

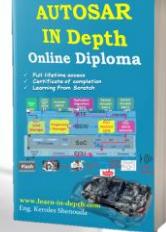
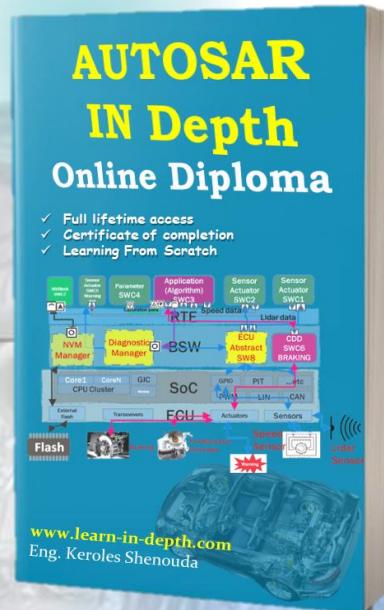
AUTOSAR In Depth

Online Diploma

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

Unit(AUTOSAR Methodology). Session(Configuration Parameters)

- ▶ Configuration Classes in AUTOSAR
 - ▶ Pre-compile time
 - ▶ Pre-Compile Examples
 - ▶ Link time
 - ▶ Link time Examples
 - ▶ Post-build time
 - ▶ Post-Build Loadable
 - ▶ Post-Build Selectable
 - ▶ Examples
- ▶ Specification of C Implementation Rules
 - ▶ Pre-compile time
 - ▶ Link time
- ▶ AUTOSAR Module Folder Structure



#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/>



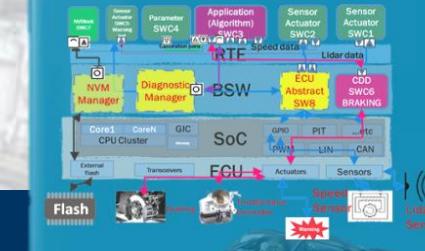
#LEARN_IN_DEPTH
#Be_professional_in
embedded_system

