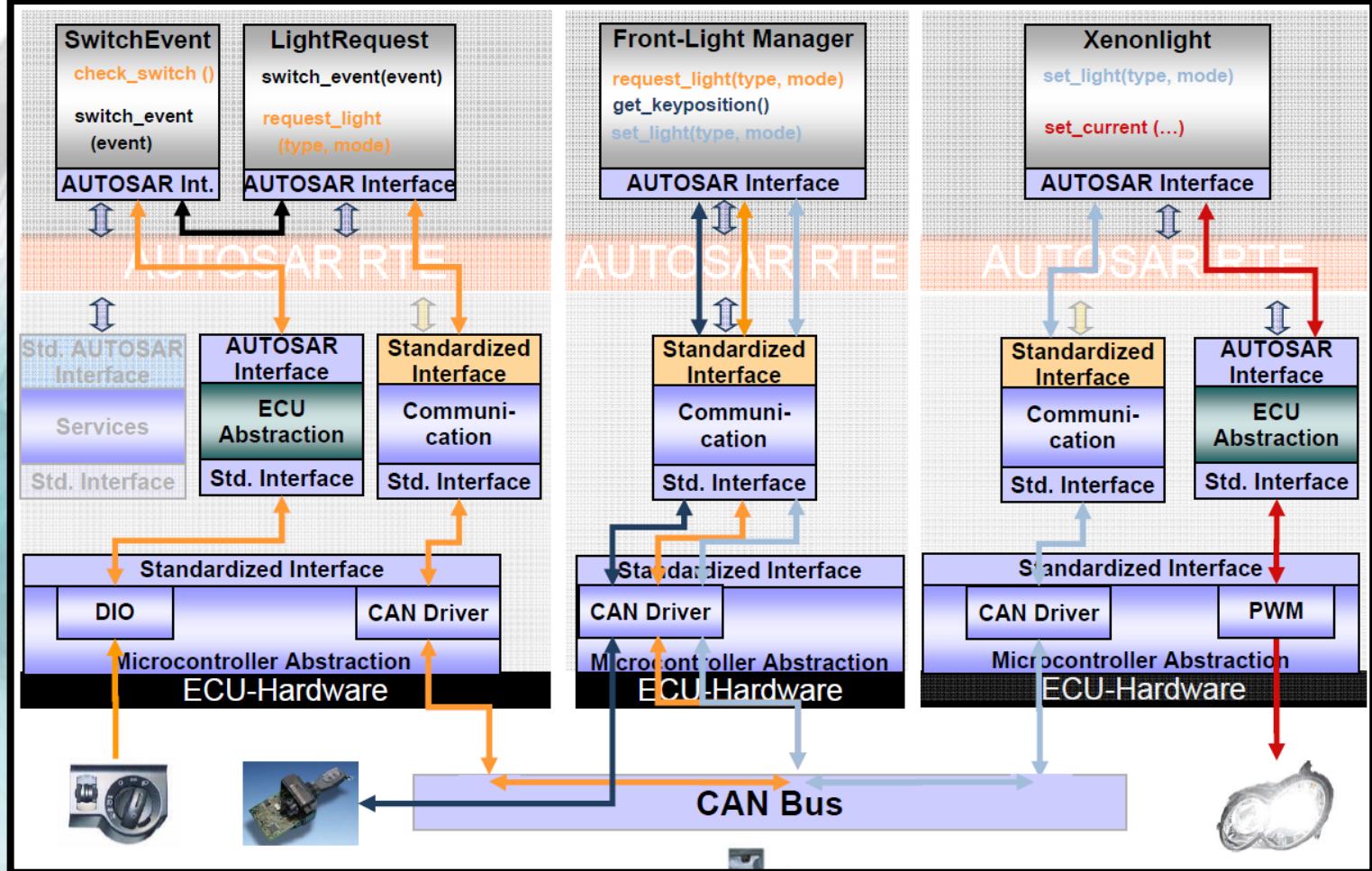
Use case 'Front-Light Management' in AUTOSAR



Source: www.autosar.org

<https://www.facebook.com/groups/embedded.system.KS/>

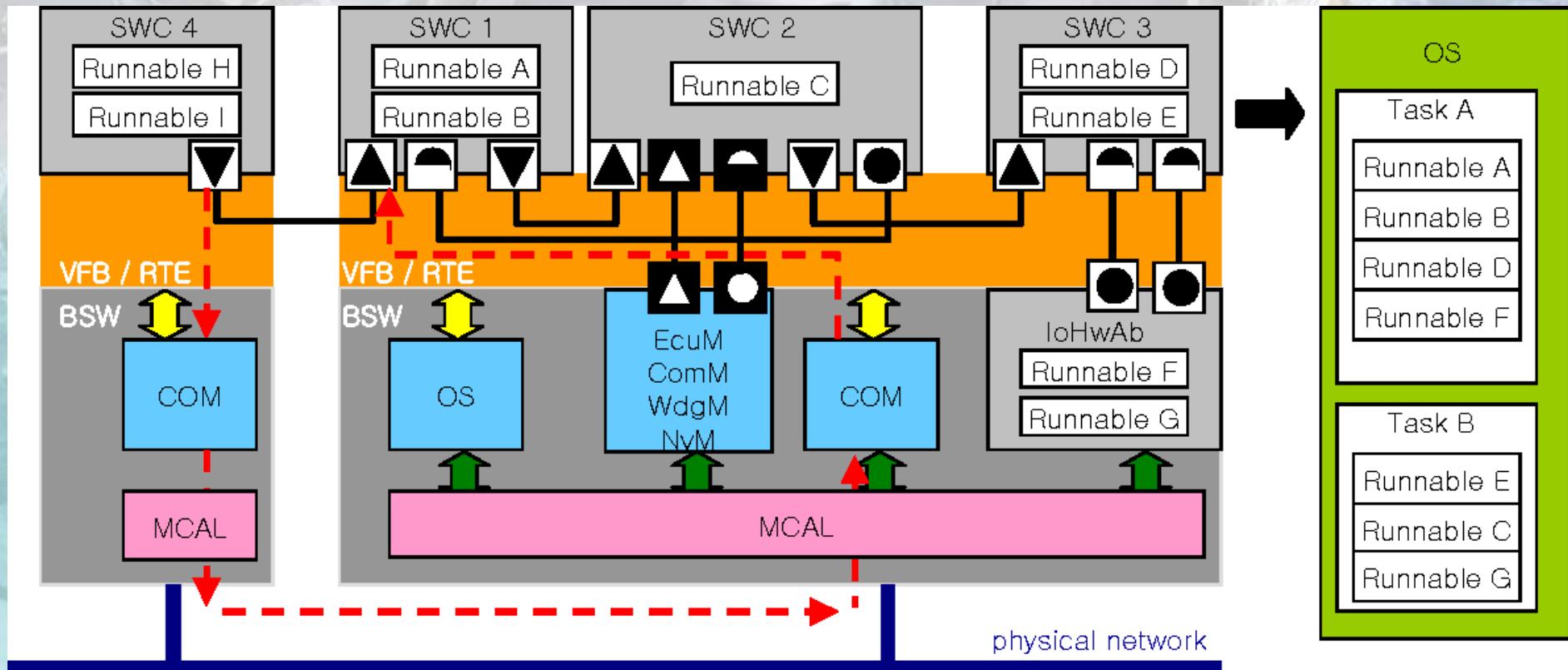
Use case ‘Front-Light Management’ in AUTOSAR on different ECUs



Source: www.autosar.org

<https://www.learn-in-depth.com/>
<https://www.facebook.com/groups/embedded.system.KS/>

Another Example



<https://www.learn-in-depth.com/>
<https://www.facebook.com/groups/embedded.system.KS/>



#LEARN_IN_DEPTH

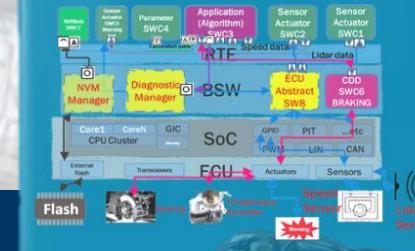
#Be_professional_in
embedded_system

eng. Keroles Shenouda

<https://www.facebook.com/groups/embedded.system.KS/>

AUTOSAR IN Depth Online Diploma

- ✓ Full lifetime access
- ✓ Certificate of completion
- ✓ Learning From Scratch



www.learn-in-depth.com
Eng. Keroles Shenouda

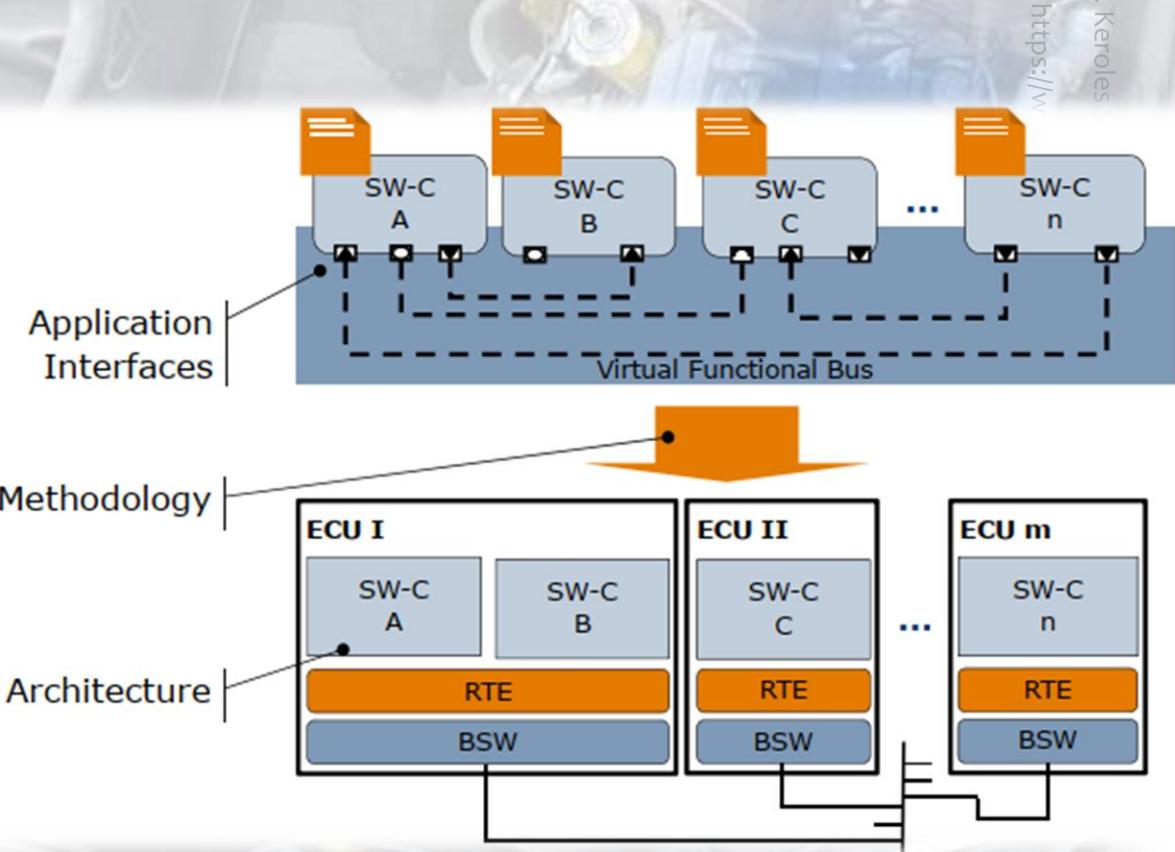
LEARN-IN-DEPTH
Be professional in
embedded system



<https://www.learn-in-depth.com/>
<https://www.facebook.com/groups/embedded.system.KS/>

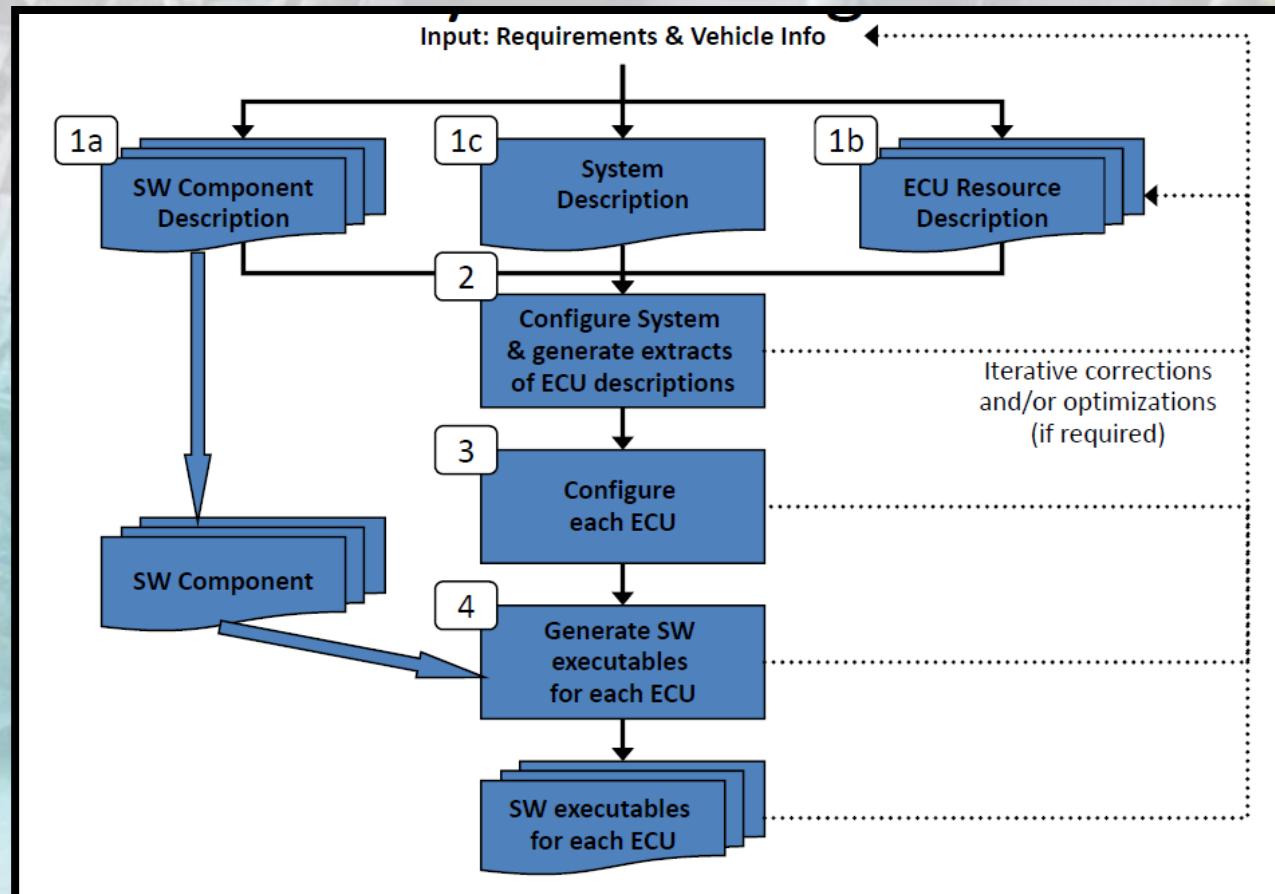
Application Layer

- ▶ The AUTOSAR software architecture enables the software developer to design an AUTOSAR application almost independent from the involved hardware.
- ▶ There is no knowledge about the network required, since the AUTOSAR software architecture and especially the RTE hide the network from the application.
- ▶ There is also nearly no knowledge about the used ECUs required, since the software architecture abstracts from the specific ECU and the controller on it.
- ▶ But there is of cause knowledge required about the sensors and actuators of an ECU which are used in the specific AUTOSAR application, so the software development can not be completely independent from the existent hardware components.



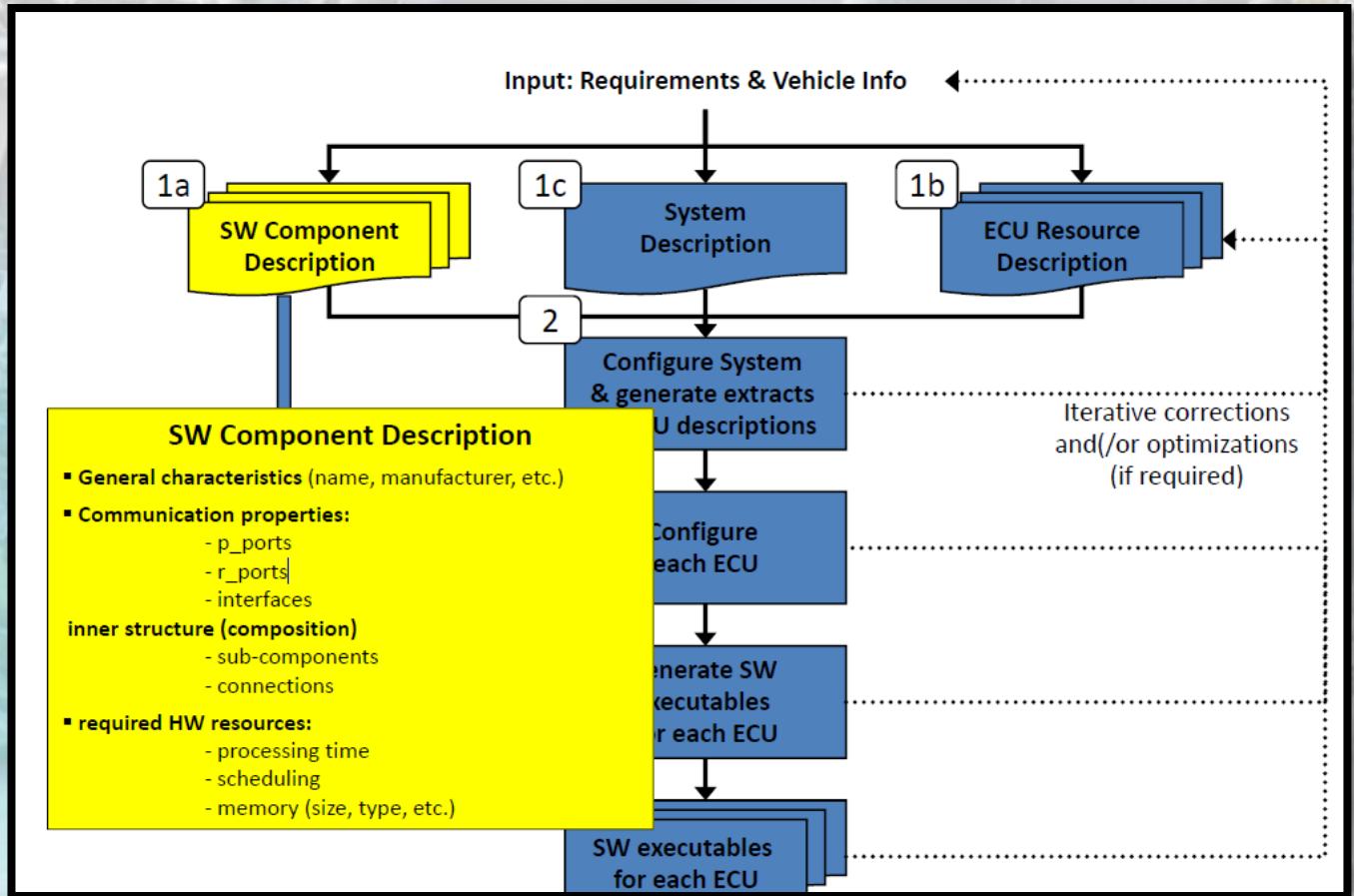
<https://www.learn-in-depth.com/>
<https://www.facebook.com/groups/embedded.system.KS/>