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

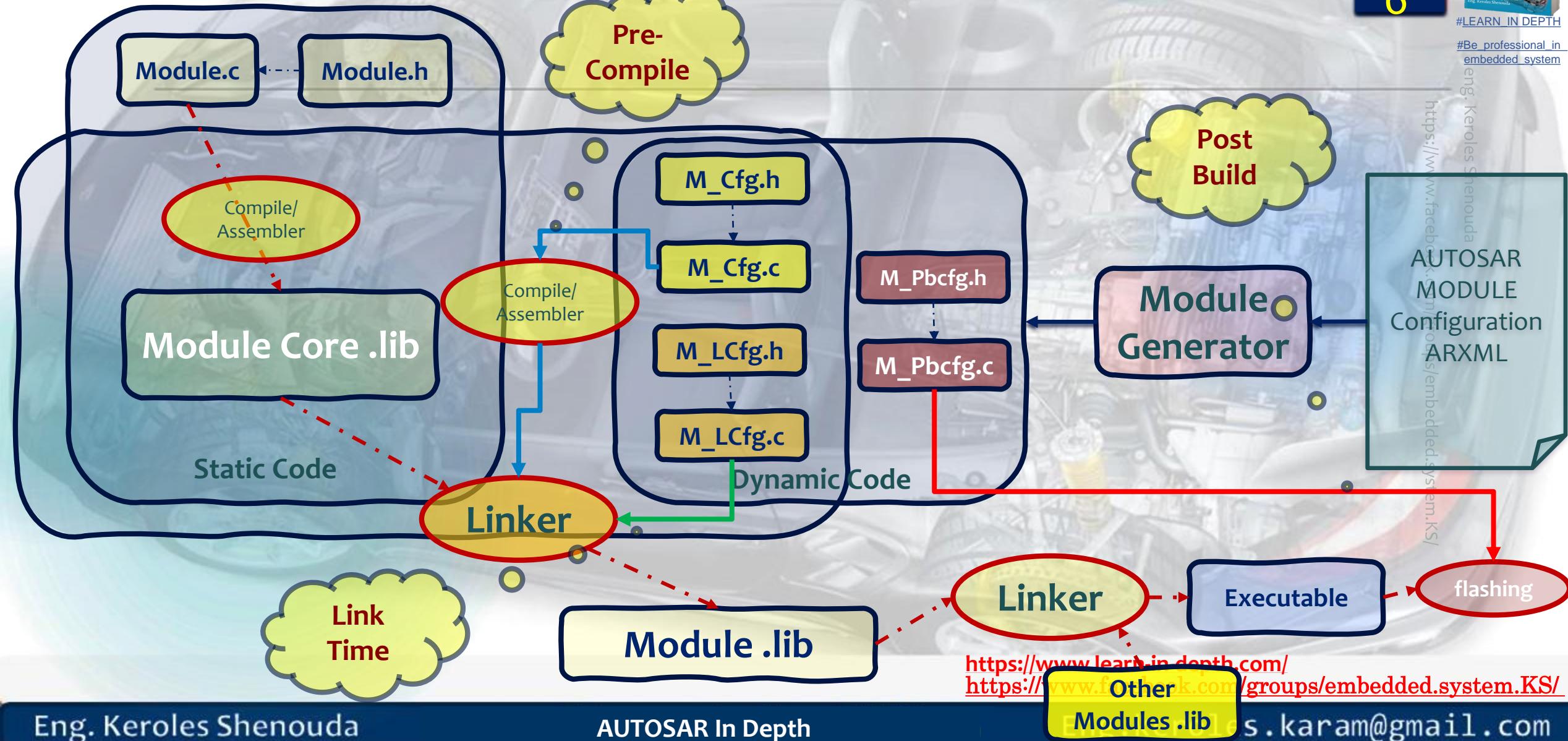
LEARN-IN-DEPTH
Be professional in
embedded system

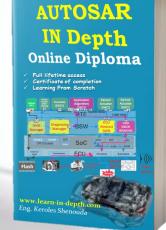
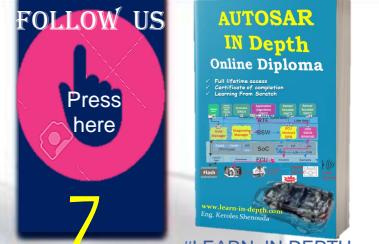


<https://www.learn-in-depth.com/>
<https://www.facebook.com/groups/embedded.system.KS/>

Configuration Classes in AUTOSAR

AUTOSAR Module Structure





#LEARN_IN_DEPTH
#Be_professional_in
embedded_system

eng. Keroles Shenouda

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

AUTOSAR Configuration Parameters Concept

- ▶ Based on the time when the configuration of parameters should occur,
- ▶ AUTOSAR defines three configuration classes: **Pre-compile time**, **link time** and **post-build time**.
- ▶ Each BSW module supports **one to three** configuration variants: **Pre-compile**, **link time** and **post-build**.
(chapter 10 in each SWS).

- 1 Introduction and functional overview
- 2 Acronyms and Abbreviations
- 3 Related documentation
- 4 Constraints and assumptions
- 5 Dependencies to other modules
- 6 Requirements Tracing
- 7 Functional specification
- 8 API specification
- 9 Sequence diagrams
- 10 Configuration specification
 - 10.1 Containers and configuration parameters
 - 10.1.1 CanIf
 - 10.1.2 CanIfPrivateCfg
 - 10.1.3 CanIfPublicCfg
 - 10.1.4 CanIfInitCfg
 - 10.1.5 CanIfTxPduCfg
 - 10.1.6 CanIfRxPduCfg
 - 10.1.7 CanIfRxPduCanIdRange