AUTOSAR System Design Process



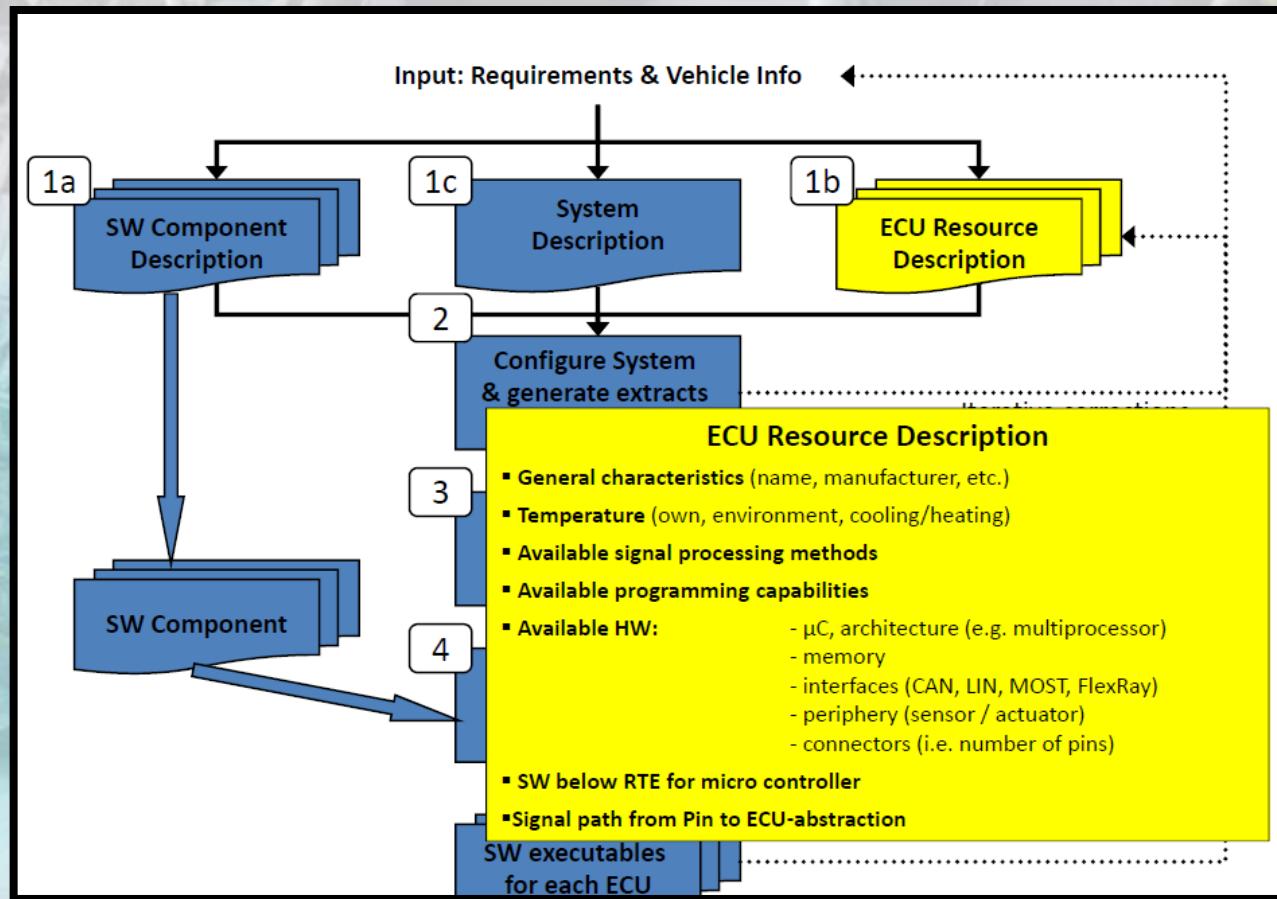
<https://www.learn-in-depth.com/>
<https://www.facebook.com/groups/embedded.system.KS/>

AUTOSAR System Design Process



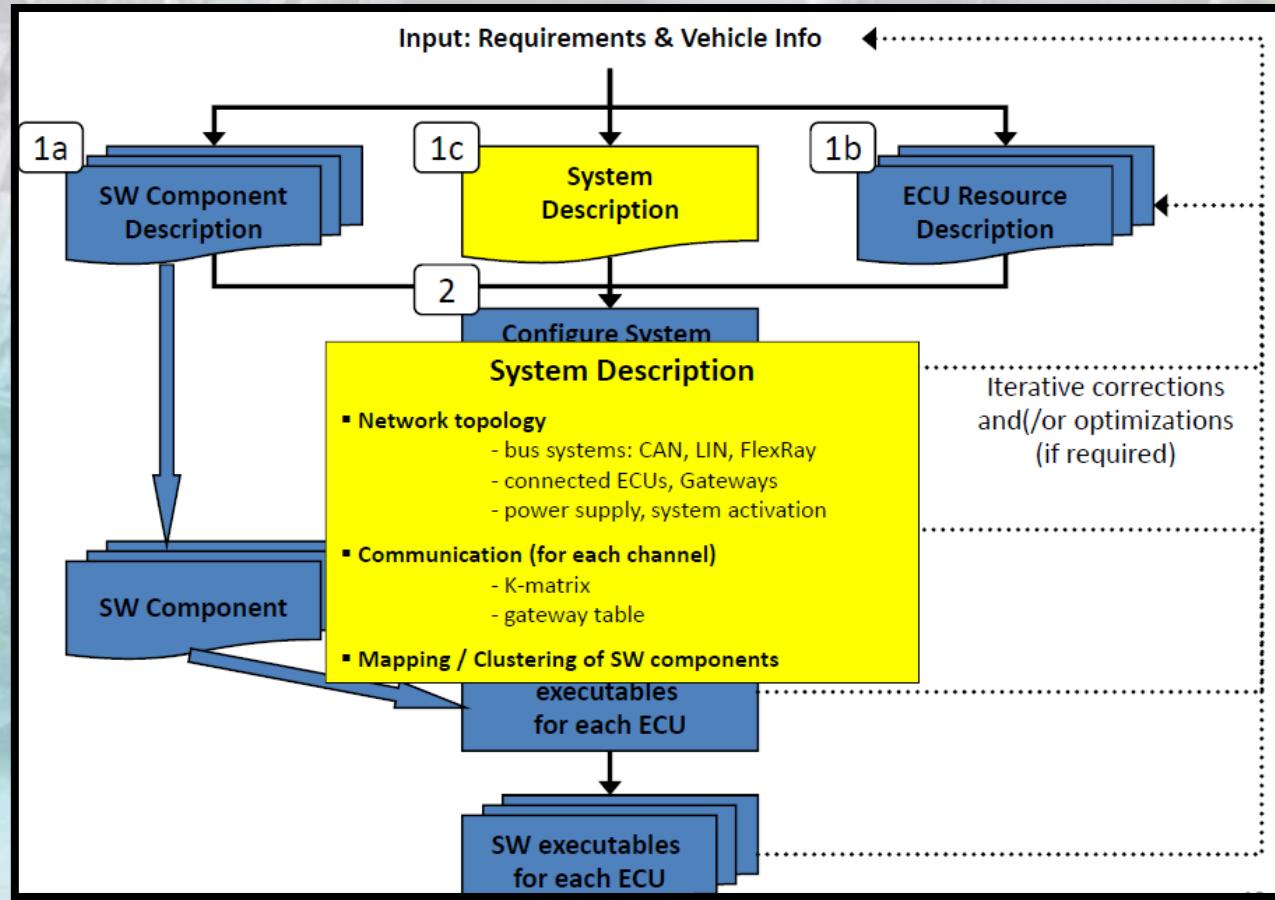
<https://www.learn-in-depth.com/>
<https://www.facebook.com/groups/embedded.system.KS/>

AUTOSAR System Design Process



<https://www.learn-in-depth.com/>
<https://www.facebook.com/groups/embedded.system.KS/>

AUTOSAR System Design Process



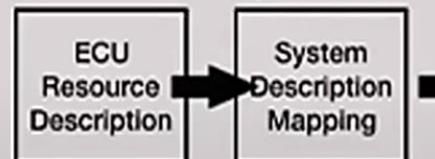
System Description c'est où on va faire le mapping, il comporte la description des outils et des protocoles à utilisés. (30:13)

<https://www.learn-in-depth.com/>
<https://www.facebook.com/groups/embedded.system.KS/>

Autosar Authoring Tool

VFB

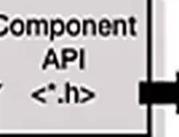
SWC-Description:
.arxml file



Mapping Details
(Page 16)



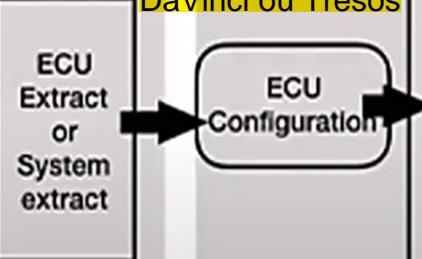
Ces fichiers contiennent les prototypes des fonctions nécessaires pour la communication entre les composants.



SWC Implementation Static Code



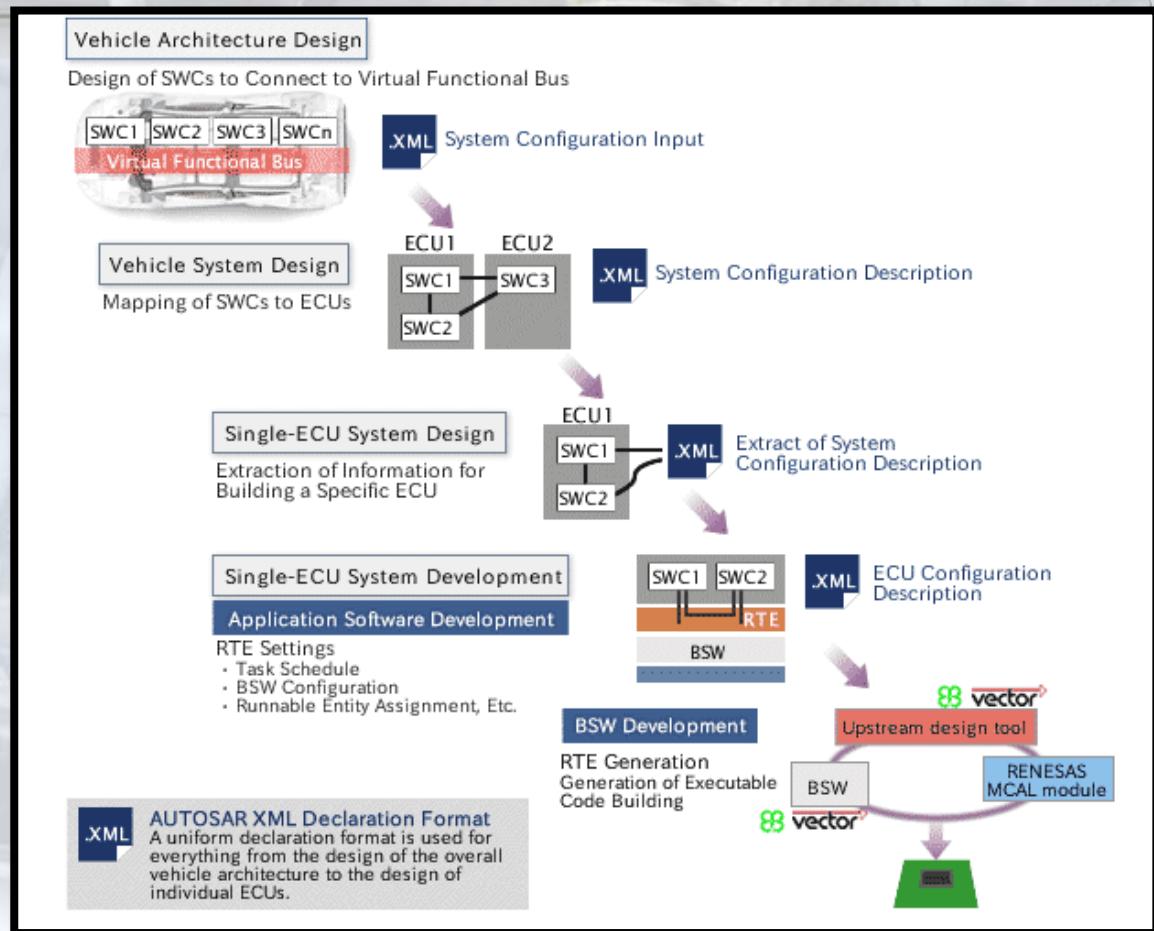
ECU1
ECU2
ECU3...
ECU Configuration pour chaque module en utilisant des outils comme DaVinci ou Tresos



Un fichier arxml qui comporte les détails système pour chaque ECU (Extraction du fichier System Des)

Design Sequence Using AUTOSAR Methodology

- In AUTOSAR, the entire sequence from upstream design to detailed design and mounting is standardized, and communication network, ECU, and microcontroller (MCU) settings are laid out and performed in order.



<https://www.learn-in-depth.com/>
<https://www.facebook.com/groups/embedded.system.KS/>

Design Sequence Using AUTOSAR Methodology

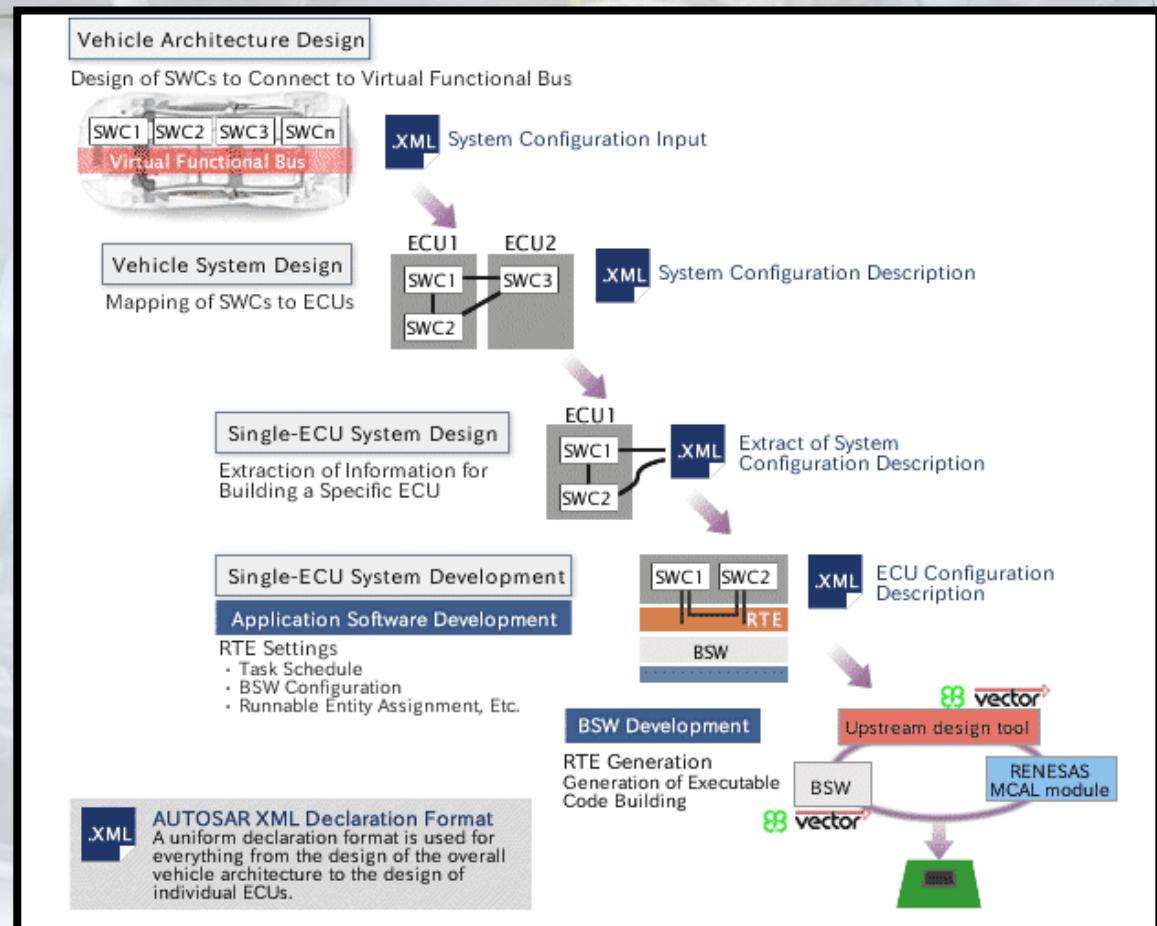
- ▶ Firstly, the vehicle architecture design is defined. In this step, the main task is to select
 - ▶ the software components and the hardware and the entire system constraints are declared.
 - ▶ Lastly, in this step, all the necessary information is filled into the system configuration input XML file using appropriate templates.

System constraints:

SWC-Description

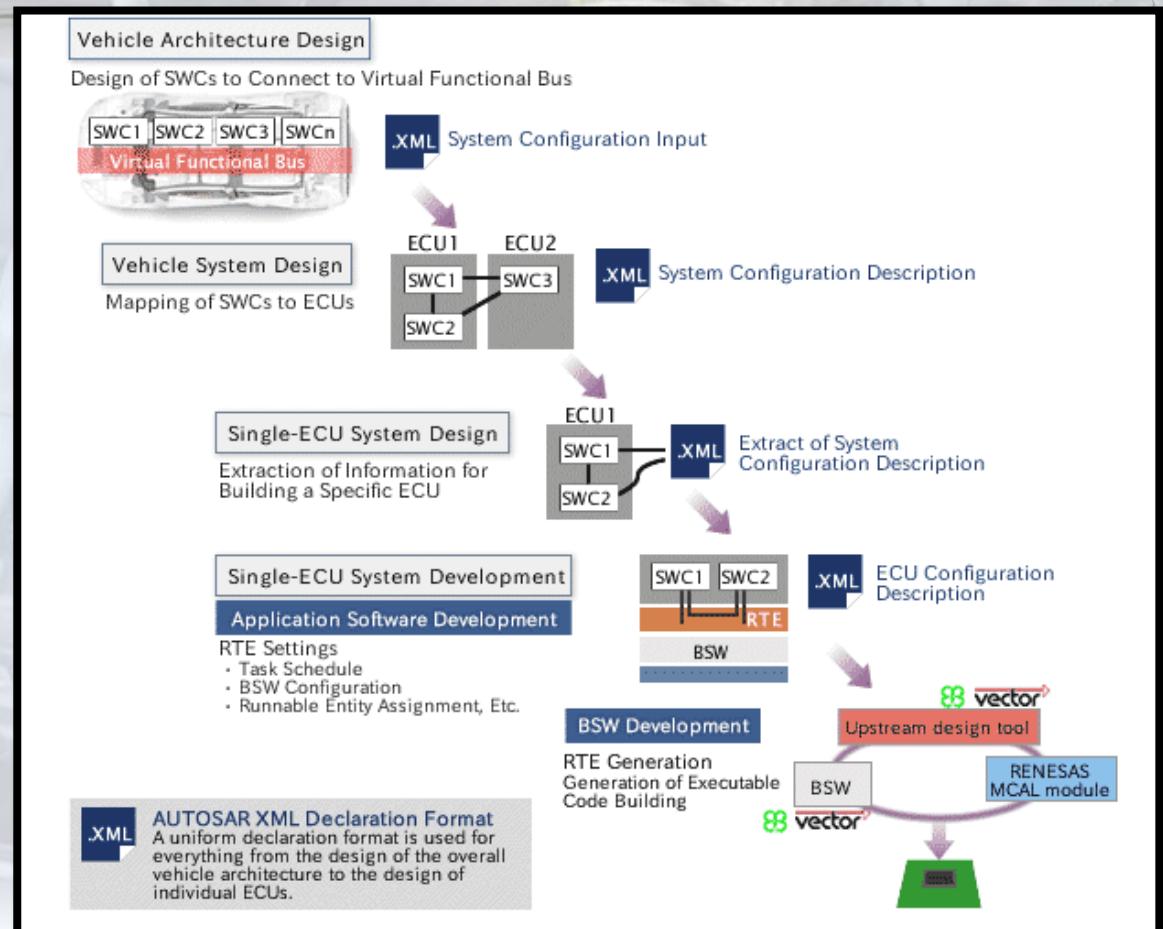
ECU Resources Description

System Description



Design Sequence Using AUTOSAR Methodology

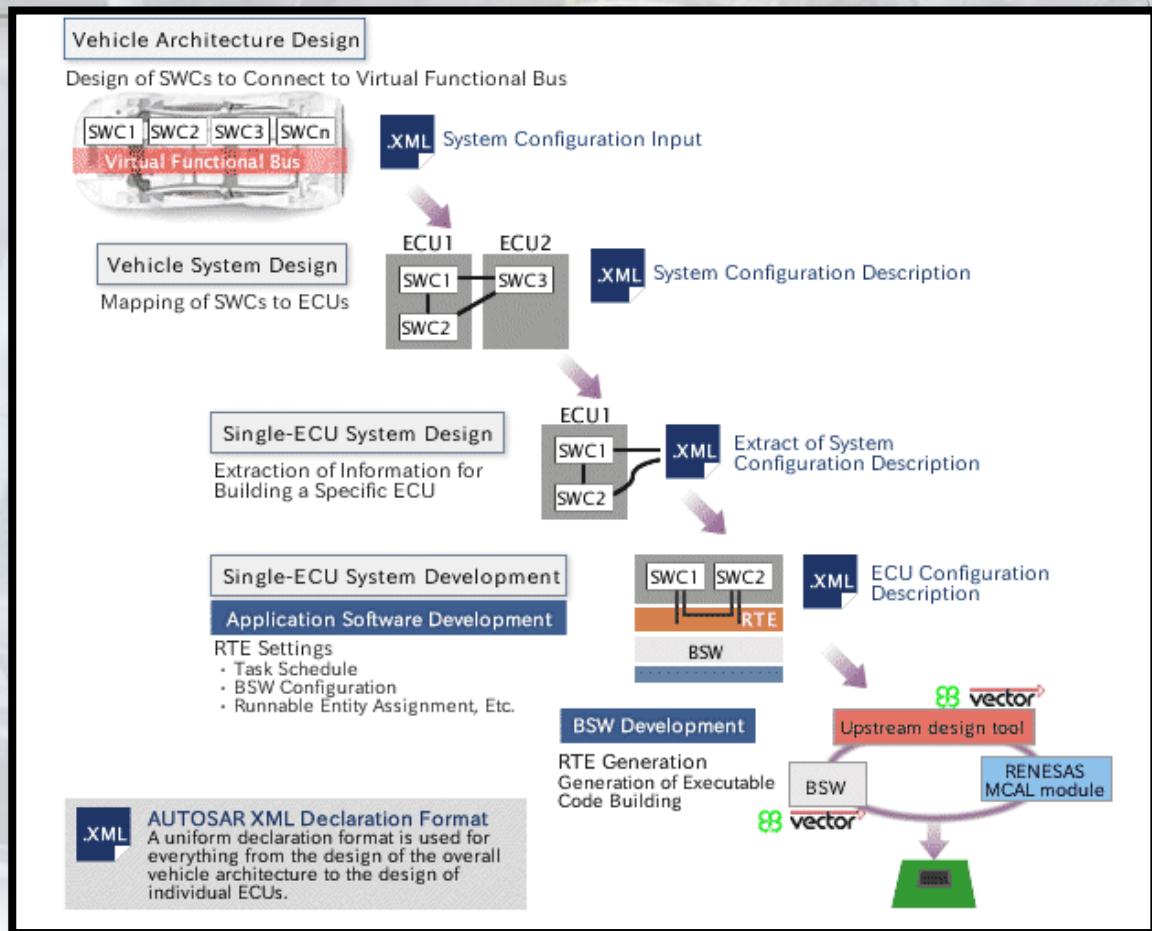
- ▶ The next step is vehicle system design.
 - ▶ The output of this step is a system configuration description XML file, which specifies the system information like bus mapping or topology and the mapping of SWCs to ECUs.
 - ▶ Further on, it comes to the stage of single ECU system design and the main task of this stage is to extract the information from the system configuration description file for building a specific ECU



<https://www.learn-in-depth.com/>
<https://www.facebook.com/groups/embedded.system.KS/>

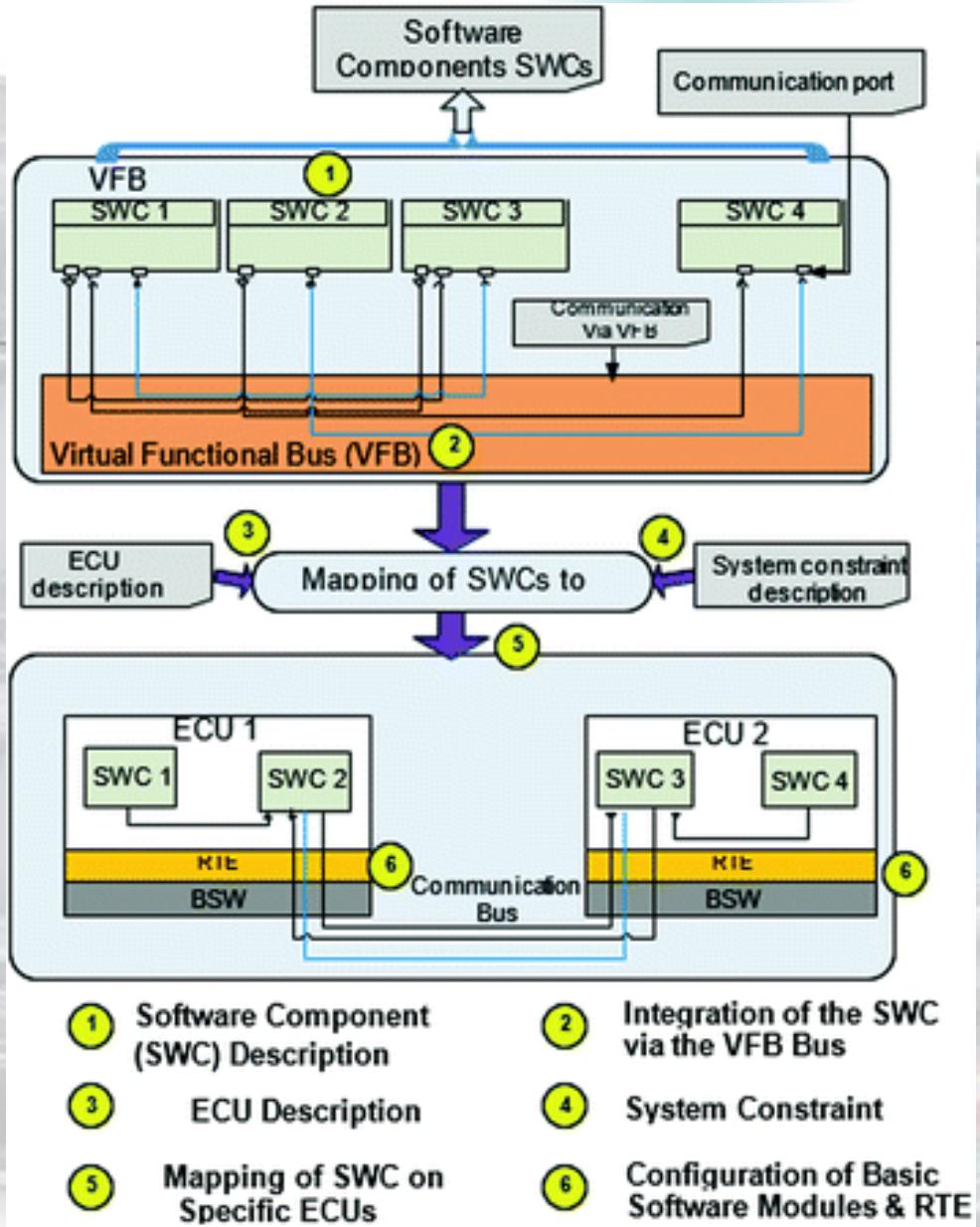
Design Sequence Using AUTOSAR Methodology

- ▶ The next stage is the single-ECU system development.
 - ▶ Here the application software is developed, including task schedule, the RTE settings, BSW configuration and Runnable entity assignment, etc.
- ▶ Last step is to generate an executable file and download it to the ECU.
 - ▶ Typically, this step involves code generation and compilation and links everything together into an executable.



<https://www.learn-in-depth.com/>
<https://www.facebook.com/groups/embedded.system.KS/>

Design Sequence Using AUTOSAR Methodology

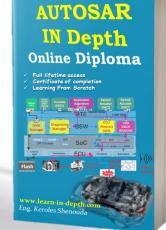


<https://www.learn-in-depth.com/>
<https://www.facebook.com/groups/embedded.system.KS/>





23



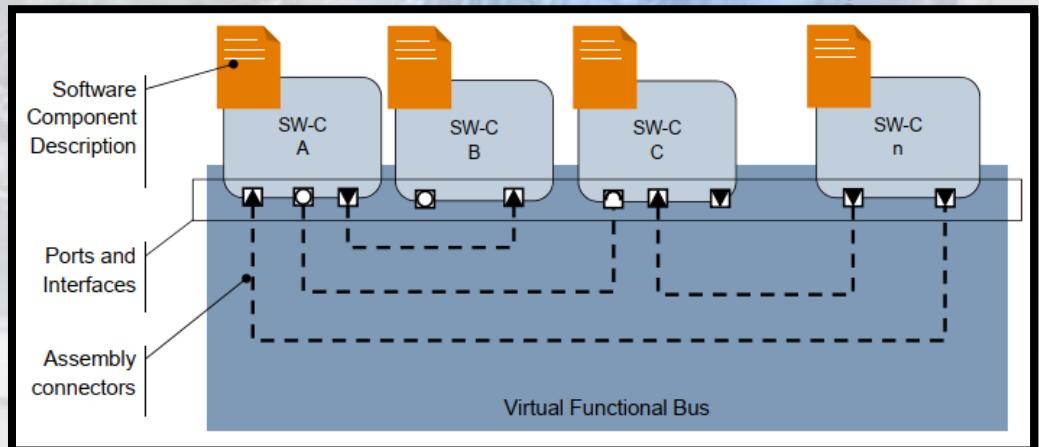
#LEARN_IN_DEPTH
#Be_professional_in
embedded_system

eng. Keroles Shenouda

<https://www.facebook.com/groups/embedded.system.KS/>

AUTOSAR defines four key concepts

- ▶ **Software components (SW-C)**
 - A piece of software to be run in an AUTOSAR system
- ▶ **Virtual Functional Bus (VFB)**
 - High level communication abstraction
- ▶ **Run Time Environment (RTE)**
 - Implements the VFB on one ECU
- ▶ **Basic Software (BSW)**
 - Standard software for standard ECU functionality (OS, communication, memory, hardware drivers, diagnostics etc)



<https://www.learn-in-depth.com/>
<https://www.facebook.com/groups/embedded.system.KS/>



24

#LEARN_IN_DEPTH
#Be_professional_in_embedded_system

eng. Keroles Shenouda

<https://www.facebook.com/groups/embedded.system.KS/>

AUTOSAR IN Depth Online Diploma

- ✓ Full lifetime access
- ✓ Certificate of completion
- ✓ Learning From Scratch



www.learn-in-depth.com
Eng. Keroles Shenouda

LEARN-IN-DEPTH
Be professional in
embedded system

<https://www.learn-in-depth.com/>
<https://www.facebook.com/groups/embedded.system.KS/>



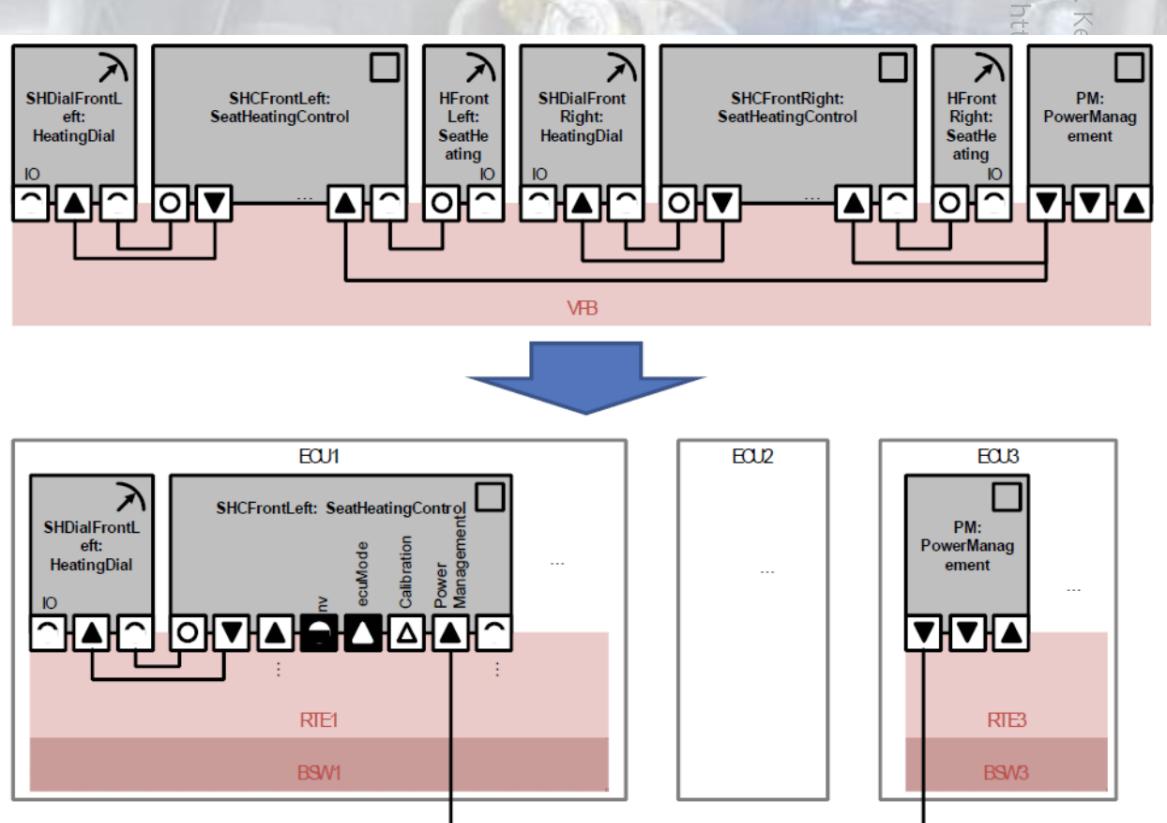
Virtual Functional Bus
AUTOSAR CP Release 4.4.0

Document Title	Virtual Functional Bus
Document Owner	AUTOSAR
Document Responsibility	AUTOSAR
Document Identification No	056
Document Status	Final
Part of AUTOSAR Standard	Classic Platform
Part of Standard Release	4.4.0

Virtual Functional Bus (VFB) / Runtime Environment (RTE)

Virtual Function Bus

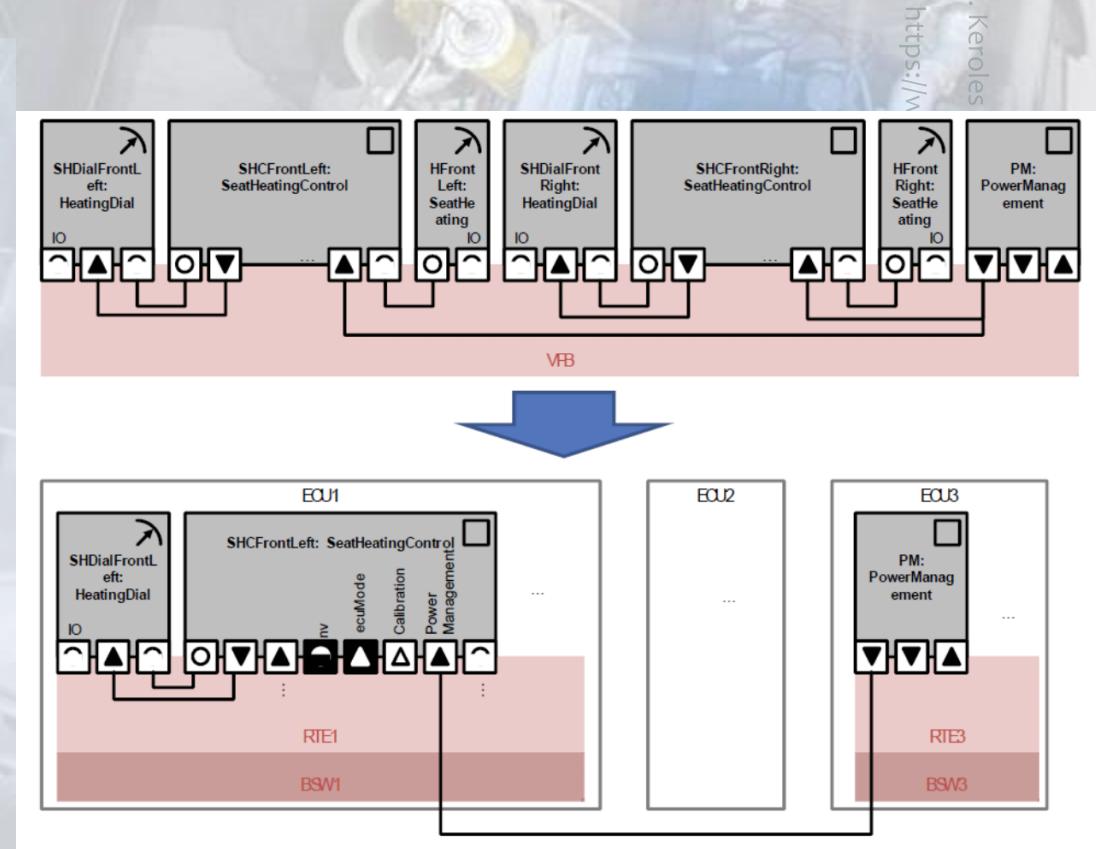
- ▶ From a general perspective, the virtual function bus can be described as a system modeling and communication concept.
- ▶ It is **logical** entity that facilitates the concept of relocatability within the AUTOSAR software architecture by providing a virtual infrastructure that is independent from any actual underlying infrastructure and provides all services required for a virtual interaction between AUTOSAR components
- ▶ It provides generic communication services that can be consumed by any existing AUTOSAR software component.
- ▶ Although any of these services are virtual, they will then in a later development phase be mapped to actual implemented methods, that are specific for the underlying hardware infrastructure.



<https://www.learn-in-depth.com/>
<https://www.facebook.com/groups/embedded.system.KS/>

Runtime Environment (RTE)

- ▶ In contrast to the purely virtual specification of the communication topology and interaction between components which is done via the virtual function bus,
- ▶ the runtime environment provides an **actual implementation** for these artifacts.
- ▶ It could also be said that the runtime environment provides **an actual representation of the virtual concepts of the VFB for one specific ECU.**



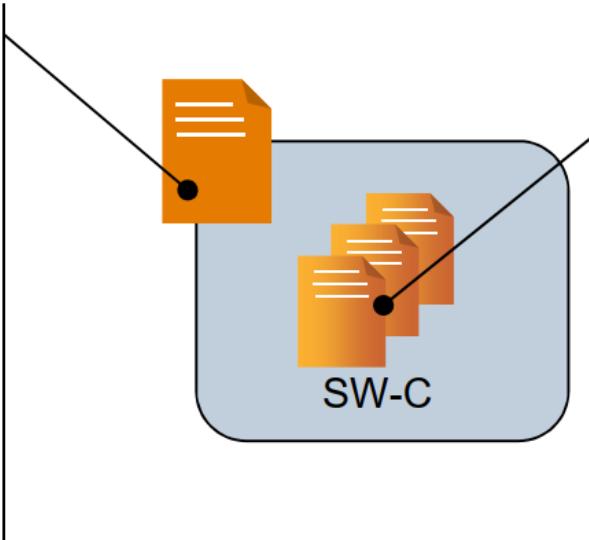
<https://www.learn-in-depth.com/>
<https://www.facebook.com/groups/embedded.system.KS/>

Software Component Description (SWCD)

- ▶ The Software Component Description (SWCD) is an XML file That completely define the SW-C (e.g. **ports**, **interfaces** and **behavior**)
- ▶ The SW-C contains an SWCD and the SW-C implementation

```
<?xml version="1.0" encoding="UTF-8"?>
<AUTOSAR xmlns="http://autosar.org/3.1.2">
  <TOP-LEVEL-PACKAGES>
    <AR-PACKAGE>
      <SHORT-NAME>MySwcDescription</SHORT-NAME>
      <ELEMENTS>

        ....
      </ELEMENTS>
    </AR-PACKAGE>
  </TOP-LEVEL-PACKAGES>
</AUTOSAR>
```



```
#include "MySwc.h"

Std_ReturnType fun1()
{
  /* Implementation */

  return E_OK;
}

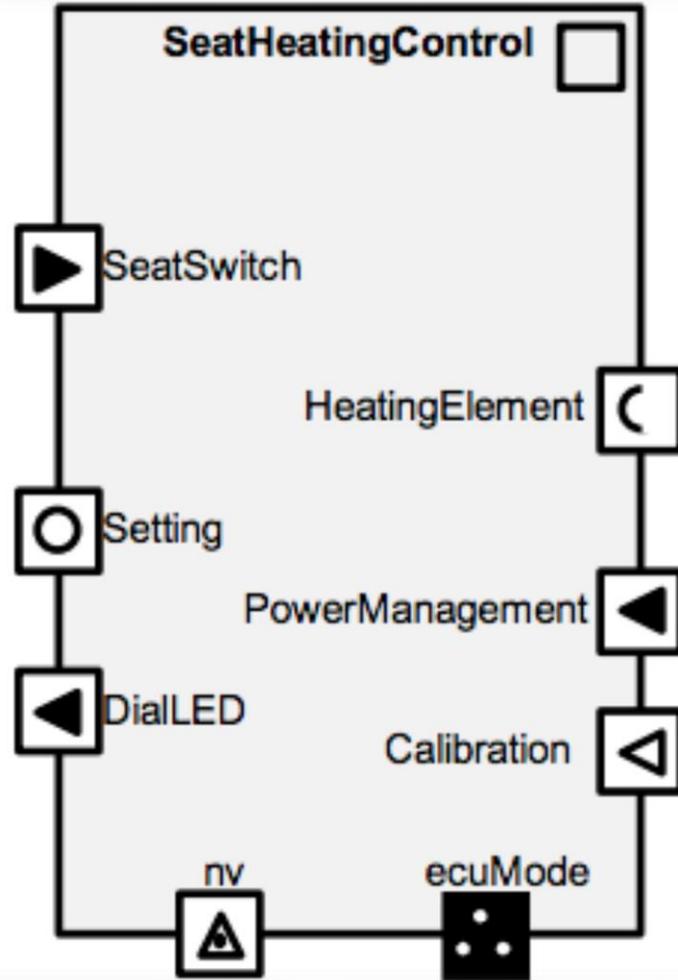
Std_ReturnType fun2()
{
  /* Implementation */
  ...
}
```

<https://www.learn-in-depth.com/>
<https://www.facebook.com/groups/embedded.system.KS/>

SWC

SWC Entity

- ▶ The purpose of the application layer is to provide the actual functionality of the system.
- ▶ This is done through the usage of **SWCs**, which are components containing software.