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

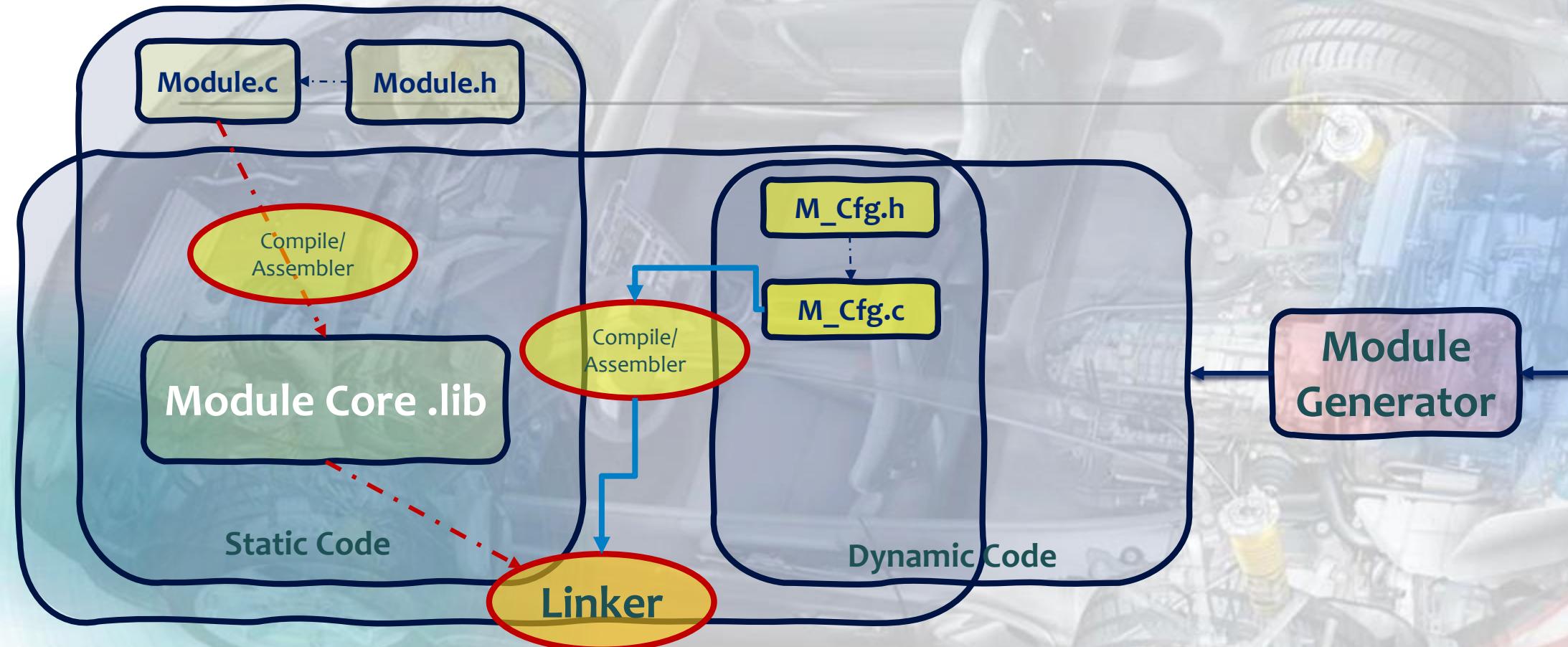
X: Support
-: Indicates not supported

SWS Item	ECUC_Com_00709 :		
Name	ComIPduCancellationSupport		
Parent Container	ComIPdu		
Description	Defines for I-PDUs with ComIPduType NORMAL: If the underlying IF-modul supports cancellation of transmit requests. Defines for I-PDUs with ComIPduType TP: If the underlying TP-module supports RX and TX cancellation of ongoing requests.		
Multiplicity	0..1		
Type	EcucBooleanParamDef		
Default value	false		
Post-Build Variant Multiplicity	true		
Post-Build Variant Value	true		
Multiplicity Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPIL
	Link time	--	
	Post-build time	X	VARIANT-POST-BUILD
Value Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPIL
	Link time	--	
	Post-build time	X	VARIANT-POST-BUILD
Scope / Dependency	scope: ECU dependency: This parameter shall not be set to true if ComCancellation-Support is set to false		