<https://www.learn-in-depth.com/>
<https://www.facebook.com/groups/embedded.system.KS/>



#LEARN_IN_DEPTH

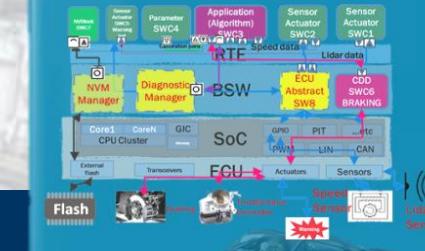
#Be_professional_in
embedded_system

eng. Keroles Shenouda

<https://www.facebook.com/groups/embedded.system.KS/>

AUTOSAR IN Depth Online Diploma

- ✓ Full lifetime access
- ✓ Certificate of completion
- ✓ Learning From Scratch



www.learn-in-depth.com
Eng. Keroles Shenouda

LEARN-IN-DEPTH
Be professional in
embedded system

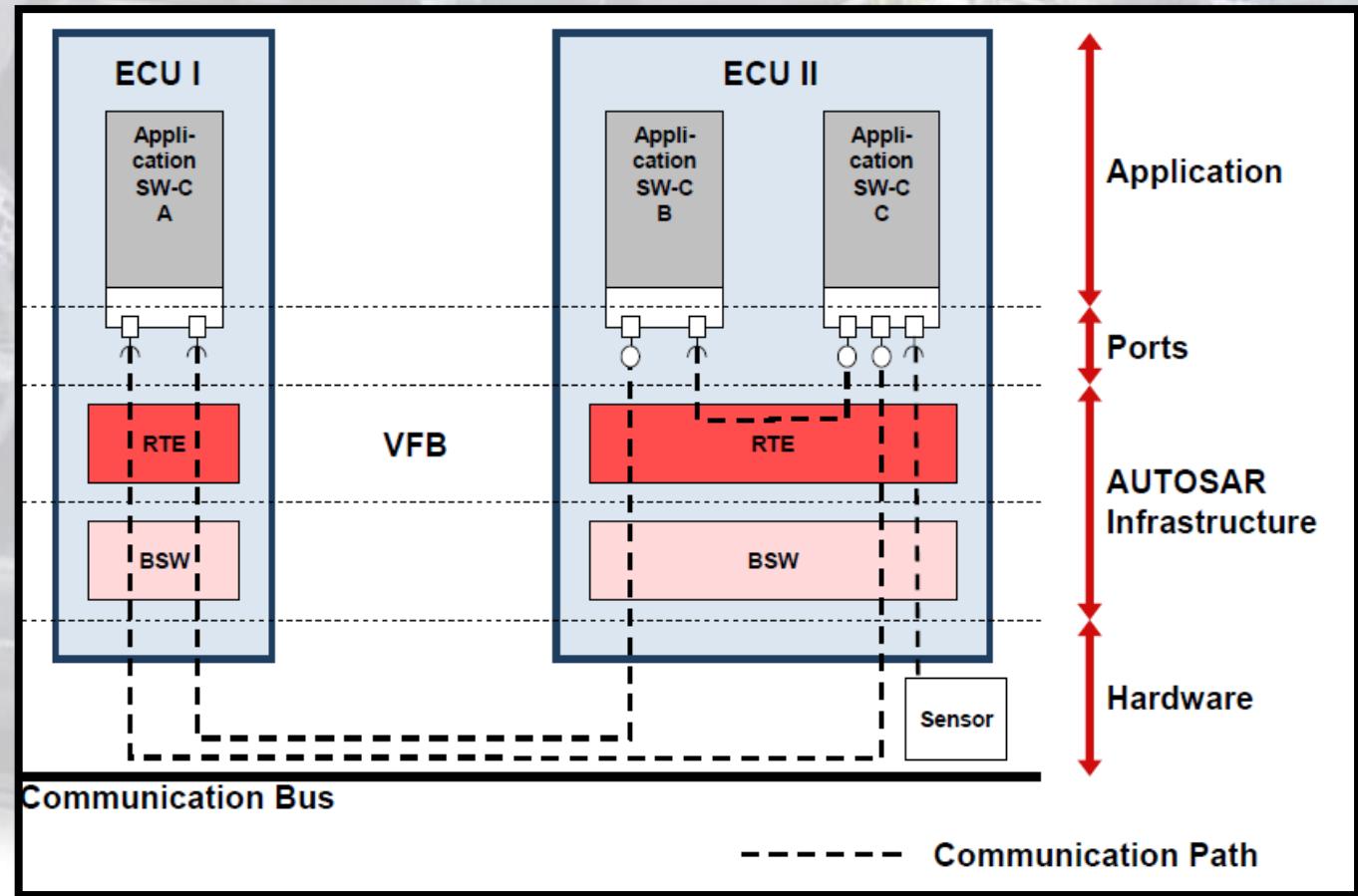


<https://www.learn-in-depth.com/>
<https://www.facebook.com/groups/embedded.system.KS/>

Intra- and Inter-ECU Communication

Software components

An **application** consists of one or more **SWCs** based in the Application layer. In order for SWCs to communicate with each other they use the virtual functional bus (**VFB**). From a SWC's point of view all it sees is the VFB and not the hardware dependent BSW and the hardware itself.



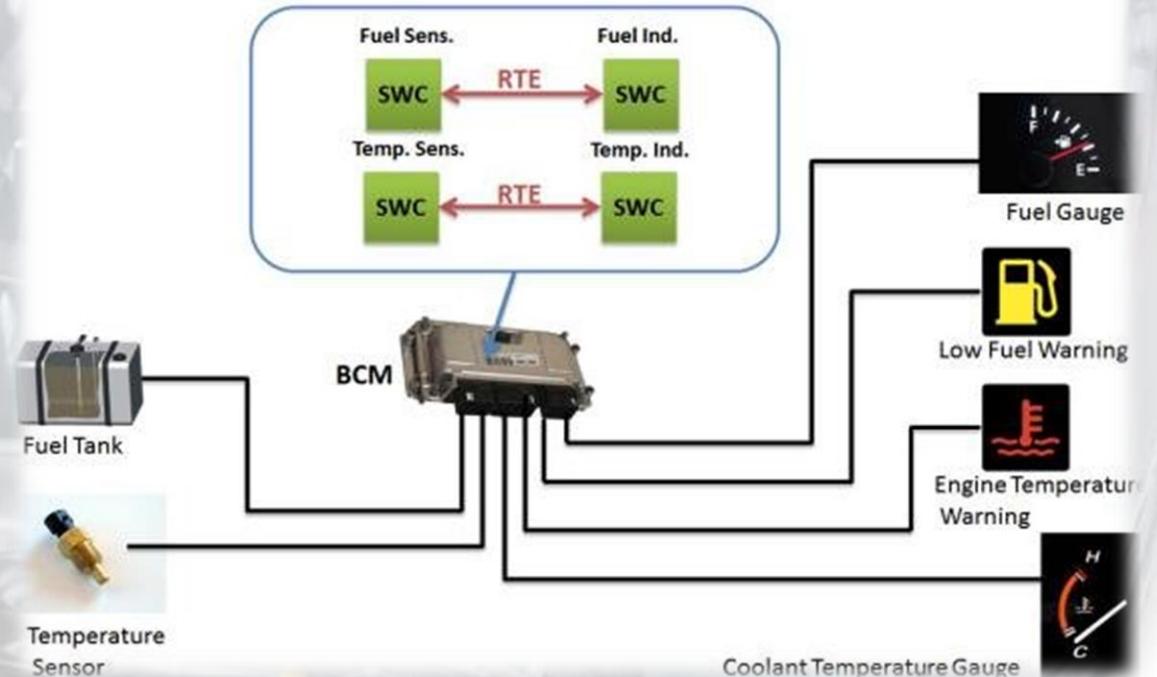
Source: www.autosar.org

<https://www.learn-in-depth.com/>
<https://www.facebook.com/groups/embedded.system.KS/>

intra-ECU communication

- In an Intra-ECU communication, all the SWCs that need to communicate with each other are available **over the same processor (ECU)**.
- The data exchange between applications occurs by means of regions inside of the chip (RAM memory).
- The implementation using Intra-ECU architecture was performed placing all SWCs (Fuel Sens., Fuel Ind., Temp. Sens. and Temp Ind.) in a single ECU connecting all the application via VFB (RTE+ BSW).
- The wire connection with temperature sensor, level sensor, gauges and warning lights are connected to a single ECU hardware.

INTRA-ECU communication

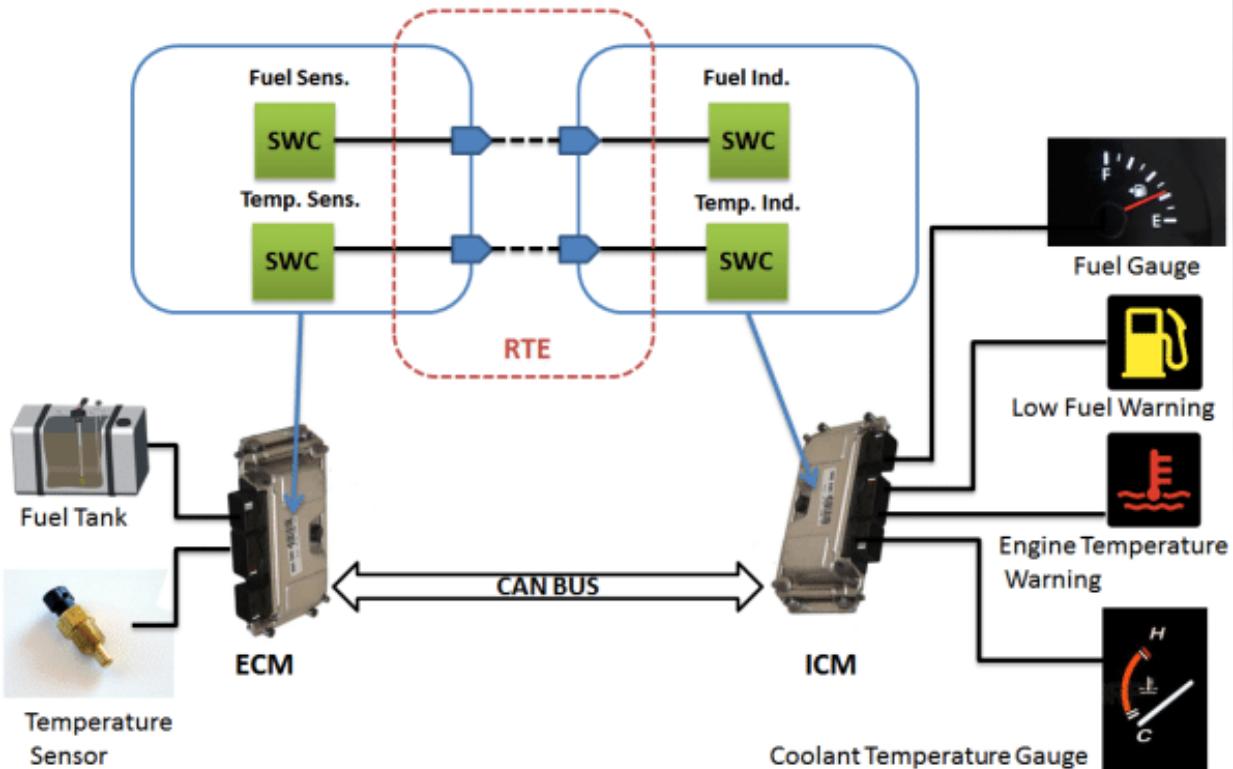


<https://www.learn-in-depth.com/>
<https://www.facebook.com/groups/embedded.system.KS/>

inter-ECU communication

- ▶ Inter-ECU communication is defined by the connection between two or more SWCs **placed inside of different ECUs**.
- ▶ The connection is performed via VFB that takes in advantage of the communication buses (**i.e. CAN, LIN, FlexRay, etc**) to software interconnection and makes possible the signals exchange over different hardware.

INTER-ECU communication



<https://www.learn-in-depth.com/>
<https://www.facebook.com/groups/embedded.system.KS/>



Conclusion

There are two different types of communication paths in the VFB

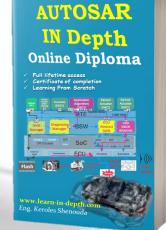
Intra-ECU (inside one ECU)

Inter-ECU (between different ECUs)

<https://www.learn-in-depth.com/>
<https://www.facebook.com/groups/embedded.system.KS/>



34

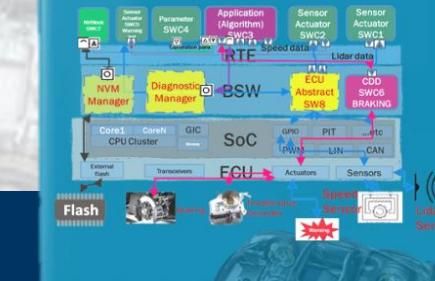
#LEARN_IN_DEPTH
#Be_professional_in_embedded_system

eng. Keroles Shenouda

<https://www.facebook.com/groups/embedded.system.KS/>

AUTOSAR IN Depth Online Diploma

- ✓ Full lifetime access
- ✓ Certificate of completion
- ✓ Learning From Scratch



www.learn-in-depth.com
Eng. Keroles Shenouda

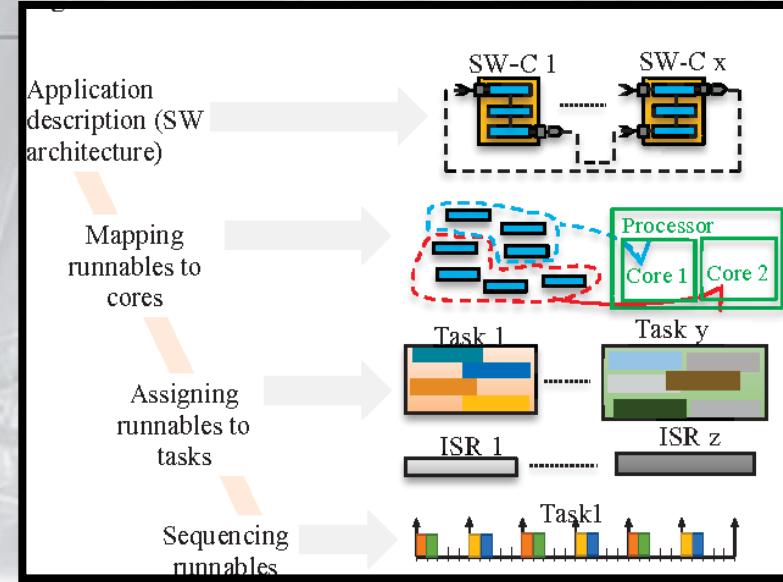
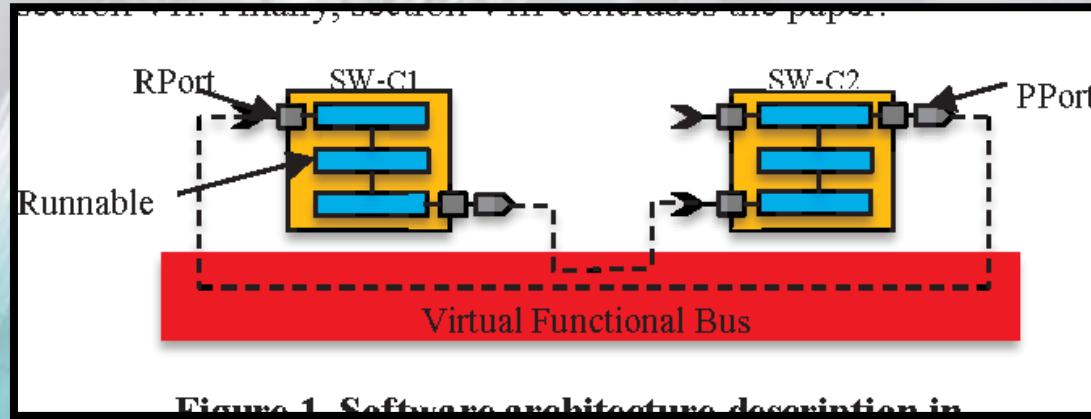
LEARN-IN-DEPTH
Be professional in
embedded system



<https://www.learn-in-depth.com/>
<https://www.facebook.com/groups/embedded.system.KS/>

(1:00:00)

VFB generate different RTE based on the INTER/INTRA ECU/Core Communication



{

Inter – ECU communication

Intra – ECU communication

 }

{

Inter – core communication

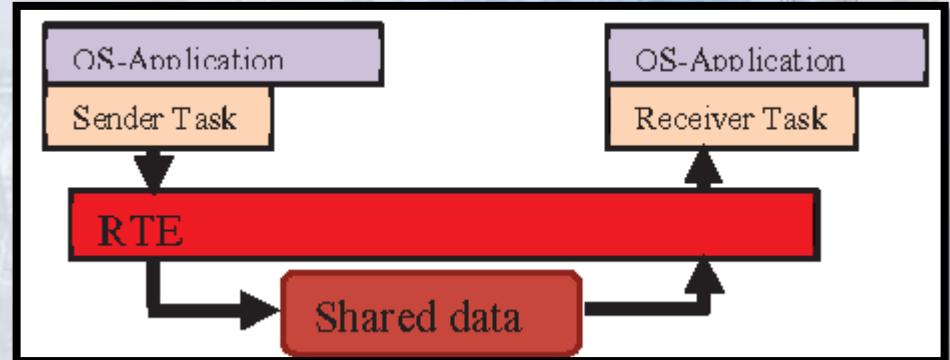
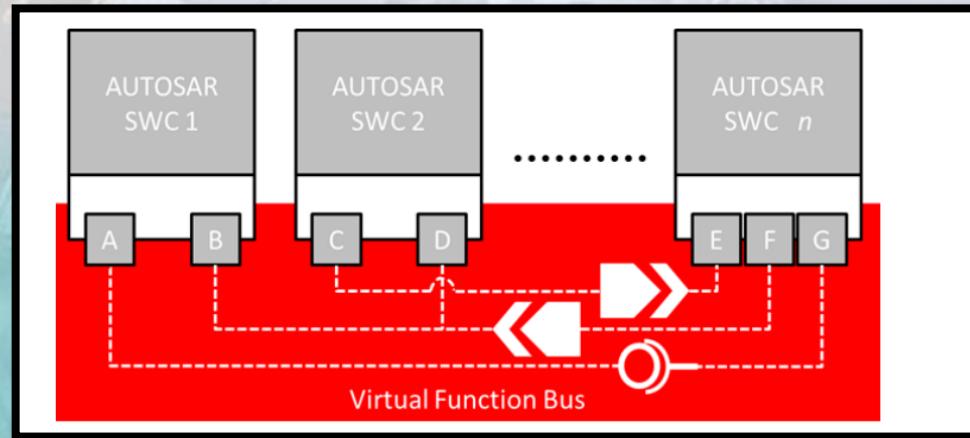
Intra – core communication

 }

(1:20:44)

<https://www.learn-in-depth.com/>
<https://www.facebook.com/groups/embedded.system.KS/>

Intra ECU Communication/ Intra Core Communication

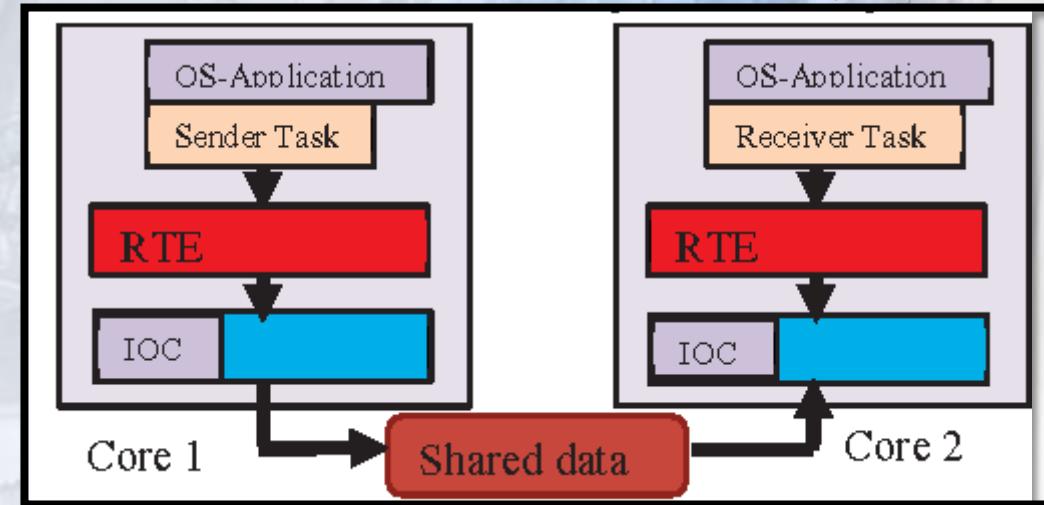
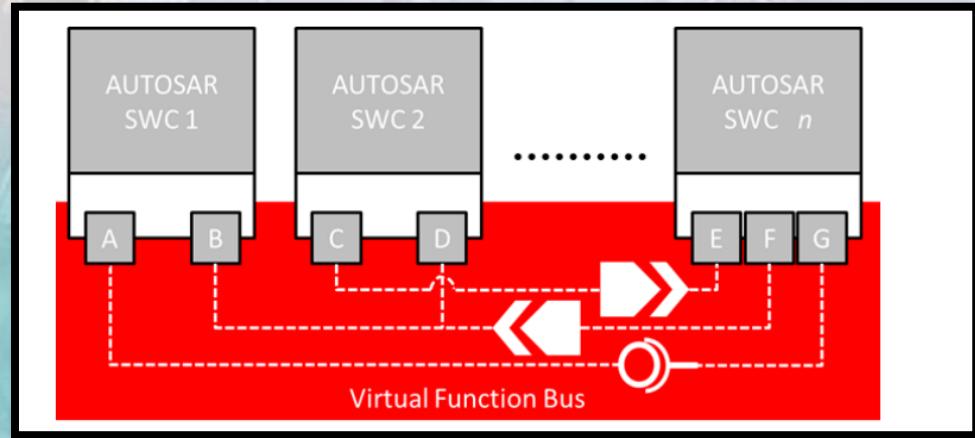


<https://www.facebook.com/groups/embedded.system.KS/>

<https://www.learn-in-depth.com/>

#LEARN_IN_DEPTH
#Be_professional_in
embedded_system

Intra ECU Communication/ Intre Core Communication



https://www.learn-in-depth.com/
Eng. Keroles Shenouda

#LEARN_IN_DEPTH
#Be_professional_in
embedded_system

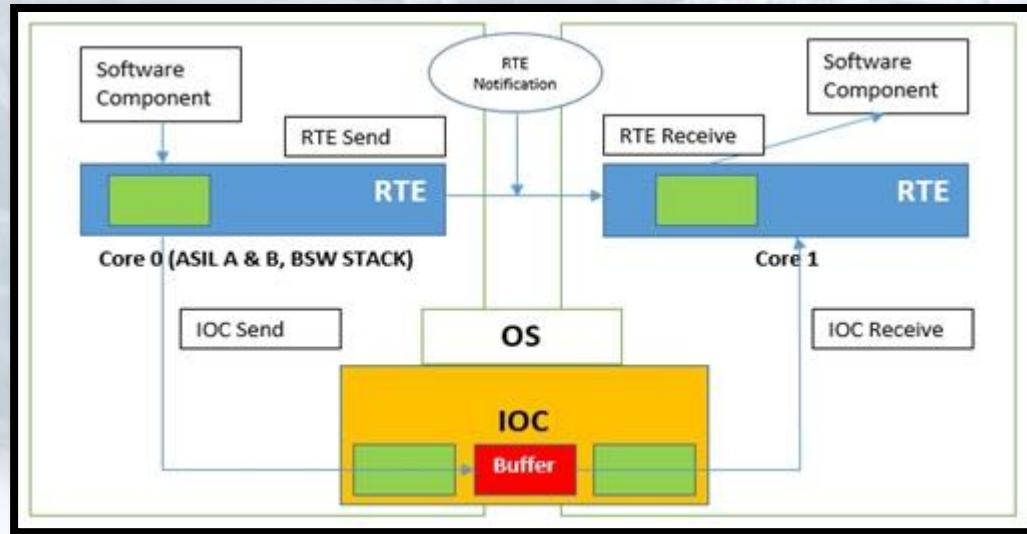
.KS/

<https://www.learn-in-depth.com/>
<https://www.facebook.com/groups/embedded.system.KS/>

IOC (Inter OS Communication)

- ▶ The “IOC” is responsible for the communication between OS-Applications and in particular for the communication crossing core or memory protection boundaries.
- ▶ Its internal functionality is closely connected to the Operating System.

1:24:36



<https://www.learn-in-depth.com/>
<https://www.facebook.com/groups/embedded.system.KS/>

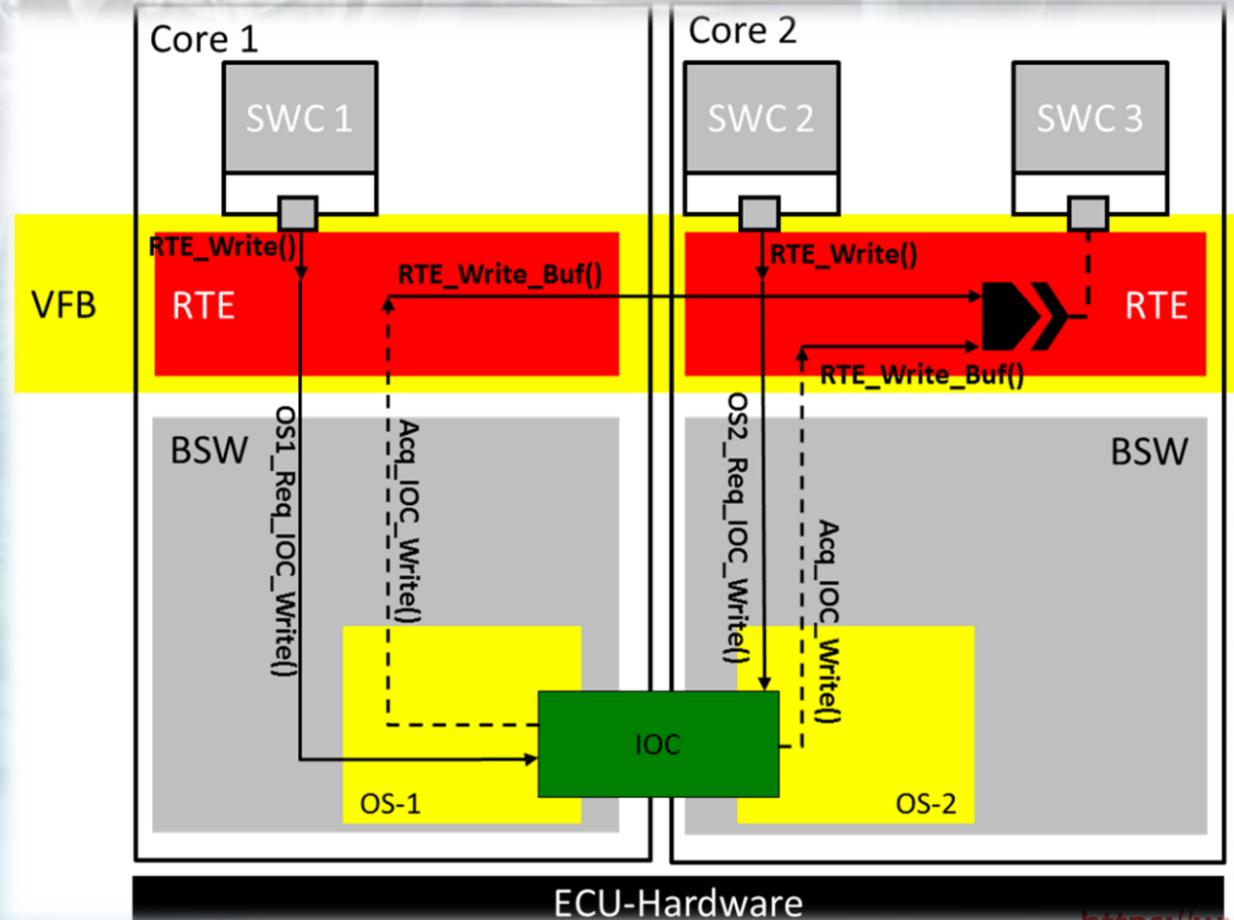
Example of “Custom Multicore AUTOSAR System”

- ▶ **RTE_Write_()**: A runnable in a SWC writes sends a data to the RTE layer with the destination port.
- ▶ **OS_Request_IOC_Write_()**: The RTE requests write permission on the destination port from the IOC module. Note that a race condition may occur if multiple calls are made to the IOC module.
- ▶ **Ack_IOC_Write_()**: IOC grants write permission to the RTE layer by acknowledging its request
- ▶ **RTE_Write_Buffer_()**: The RTE writes the data to the memory buffer location, which will be read by the receiver.
- ▶ **Ack_Completion**: The RTE acknowledges the IOC that it has finished its job and now releases the IOC module.
- ▶ **RTE_OK**: The RTE acknowledges the SWC that the write operation has completed successfully and the data is available to the reader.

<https://www.learn-in-depth.com/>
<https://www.facebook.com/groups/embedded.system.KS/>

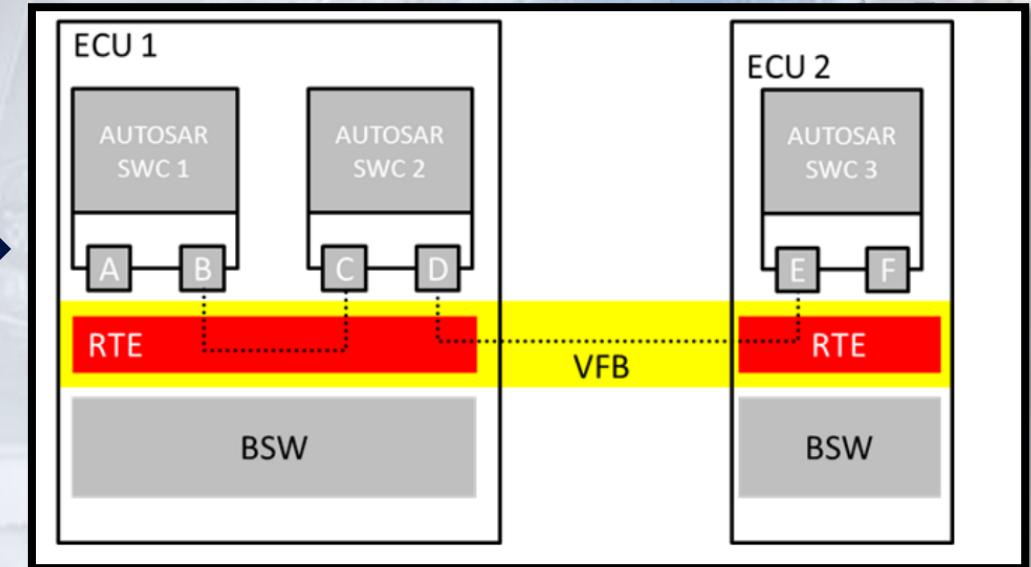
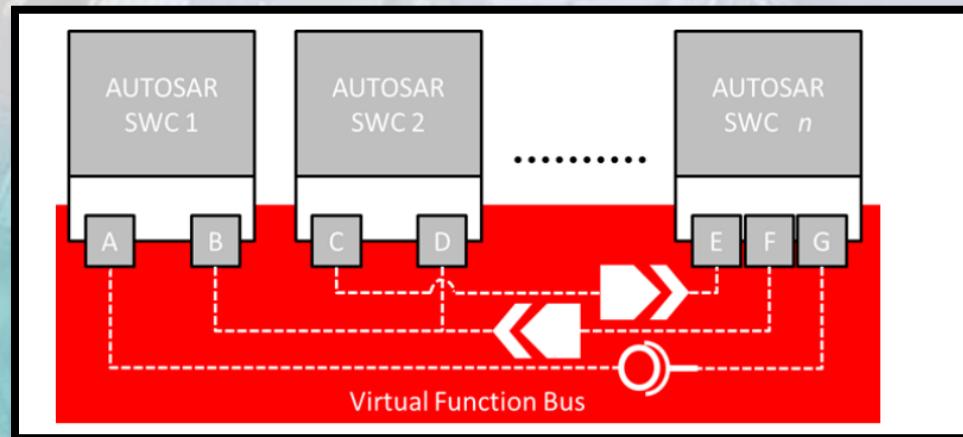
<https://www.facebook.com/groups/embedded.system.KS/>

Example of “Custom Multicore AUTOSAR System”



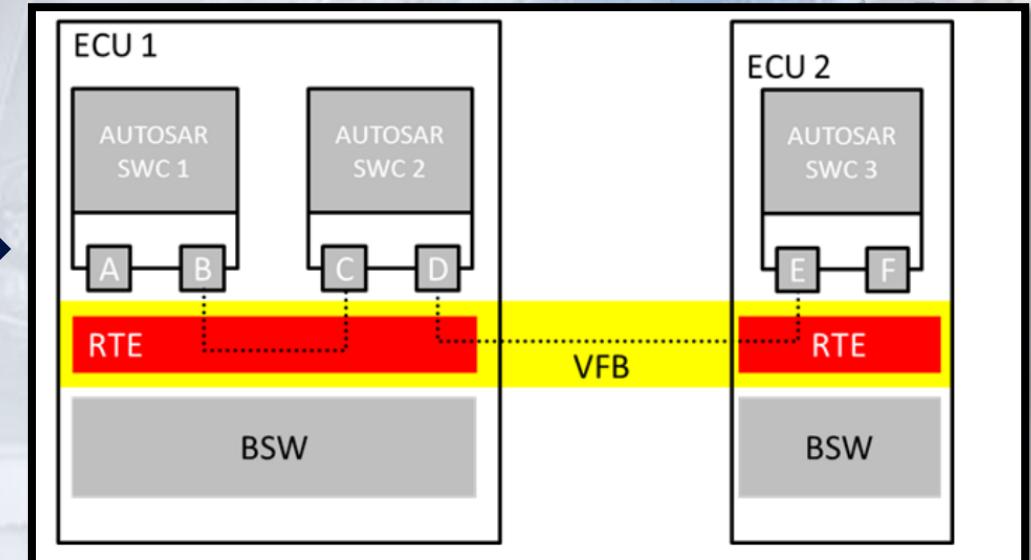
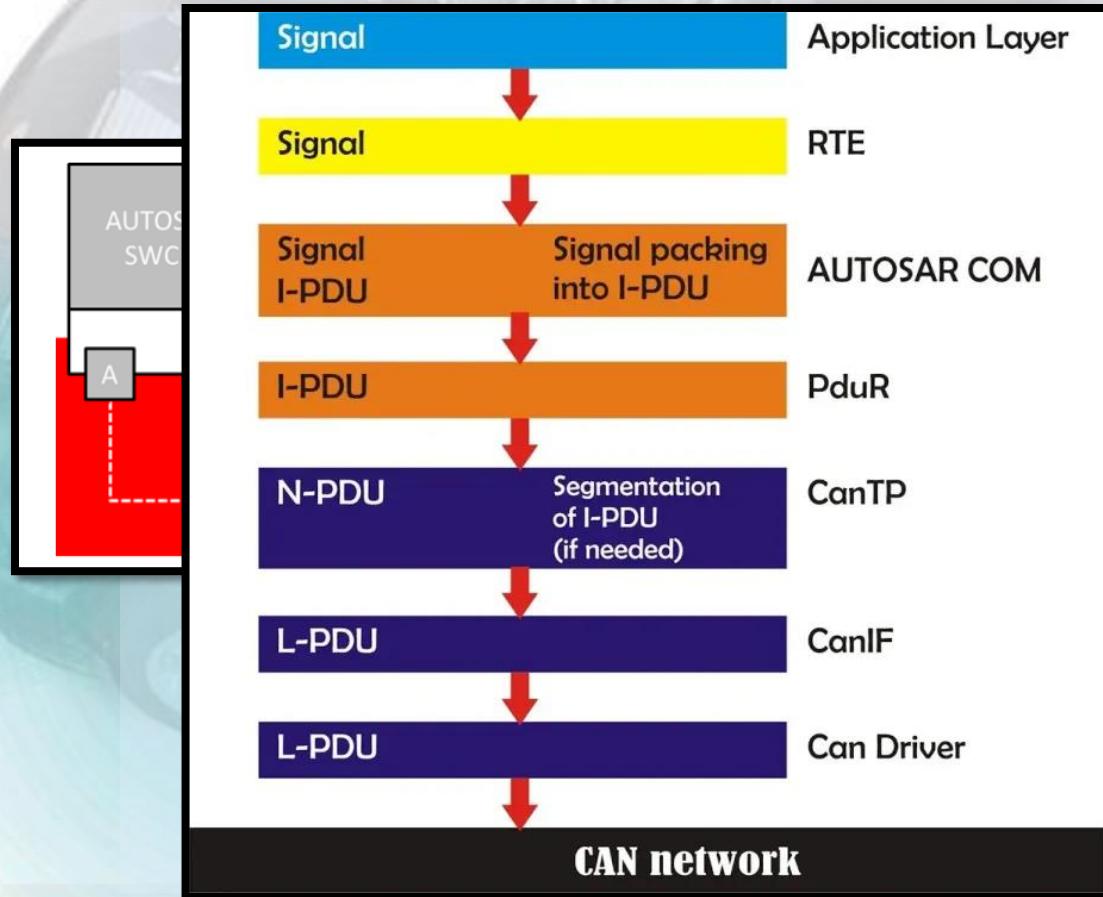
<https://www.learn-in-depth.com/>
<https://www.facebook.com/groups/embedded.system.KS/>

Inter ECU Communication



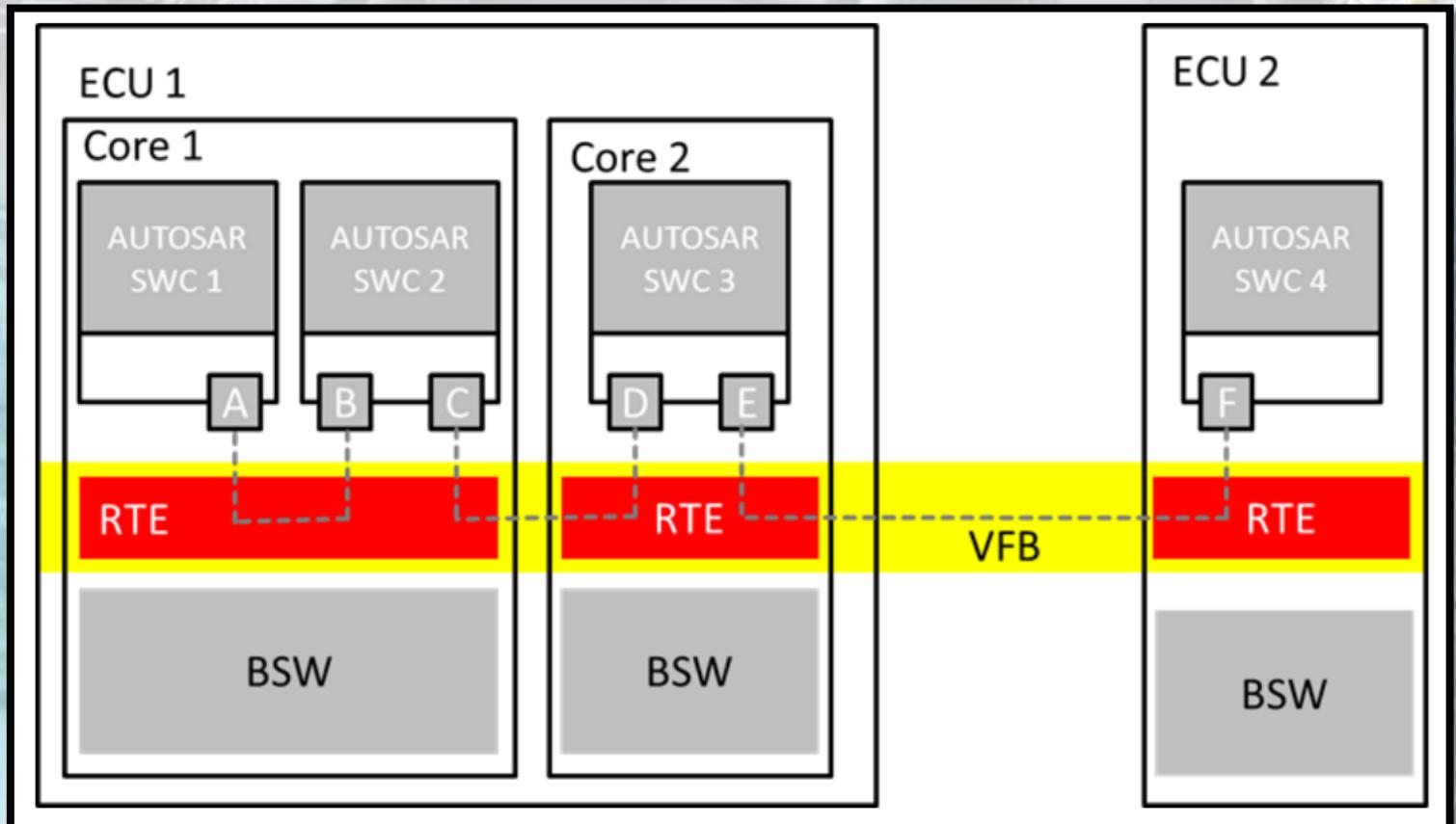
<https://www.learn-in-depth.com/>
<https://www.facebook.com/groups/embedded.system.KS/>

Inter ECU Communication

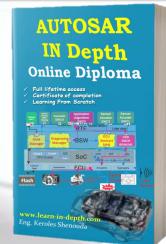


<https://www.learn-in-depth.com/>
<https://www.facebook.com/groups/embedded.system.KS/>

Mix between Inter/Intra ECU/Core Communication



<https://www.learn-in-depth.com/>
<https://www.facebook.com/groups/embedded.system.KS/>



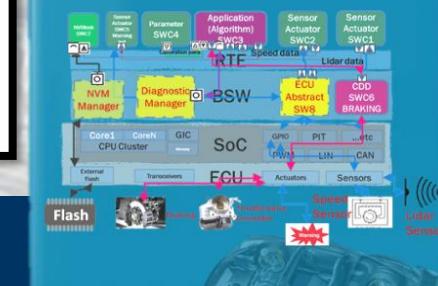
#Be_professional_in
embedded_system

eng. Keroles Shenouda

<https://www.facebook.com/groups/embedded.system.KS/>

AUTOSAR IN Depth Online Diploma

- ✓ Full lifetime access
- ✓ Certificate of completion
- ✓ Learning From Scratch



www.learn-in-depth.com
Eng. Keroles Shenouda

LEARN-IN-DEPTH
Be professional in
embedded system

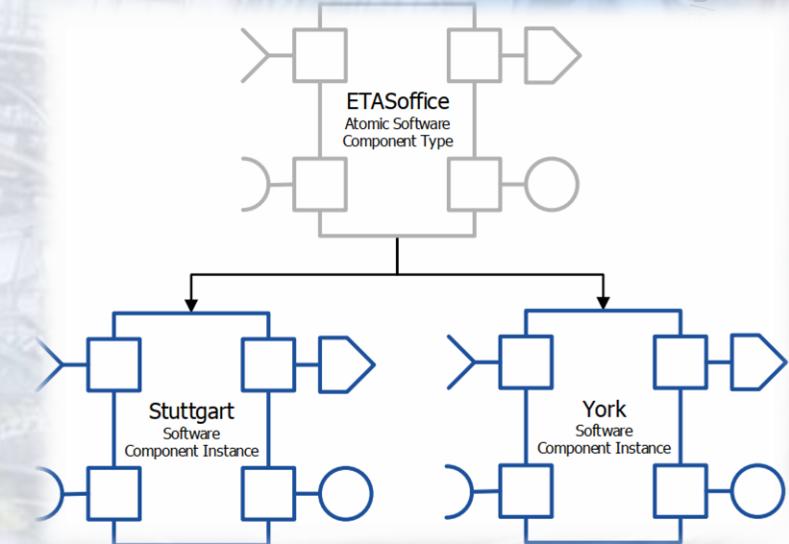
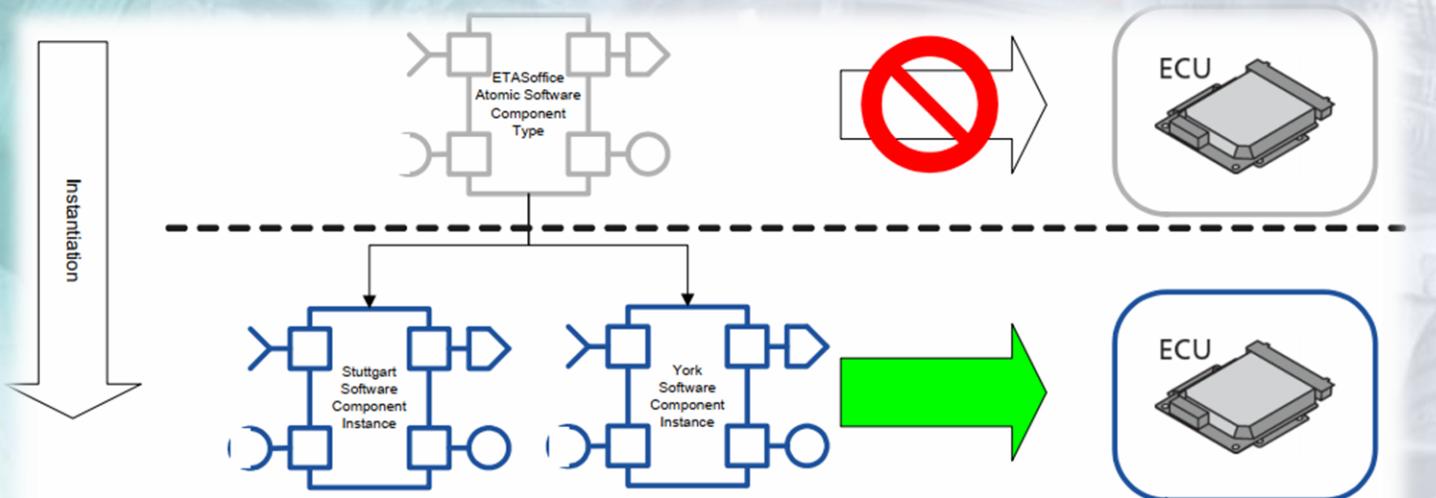


<https://www.learn-in-depth.com/>
<https://www.facebook.com/groups/embedded.system.KS/>

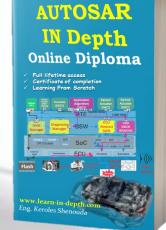
(1:31:00)

AUTOSAR SWC Template

- ▶ To allow the same component to be used it needs to be instantiated at configuration time.
- ▶ Furthermore, it is possible to configure a software component so that multiple instantiation is possible.

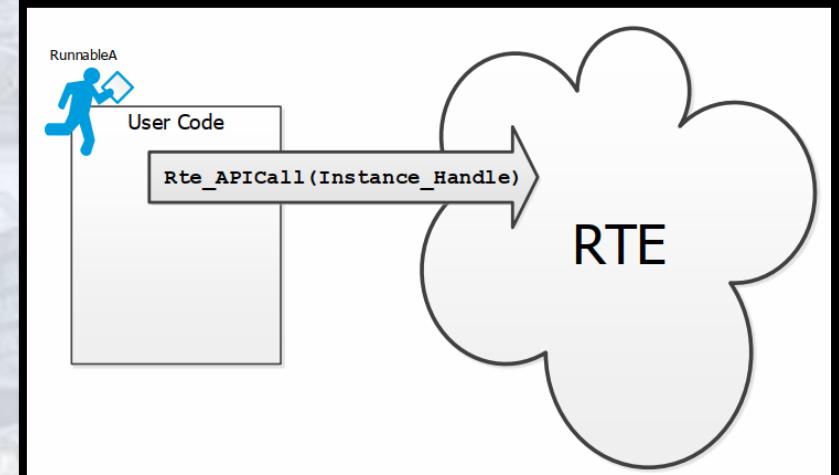
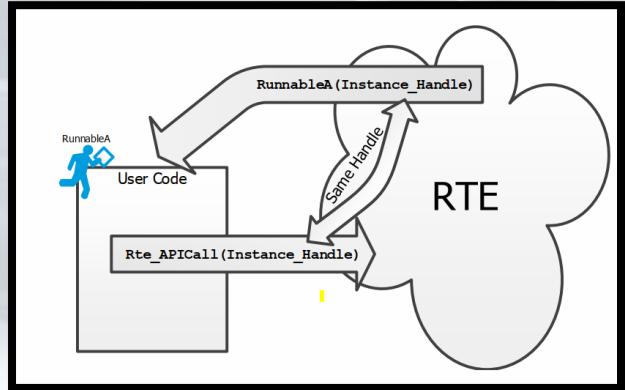


<https://www.learn-in-depth.com/>
<https://www.facebook.com/groups/embedded.system.KS/>



AUTOSAR SWC Template Cont.

- ▶ When there are multiple instances of a software component all instances of a component type share **the same code**, but each instance may have **private state** that is not shared between different instances of the component
- ▶ Each time you call an RTE service you must pass **in the instance handle** so that the component type code knows which component instance data



<https://www.learn-in-depth.com/>
<https://www.facebook.com/groups/embedded.system.KS/>



47

#LEARN_IN_DEPTH

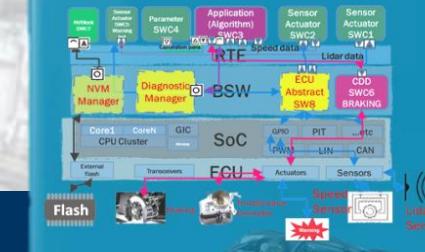
#Be_professional_in_embedded_system

eng. Keroles Shenouda

<https://www.facebook.com/groups/embedded.system.KS/>

AUTOSAR IN Depth Online Diploma

- ✓ Full lifetime access
- ✓ Certificate of completion
- ✓ Learning From Scratch



www.learn-in-depth.com
Eng. Keroles Shenouda

LEARN-IN-DEPTH
Be professional in
embedded system

<https://www.learn-in-depth.com/>
<https://www.facebook.com/groups/embedded.system.KS/>

AUTOSAR SWC (Implementation Sequence)



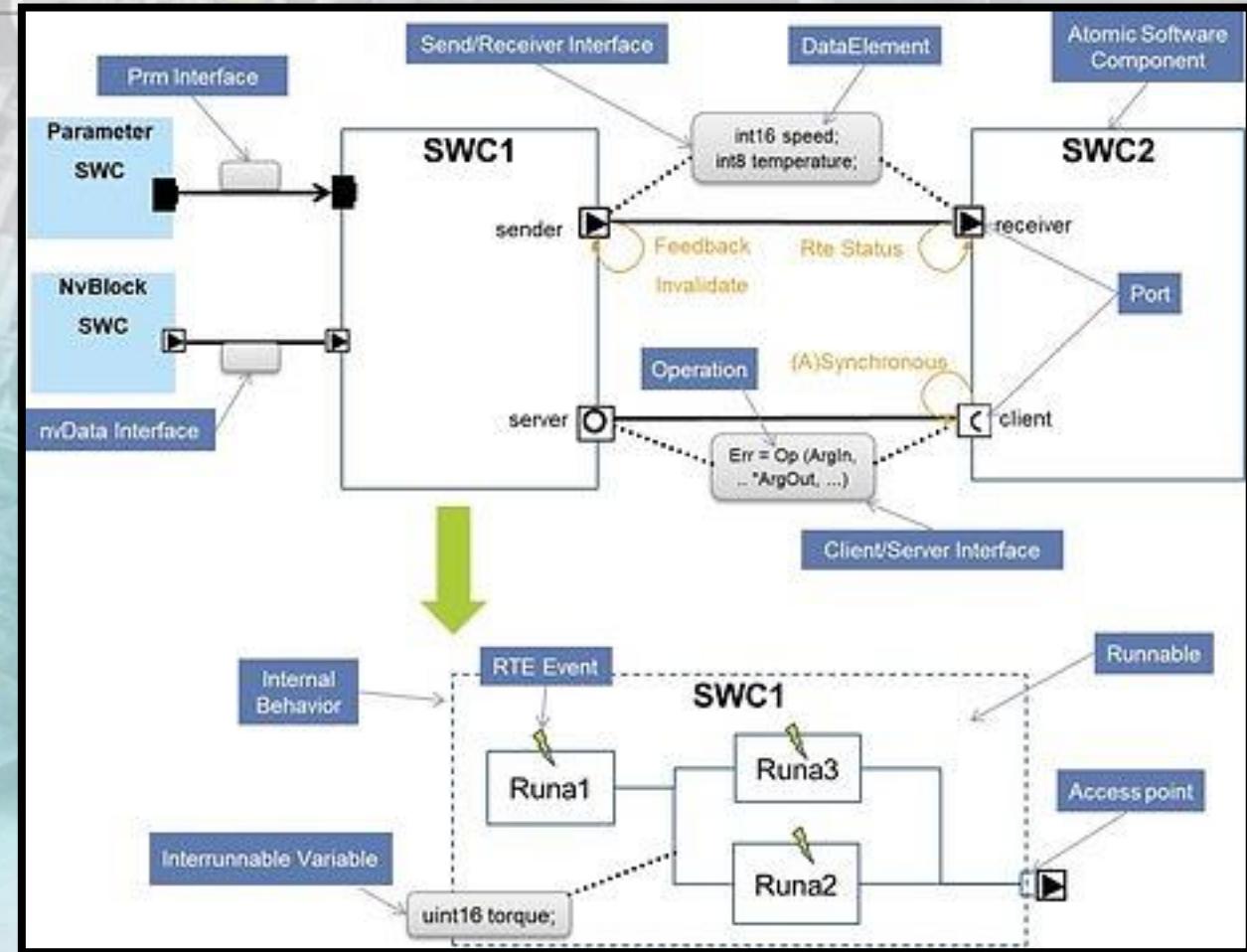
48

#LEARN_IN_DEPTH
#Be_professional_in_embedded_system

eng. Keroles Shenouda

<https://www.facebook.com/groups/embedded.system.KS/>

AUTOSAR SWC (Implementation Sequence)



learn-in-depth.com/

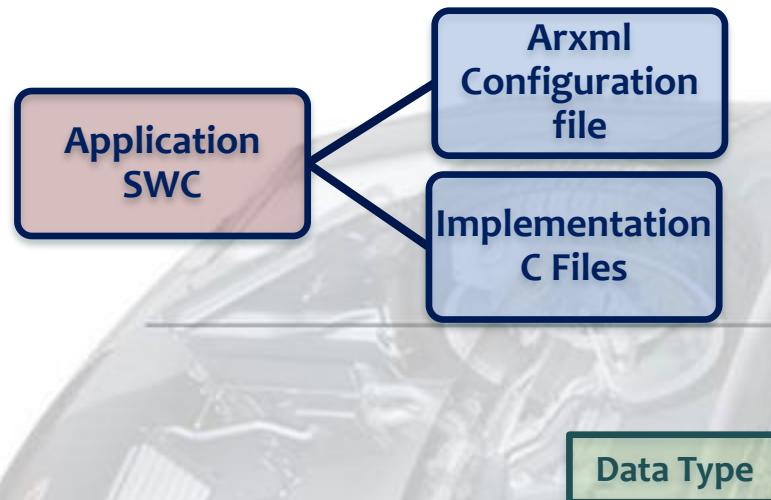
<https://www.facebook.com/groups/embedded.system.KS/>



49

#LEARN_IN_DEPTH
#Be_professional_in_embedded_system

eng. Keroles Shenouda

<https://www.facebook.com/groups/embedded.system.KS/>

- Data Types
 - Application Data Types
 - Base Types
 - Implementation Data Types
 - ...

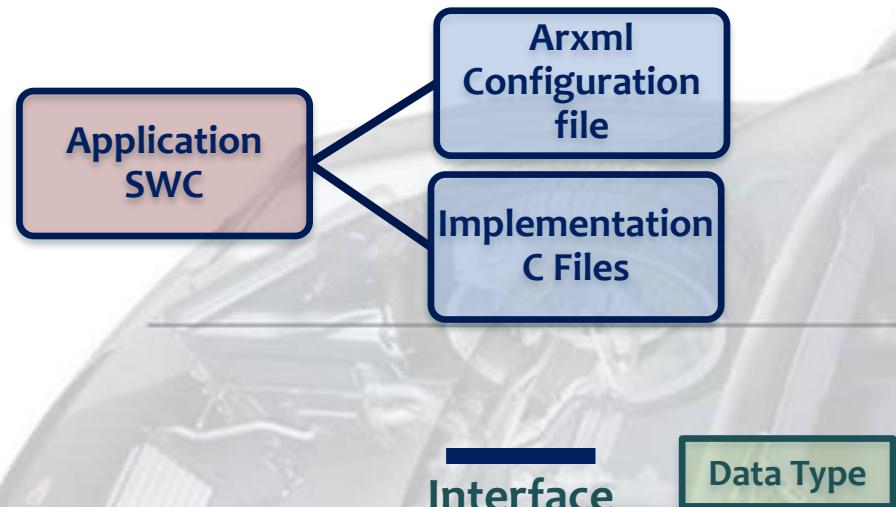
<https://www.learn-in-depth.com/>
<https://www.facebook.com/groups/embedded.system.KS/>



50

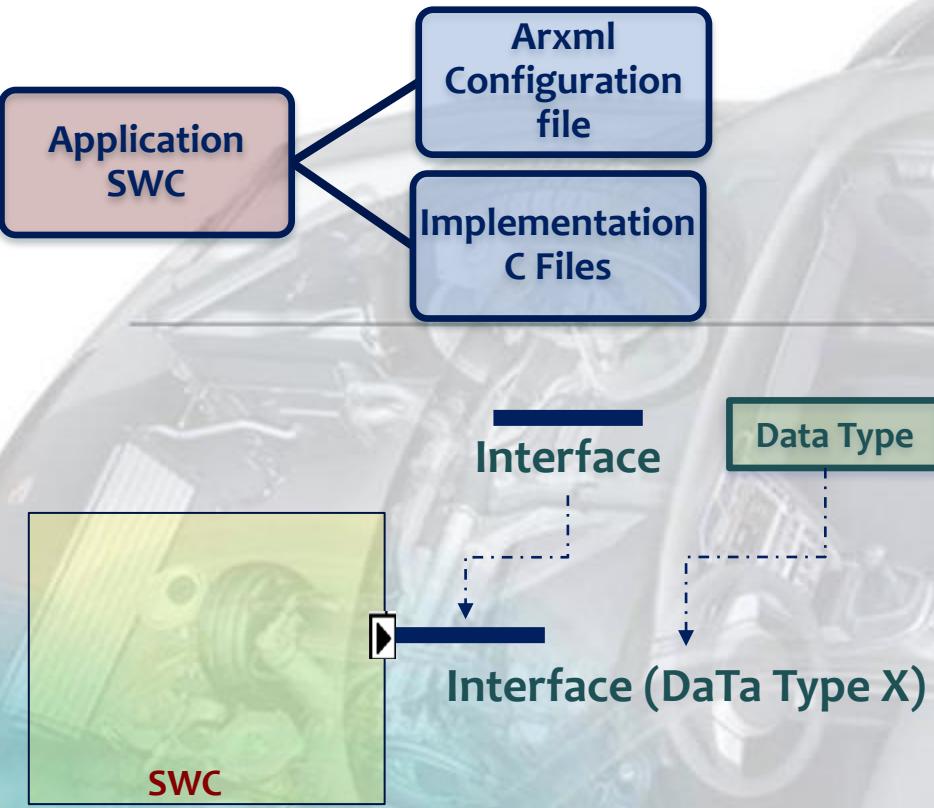
#LEARN_IN_DEPTH
#Be_professional_in
embedded_system

eng. Keroles Shenouda

<https://www.facebook.com/groups/embedded.system.KS/>

Data Types	Application Data Types
	Base Types
	Implementation Data Types
	...
Interfaces	Sender-Receiver
	Client-Server
	Calibration

<https://www.learn-in-depth.com/>
<https://www.facebook.com/groups/embedded.system.KS/>



Data Types	Application Data Types
	Base Types
	Implementation Data Types
	...
Interfaces	Sender-Receiver
	Client-Server
	Calibration

Software Component Types	Ports
	Communication Specifications

<https://www.learn-in-depth.com/>
<https://www.facebook.com/groups/embedded.system.KS/>



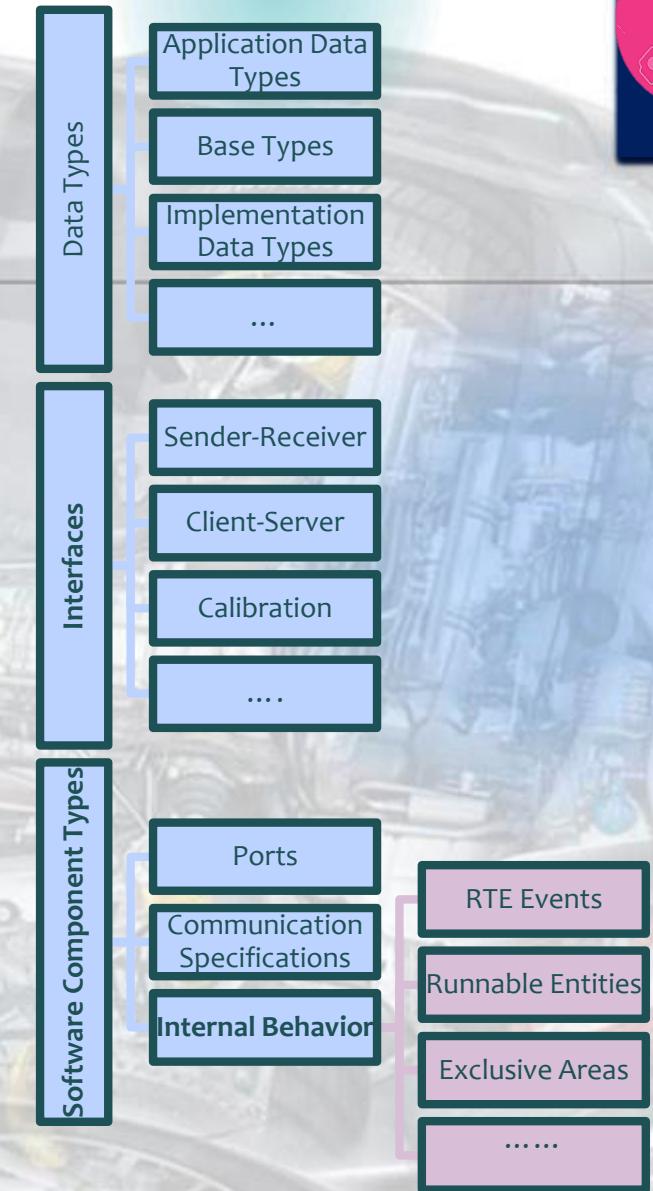
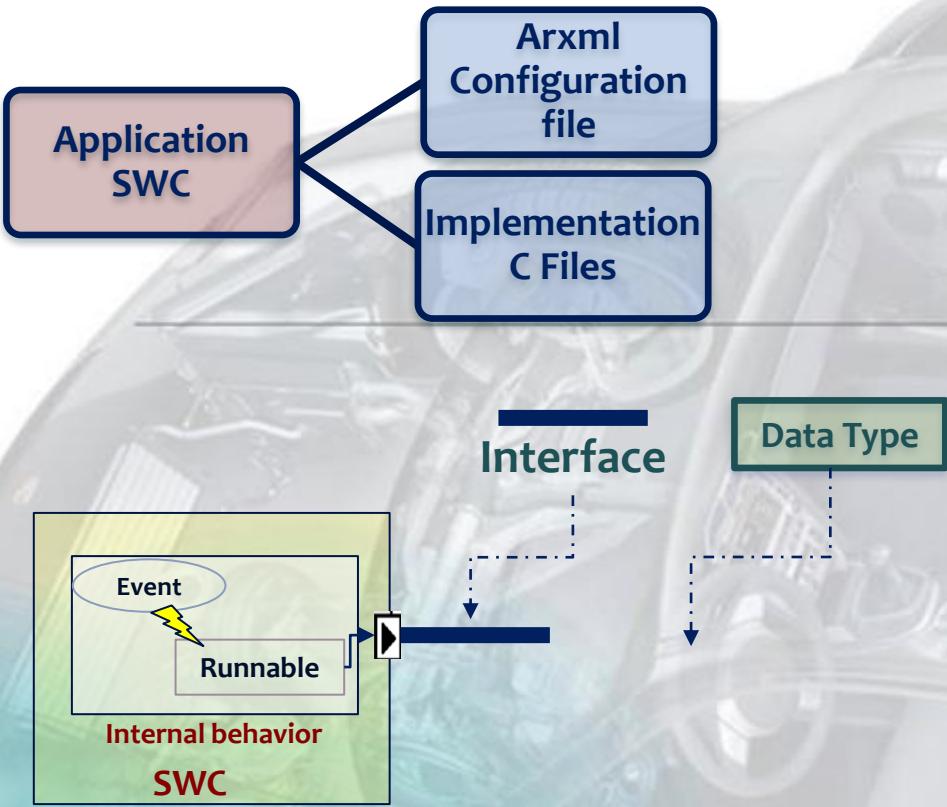
52

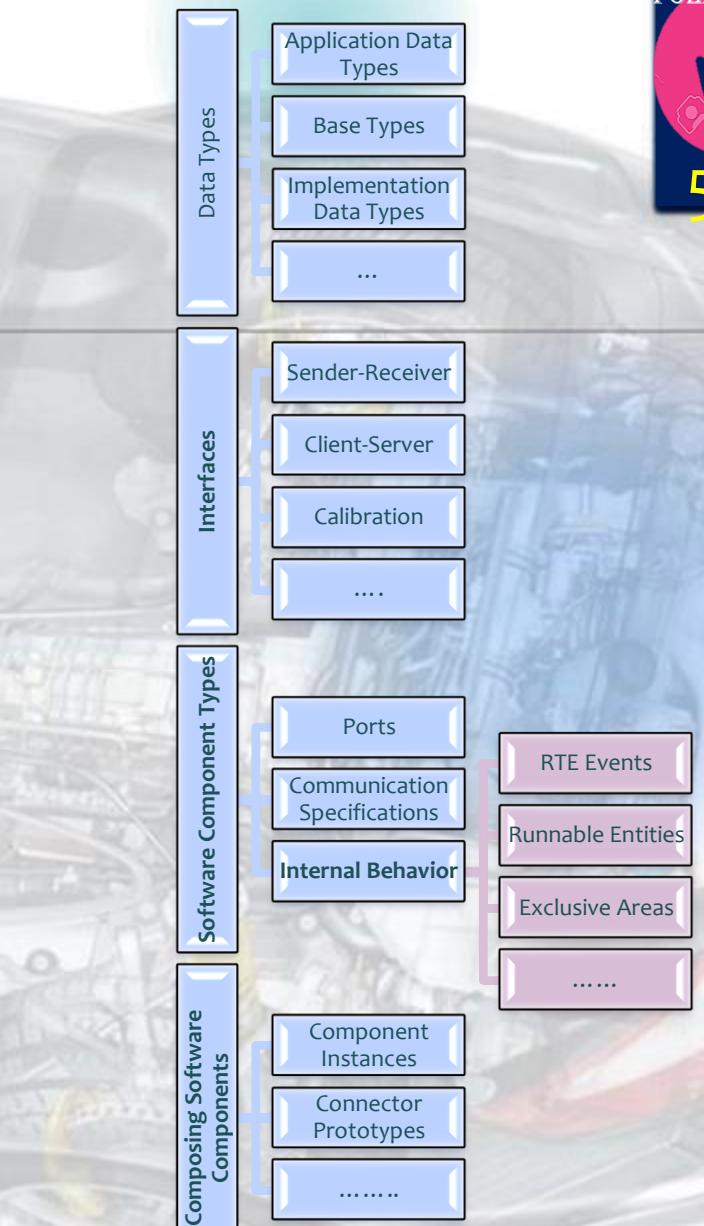
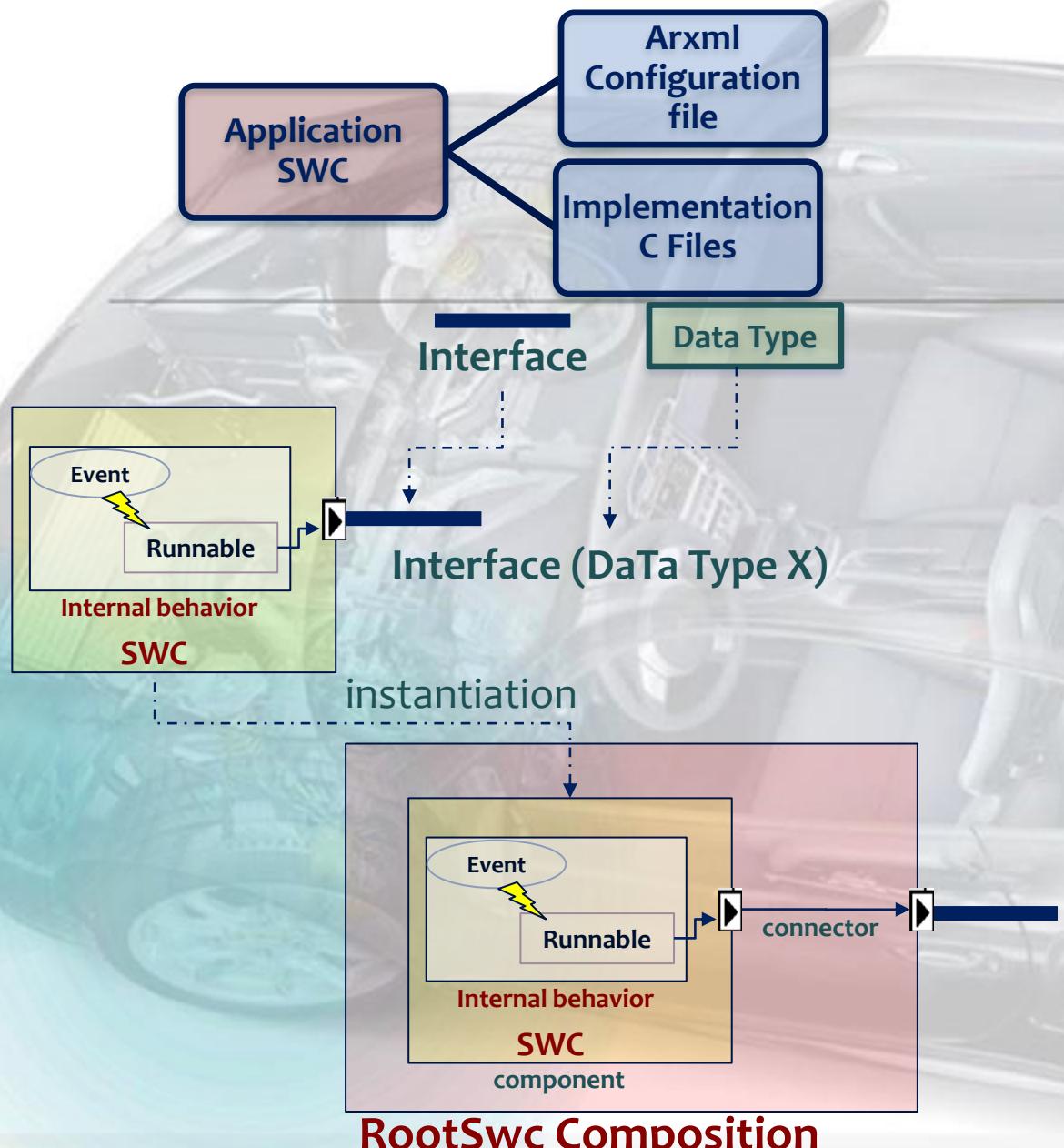
#LEARN_IN_DEPTH
#Be_professional_in_embedded_system

Eng. Keroles Shenouda

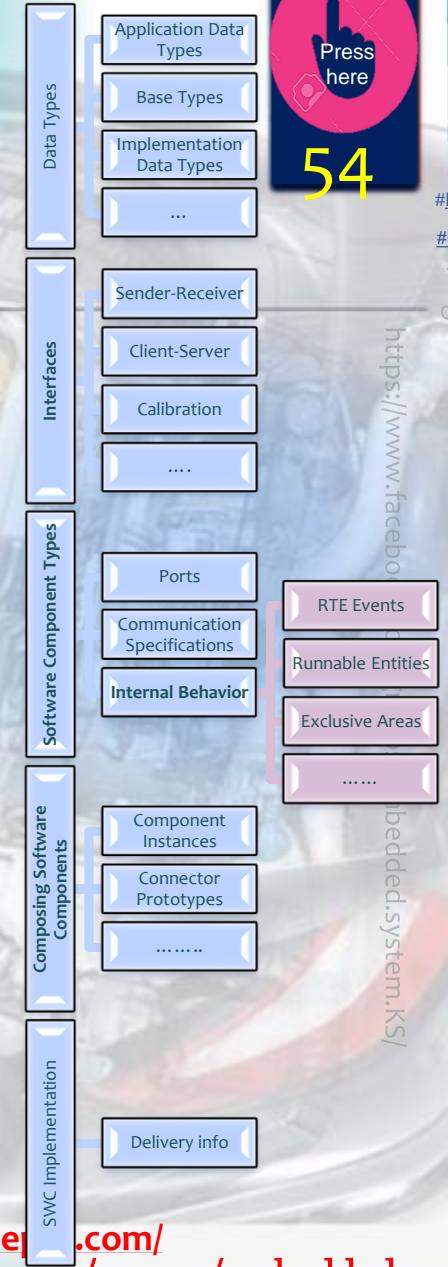
<https://www.facebook.com/groups/embedded.system.KS/>

<https://www.learn-in-depth.com/>
<https://www.facebook.com/groups/embedded.system.KS/>





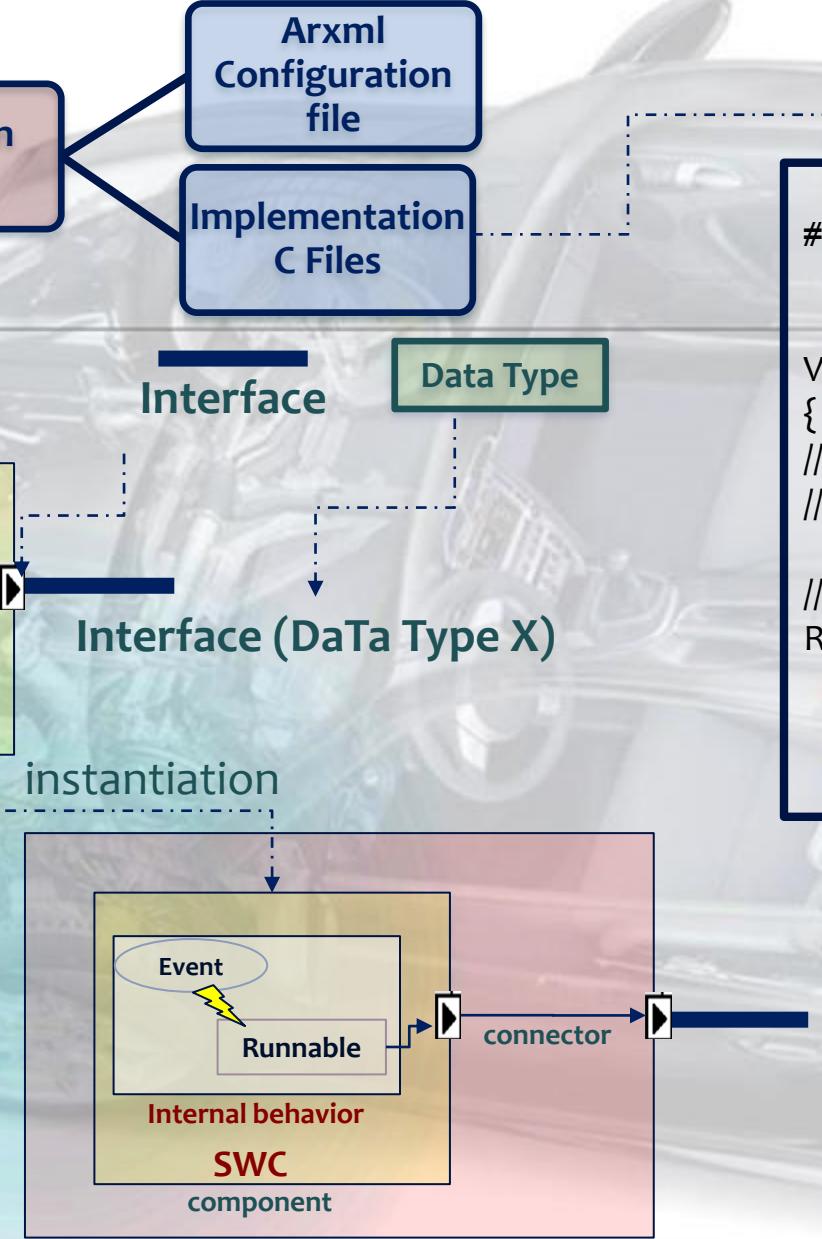
<https://www.learn-in-depth.com/>
<https://www.facebook.com/groups/embedded.system.KS/>



#include "Rte_<SWCName>.h"

```
Void runnable1 ()  
{  
//declarations  
//calculations
```

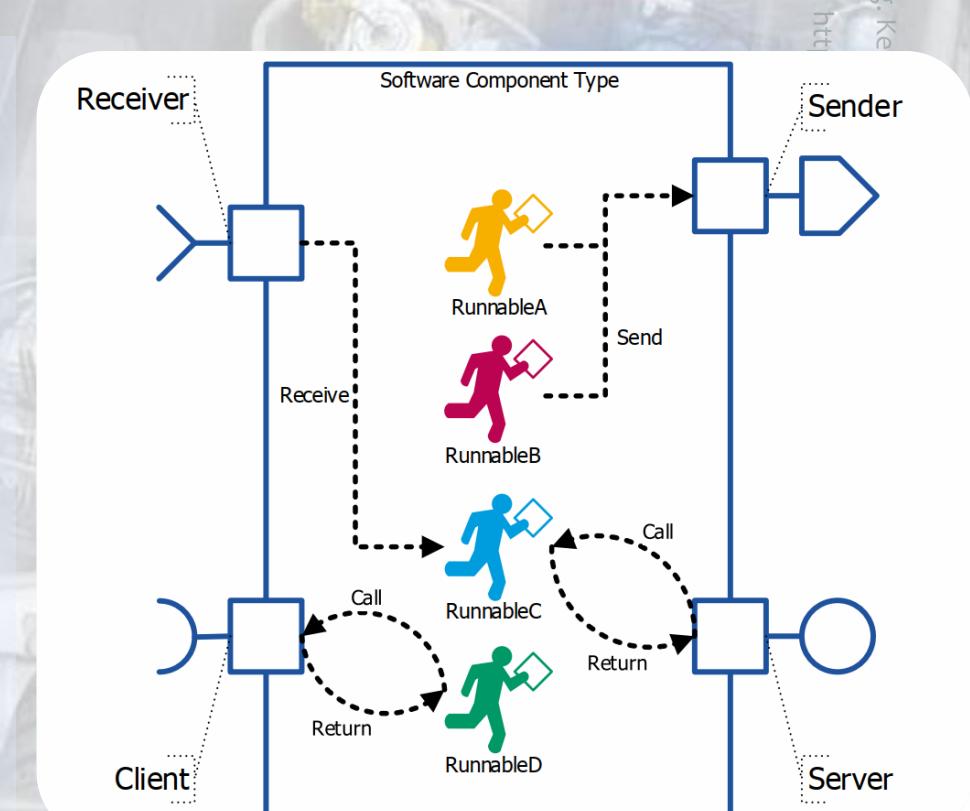
```
//Send calculated data  
Rte_Write_<Pport>_<DataName>(value);
```



<https://www.learn-in-depth.com/>
<https://www.facebook.com/groups/embedded.system.KS/>

Internal Behaviour

- ▶ The Internal Behaviour is the most complex part of the Software Component Description
- ▶ Each Atomic Software Component is advised to have only one Internal Behaviour
- ▶ the **Internal Behavior** consists of the following main elements
 - ▶ Runnable Entities
 - ▶ RTE Events
 - ▶ Exclusive Areas
 - ▶ RunnableExecution Constraints
 - ▶ PerInstanceMemory (PIM)



<https://www.learn-in-depth.com/>
<https://www.facebook.com/groups/embedded.system.KS/>



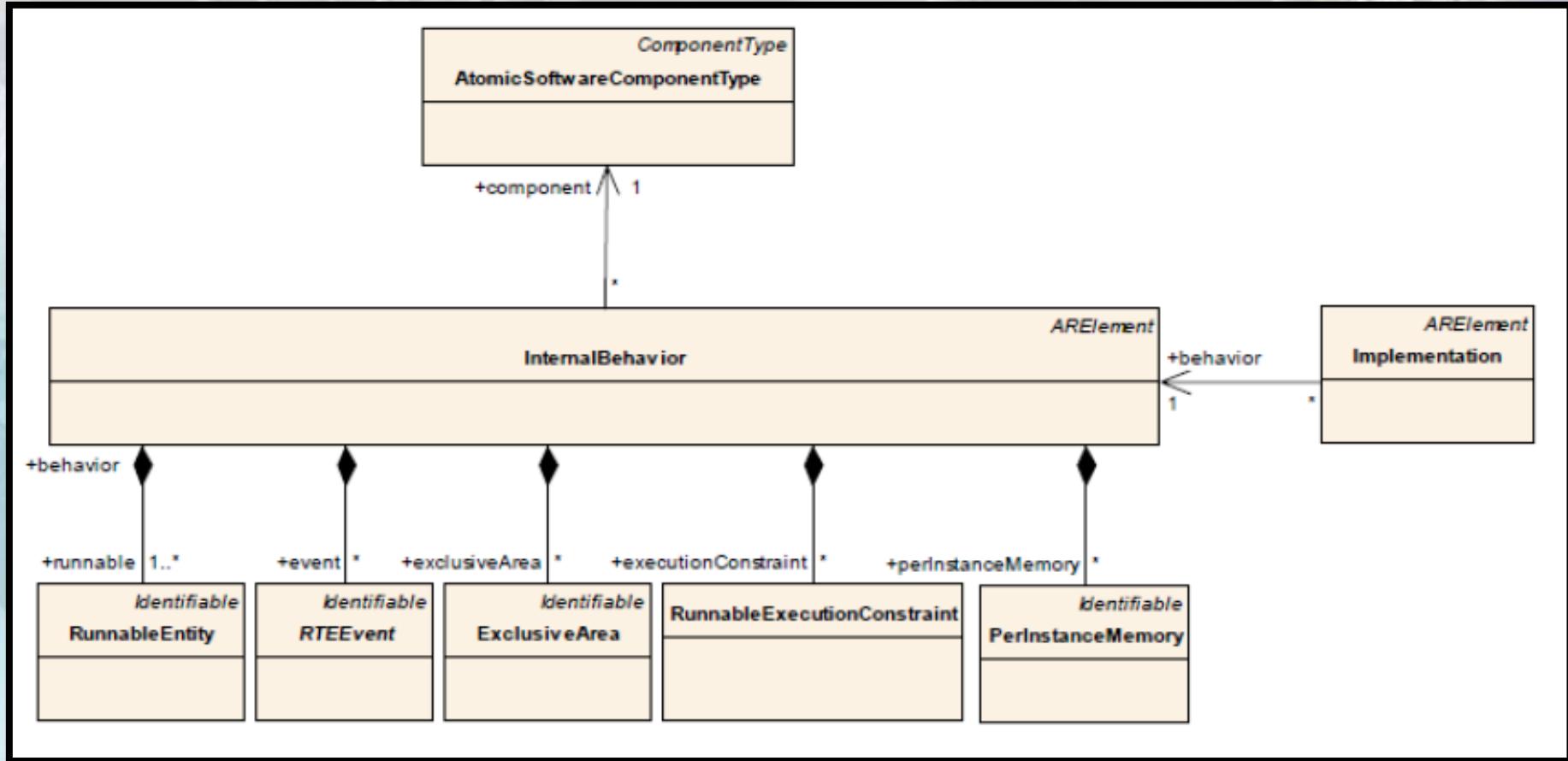
56

eng. Keroles Shenouda

<https://www.facebook.com/groups/embedded.system.KS/>

#Be_professional_in
embedded_system

Meta-model of the Internal Behavior



<https://www.learn-in-depth.com/>
<https://www.facebook.com/groups/embedded.system.KS/>

AUTOSAR

Software Component Template
AUTOSAR CP Release 4.4.0

7 Internal Behavior

7.1 Introduction

[TPS_SWCT_01075] **SwInternalBehavior** [**SwInternalBehavior** provides means for formally defining the behavior of an **AtomicSwComponentType**.] (RS_SWCT_03040)

This chapter focuses on the description of the **SwInternalBehavior** meta-class and the various meta-classes it aggregates. An overview of the meta-class is sketched in Figure 7.2. Please note that **SwInternalBehavior** inherits from **InternalBehavior**.

The role of **SwInternalBehavior** in the context of an AUTOSAR software-component is depicted in Figure 7.1. As mentioned in section 3.2, the reason to make the aggregation of **SwInternalBehavior** to **AtomicSwComponentType** «**atpSplittable**» is to allow for the development of **SwInternalBehavior** in a later process step (e.g. after the VFB view has been completed).

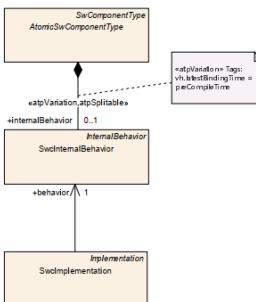


Figure 7.1: The “big picture” of **SwInternalBehavior**

Class	InternalBehavior (abstract)
Package	M2::AUTOSARTemplates::CommonStructure::InternalBehavior
Note	Common base class (abstract) for the internal behavior of both software components and basic software modules/clusters.
Base	APObject, AtpClassifier, AtpFeature, AtpStructureElement, Identifiable, MultilanguageReferable, Referable
Subclasses	BswInternalBehavior, SwInternalBehavior

AUTOSAR

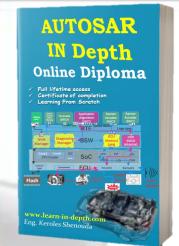
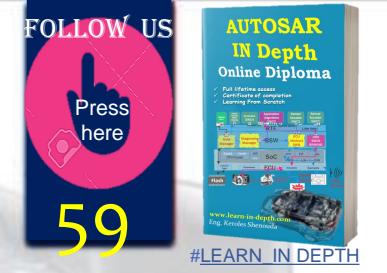
Software Component Template
AUTOSAR CP Release 4.4.0

△

Class	InternalBehavior (abstract)	Type	Mul.	Kind	Note
constantMemory	ParameterDataPrototype	*	aggr		Describes a read only memory object containing characteristic value(s) implemented by this Internal Behavior. The shortName of ParameterDataPrototype has to be equal to the ‘C’ identifier of the described constant. The characteristic value(s) might be shared between SwComponentPrototypes of the same SwComponent Type. The aggregation of constantMemory is subject to variability with the purpose to support variability in the software component or module implementations. Typically different algorithms in the implementation are requiring different number of memory objects. Stereotypes: atpSplittable; atpVariation Tags: atp.Splittkey=shortName, variationPoint.shortLabel vh.latestBindingTime=preCompileTime
constantValueMapping	ConstantSpecificationMappingSet	*	ref		Reference to the ConstantSpecificationMapping to be applied for the particular InternalBehavior Stereotypes: atpSplittable Tags: atp.Splittkey=constantValueMapping
dataTypeMapping	DataTypeMappingSet	*	ref		Reference to the DataTypeMapping to be applied for the particular InternalBehavior Stereotypes: atpSplittable Tags: atp.Splittkey=dataTypeMapping
exclusiveArea	ExclusiveArea	*	aggr		This specifies an ExclusiveArea for this InternalBehavior. The exclusiveArea is local to the component resp. module. The aggregation of ExclusiveAreas is subject to variability. Note: the number of ExclusiveAreas might vary due to the conditional existence of RunnableEntities or BswModule Entities. Stereotypes: atpSplittable; atpVariation Tags: atp.Splittkey=shortName, variationPoint.shortLabel vh.latestBindingTime=preCompileTime
exclusiveAreaNestingOrder	ExclusiveAreaNestingOrder	*	aggr		This represents the set of ExclusiveAreaNestingOrder owned by the InternalBehavior. Stereotypes: atpSplittable; atpVariation Tags: atp.Splittkey=shortName, variationPoint.shortLabel vh.latestBindingTime=preCompileTime
staticMemory	VariableDataPrototype	*	aggr		Describes a read and writeable static memory object representing measurement variables implemented by this software component. The term “static” is used in the meaning of “non-temporary” and does not necessarily specify a linker encapsulation. This kind of memory is only supported if supportsMultipleInstantiation is FALSE. The shortName of the VariableDataPrototype has to be equal with the ‘C’ identifier of the described variable. The aggregation of staticMemory is subject to variability with the purpose to support variability in the software component’s implementations. ▽

▽

<https://www.learn-in-depth.com/>
<https://www.facebook.com/groups/embedded.system.KS/>



www.learn-in-depth.com

Eng. Keroles Shenouda

#LEARN_IN_DEPTH

#Be_professional_in

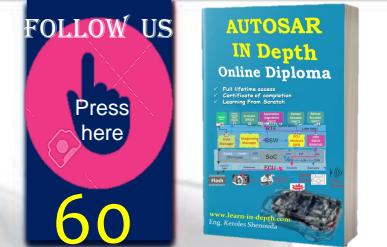
embedded_system

eng. Keroles Shenouda

<https://www.facebook.com/groups/embedded.system.KS/>

Source: www.autosar.org

<https://www.learn-in-depth.com/>
<https://www.facebook.com/groups/embedded.system.KS/>



60

#LEARN_IN_DEPTH
#Be_professional_in_embedded_system

eng. Keroles Shenouda

<https://www.facebook.com/groups/embedded.system.KS/>

References



- ▶ <https://www.autosar.org>
- ▶ **Embedded Microcomputer Systems Real Time Interfacing Third Edition**
Jonathan W. Valvano University of Texas at Austin.
- ▶ **MicroC/OS-II the real-time kernel second edition jean j.labrosse.**
- ▶ **RTOS Concepts** <http://www.embeddedcraft.org>.
- ▶ **OSEK/VDX Operating System Specification 2.2.3**
- ▶ **AUTOSAR Layered Software Architecture**
- ▶ **The Trampoline Handbook release 2.0**
- ▶ **Trampoline (OSEK/VDX OS) Test Implementation -Version 1.0, Florent PAVIN ; Jean-Luc BECHENNEC**
- ▶ **Autosar Architecture (Learn from Scratch with Demo) Udemy Course**

<https://www.learn-in-depth.com/>
<https://www.facebook.com/groups/embedded.system.KS/>



eng. Keroles Shenouda

#LEARN_IN_DEPTH

#Be_professional_in
embedded_system

References

- ▶ Trampoline:an open platform for (small) embedded systems based on OSEK/VDX and AUTOSAR

<http://trampoline.rts-software.org/>

Jean-Luc Béchennec^{1;2}, Sébastien Faucou^{1;3}

¹IRCCyN (Institute of Research in Communications and Cybernetics of Nantes)

²CNRS (National Center for Scientific Research) / ³University of Nantes

10th Libre Software Meeting

<https://www.learn-in-depth.com/>
<https://www.facebook.com/groups/embedded.system.KS/>



References

► Real Time Systems (RETSY)

Jean-Luc Béchennec - Jean-Luc.Bechennec@ircbyn.ec-nantes.fr

Sébastien Faucou - Sebastien.Faucou@univ-nantes.fr

jeudi 12 novembre 15

► AUTOSAR Specification of Operating System V5.0.0 R4.0 Rev 3

► OSEK - Basics <http://taisnotes.blogspot.com.eg/2016/07/osek-basic-task-vs-extended-task.html>

► OSEK OS Session Speaker Deepak V.

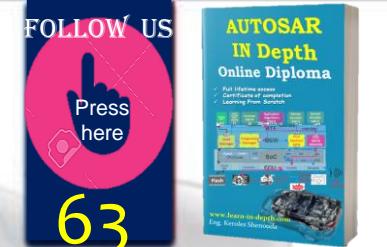
M.S Ramaiah School of Advanced Studies - Bangalore 1

► Introducción a OSEK-OS - El Sistema Operativo del CIAA-Firmware

Programación de Sistemas Embebidos

MSc. Ing. Mariano Cerdeiro

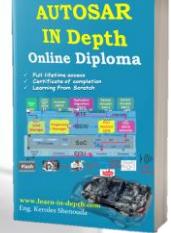
<https://www.learn-in-depth.com/>
<https://www.facebook.com/groups/embedded.system.KS/>



63

Eng. Keroles Shenouda

<https://www.facebook.com/groups/embedded.system.KS/>



#LEARN_IN_DEPTH
#Be_professional_in
embedded_system

References

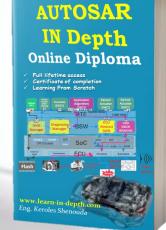
- ▶ Introduction to AUTOSAR, Stephen Waldron, **Vector webinar**
Wednesday 7th May 2014

https://vector.com/portal/medien/cmc/events/Webinars/2014/Vector_Webinar_AUTOSAR_Introduction_20140507_EN.pdf

- ▶ Introduction to AUTOSAR, Stephen Waldron, **Vector webinar**
Tuesday 5th May 2015

https://vector.com/portal/medien/cmc/events/Webinars/2015/Vector_Webinar_AUTOSAR_Introduction_20150505_EN.pdf

<https://www.learn-in-depth.com/>
<https://www.facebook.com/groups/embedded.system.KS/>



#LEARN_IN_DEPTH
#Be_professional_in
embedded_system

eng_Keroles Shenouda

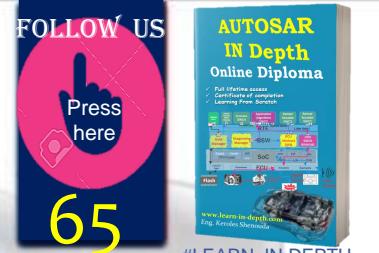
References

- ▶ Applying AUTOSAR in Practice Available Development Tools and Migration Paths Master Thesis, Computer Science Authors: Jesper Melin

<http://www.idt.mdh.se/utbildning/exjobb/files/TR1171.pdf>

Freescale [AUTOSAR Software Overview.pdf](#)

<https://www.learn-in-depth.com/>
<https://www.facebook.com/groups/embedded.system.KS/>



References

- AUTOSAR Method, Vector Webinar 2013-04-17

https://vector.com/portal/medien/cmc/events/Webinars/2013/Vector_Webinar_AUTOSAR_Method_20130417.pdf

- AUTOSAR Configuration Process - How to handle 1000s of parameters
Vector Webinar 2013-04-19

https://vector.com/portal/medien/cmc/events/Webinars/2013/Vector_Webinar_AUTOSAR_Configuration_Process_20130419_EN.pdf

- AUTOSAR Runtime Environment and Virtual Function Bus, Nico Naumann

https://hpi.de/fileadmin/user_upload/fachgebiete/giese/Ausarbeitungen_AUTOSAR0809/NicoNaumann_RTE_VFB.pdf

<https://www.learn-in-depth.com/>
<https://www.facebook.com/groups/embedded.system.KS/>



66

#LEARN_IN_DEPTH
#Be_professional_in_embedded_system

eng. Keroles Shenouda

References

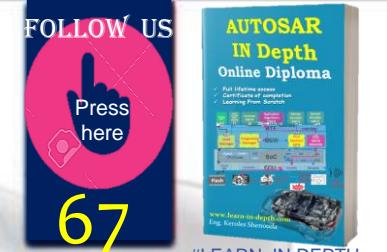
- ▶ **The AUTOSAR Adaptive Platform for Connected and Autonomous Vehicles**, Simon Fürst, AUTOSAR Steering Committee 8th Vector Congress 29-Nov-2016, Alte Stuttgarter Reithalle, Stuttgart, Germany

https://vector.com/congress/files/presentations/VeCo16_06_29Nov_Reithalle_Fuerst_BMW.pdf

- ▶ **A Review of Embedded Automotive Protocols**, Nicolas Navet¹, Françoise Simonot-Lion² April 14, 2008

https://www.realtimeatwork.com/wp-content/uploads/chapter4_CRC_2008.pdf

<https://www.learn-in-depth.com/>
<https://www.facebook.com/groups/embedded.system.KS/>



67

Eng. Keroles Shenouda

References

- AUTOSAR Adaptive Platform

https://vector.com/conference_india/files/presentations/Day1/3_AUTOSAR%20Adaptive%20Platform.pdf

- AUTOAR Specification of Diagnostic Communication Manager

https://www.autosar.org/fileadmin/user_upload/standards/classic/3-1/AUTOSAR_SWS_DCM.pdf

- Automotive & Embedded Info

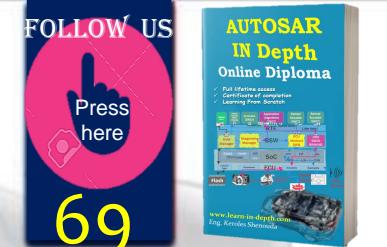
<https://automotiveembeddedsite.wordpress.com/memory-stack/>

<https://www.learn-in-depth.com/>
<https://www.facebook.com/groups/embedded.system.KS/>

References

- ▶ <http://www.autosar.org/about/technical-overview/ecu-software-architecture/autosar-basic-software/>
- ▶ <http://www.autosar.org/standards/classic-platform/>
- ▶ https://automotivetechis.files.wordpress.com/2012/05/communicationsstack_gosda.pdf
- ▶ https://automotivetechis.files.wordpress.com/2012/05/autosar_ppt.pdf
- ▶ <https://automotivetechis.wordpress.com/autosar-concepts/>
- ▶ https://automotivetechis.files.wordpress.com/2012/05/autosar_exp_layeredsoftwarearchitecture.pdf
- ▶ <http://www.slideshare.net/FarzadSadeghi1/autosar-software-component>
- ▶ <https://www.renesas.com/en-us/solutions/automotive/technology/autosar/autosar-mcal.html>
- ▶ https://github.com/parai/OpenSAR/blob/master/include/Std_Types.h

<https://www.learn-in-depth.com/>
<https://www.facebook.com/groups/embedded.system.KS/>



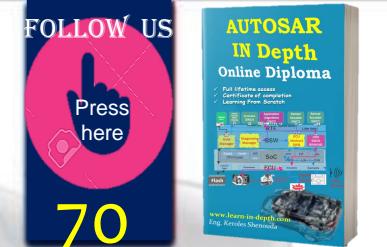
Embedded system_KS

#LEARN_IN_DEPTH
#Be_professional_in_embedded_system

References

- ▶ <https://www.softwaretestingmaterial.com/stlc-software-testing-life-cycle/>
- ▶ <https://www.amarinfotech.com/roles-responsibilities-agile-tester.html>
- ▶ <https://www.softwaretestinghelp.com/scrum-artifacts/>
- ▶ <https://www.visual-paradigm.com/scrum/what-are-scrum-ceremonies/>
- ▶ **ISO 26262 - Software Advanced Training (Part II)**
Dr. Julian Wolf TÜV SÜD Product Service GmbH
http://documents.irevues.inist.fr/bitstream/handle/2042/56194/lm19_com_4D-5_062_A_Mihalache.pdf?sequence=1
- ▶ **ISO 26262-6 part 6**
 - ▶ Second edition 2018-12
- ▶ Verification and Validation: A Quick Introduction
 - ▶ <https://slideplayer.com/slide/4564006/>

<https://www.learn-in-depth.com/>
<https://www.facebook.com/groups/embedded.system.KS/>



70

#LEARN_IN_DEPTH
#Be_professional_in_embedded_system

Embedded_system_KS

References

- ▶ <http://www.starkinfotech.com/manual-testing/>
- ▶ <https://www.embitel.com/blog/embedded-blog/understanding-how-iso-26262-asil-is-determined-for-automotive-applications>
- ▶ Increasing Efficiency of ISO 26262 Verification and Validation by Combining Fault Injection and Mutation Testing with Model based Development
 - ▶ <https://www.scitepress.org/papers/2013/45920/45920.pdf>
- ▶ <https://www.techdesignforums.com/practice/technique/formal-fault-analysis-for-iso26262-find-faults-before-they-find-you/>

<https://www.learn-in-depth.com/>
<https://www.facebook.com/groups/embedded.system.KS/>

References

- ▶ <https://docs.google.com/viewer?a=v&pid=sites&srcid=ZmtlLnV0bS5teXxyaWR6dWFuLXMtd2Vic2I0ZXxneDo2ODU0NzIKM2JkOTg4MjRk>
- ▶ <http://www.avrprojects.net/index.php/avr-projects/sensors/38-humidity-and-temperature-sensor-dht11?showall=&start=1>
- ▶ <http://www.cse.wustl.edu/~lu/cse467s/slides/dsp.pdf>
- ▶ <http://www.avr-tutorials.com/>
- ▶ Microprocessor: ATmega32 (SEE3223-10)
<http://ridzuan.fke.utm.my/microprocessor-atmega32-see3223-10>
- ▶ <http://circuitdigest.com/article/what-is-the-difference-between-microprocessor-and-microcontroller>
- ▶ http://cs4hs.cs.pub.ro/wiki/roboticsisfun/chapter2/ch2_7_programming_a_microcontroller
- ▶ Embedded Systems with ARM Cortex-M Microcontrollers in Assembly Language and C Dr. Yifeng Zhu Third edition June 2018

<https://www.learn-in-depth.com/>
<https://www.facebook.com/groups/embedded.system.KS/>

References

- ▶ <http://techdifferences.com/difference-between-interrupt-and-polling-in-os.html>
- ▶ http://www.bogotobogo.com/Embedded/hardware_interrupt_software_interrupt_latency_irq_vs_fiq.php
- ▶ Preventing Interrupt Overload Presented by Jiyong Park Seoul National University, Korea 2005. 2. 22. John Regehr, Usit Duogsaa, School of Computing, University.
- ▶ First Steps Embedded Systems Byte Craft Limited reference
- ▶ COMPUTER ORGANIZATION AND ARCHITECTURE DESIGNING FOR PERFORMANCE EIGHTH EDITION William Stallings
- ▶ Getting Started with the Tiva™ TM4C123G LaunchPad Workshop

<https://www.learn-in-depth.com/>
<https://www.facebook.com/groups/embedded.system.KS/>

References

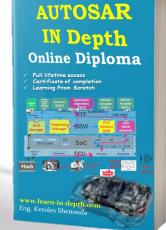
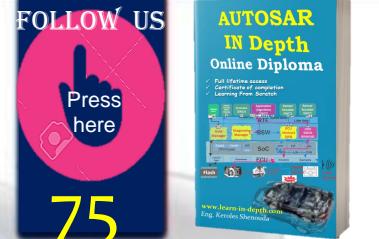
- ▶ Tiva™ TM4C123GH6PM Microcontroller DATA SHEET
- ▶ Interrupts and Exceptions COMS W6998 Spring 2010
- ▶ THE AVR MICROCONTROLLER. AND EMBEDDED SYSTEMS Using Assembly and C. Muhammad Ali Mazidi.
- ▶ <http://embedded-lab.com/blog/tinkering-ti-msp430f5529/27/>
- ▶ [How to use JIRA \(AR\)](#)

<https://www.learn-in-depth.com/>
<https://www.facebook.com/groups/embedded.system.KS/>

References

- ▶ <https://docs.google.com/viewer?a=v&pid=sites&srcid=ZmtILnV0bS5teXxyaWR6dWFuLXMfd2Vic2I0ZXxneDo2ODU0Nzlkm2JkOTg4MjRk>
- ▶ <http://www.avrprojects.net/index.php/avr-projects/sensors/38-humidity-and-temperature-sensor-dht11?showall=&start=1>
- ▶ <http://www.cse.wustl.edu/~lu/cse467s/slides/dsp.pdf>
- ▶ <http://www.avr-tutorials.com/>
- ▶ Microprocessor: ATmega32 (SEE3223-10)
<http://ridzuan.fke.utm.my/microprocessor-atmega32-see3223-10>
- ▶ <http://circuitdigest.com/article/what-is-the-difference-between-microprocessor-and-microcontroller>
- ▶ AVR Microcontroller and Embedded Systems: Using Assembly and C (Pearson Custom Electronics Technology) 1st Edition
<https://www.amazon.com/AVR-Microcontroller-Embedded-Systems-Electronics/dp/0138003319>

<https://www.learn-in-depth.com/>
<https://www.facebook.com/groups/embedded.system.KS/>



#Be_professional_in
embedded_system

eng_Keroles Shenouda

References

- ▶ <https://www.newbiehack.com/MicrocontrollersABeginnersGuideIntroductionandInterfacinganLCD.aspx>
- ▶ <http://www.slideshare.net/MathivananNatarajan/asynchronous-serial-data-communication-and-standards>
- ▶ <https://www.slideshare.net/AnkitSingh13/uart-32550652>
- ▶ [the avr microcontroller and embedded. System using assembly and c. Muhammad Ali Mazidi](#)
- ▶ [Embedded Systems lectures "Engr. Rashid Farid Chishti"](#)
- ▶ https://www.corelis.com/education/SPI_Tutorial.htm
- ▶ <http://ftm.futureelectronics.com/2014/09/nxp-macronics-nor-series-quad-spi-flash-a-simpler-faster-alternative-to-standard-spi-flash-when-adding-external-memory-to-32-bit-mcu-systems/>
- ▶ <http://www.byteparadigm.com/products/spi-storm/spi-storm-advanced-information/>
- ▶ <https://stackoverflow.com/questions/17125505/what-makes-a-better-constant-in-c-a-macro-or-an-enum>
- ▶ <https://blog.digilentinc.com/i2c-how-does-it-work/#prettyPhoto>
- ▶ <https://raphoenixmakerevolution.files.wordpress.com/2015/09/spi-and-can-bus.pdf>
- ▶ http://denethor.wlu.ca/cp316/lectures/Serial_Interconnect_Bus.pdf
- ▶ <https://aticleworld.com/i2c-interview-questions/>
- ▶ <https://www.youtube.com/watch?v=7MZ-a-unAU8>

<https://www.learn-in-depth.com/>
<https://www.facebook.com/groups/embedded.system.KS/>



<https://www.learn-in-depth.com/>
<https://www.facebook.com/groups/embedded.system.KS/>