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



AUTOSAR Module Structure

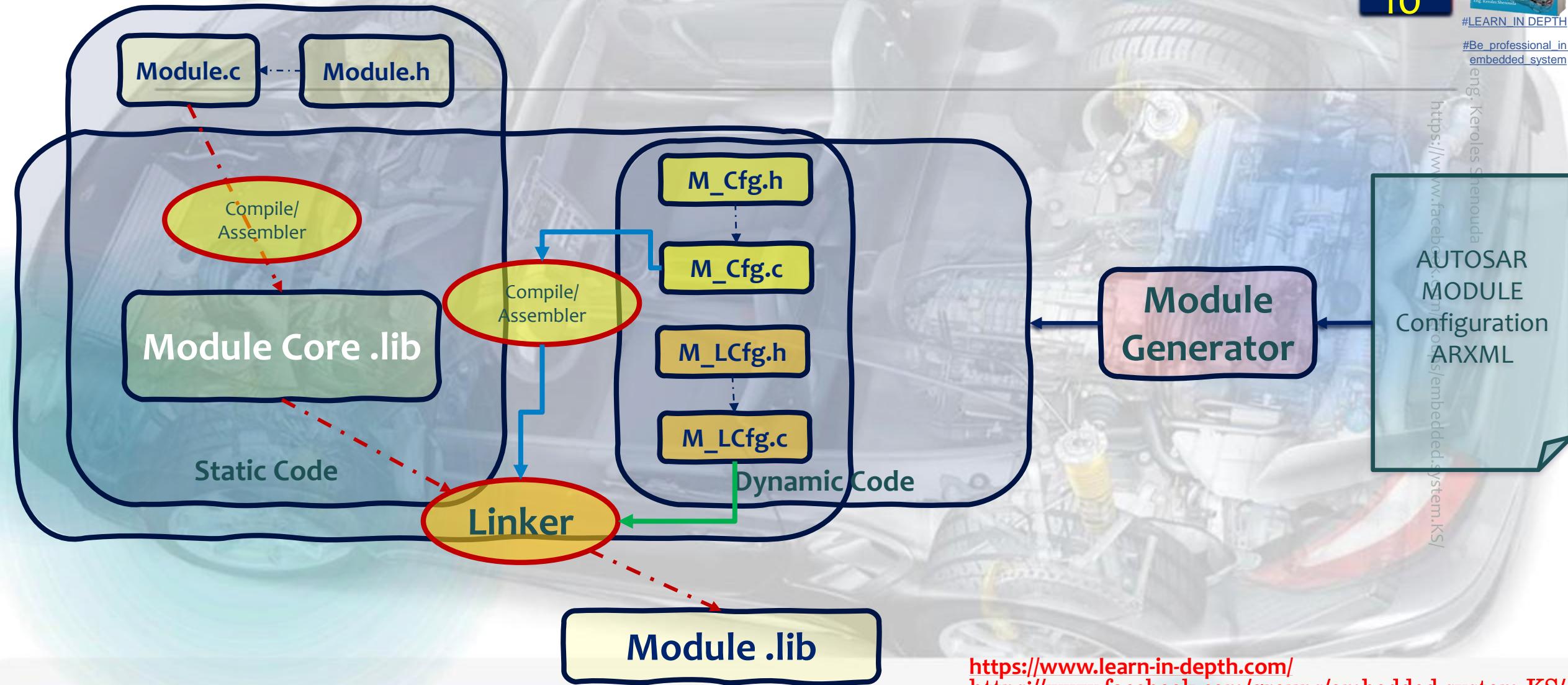


eng. Keroles Shenouda
<https://www.facebook.com/groups/embedded.system.KS/>
AUTOSAR MODULE Configuration ARXML

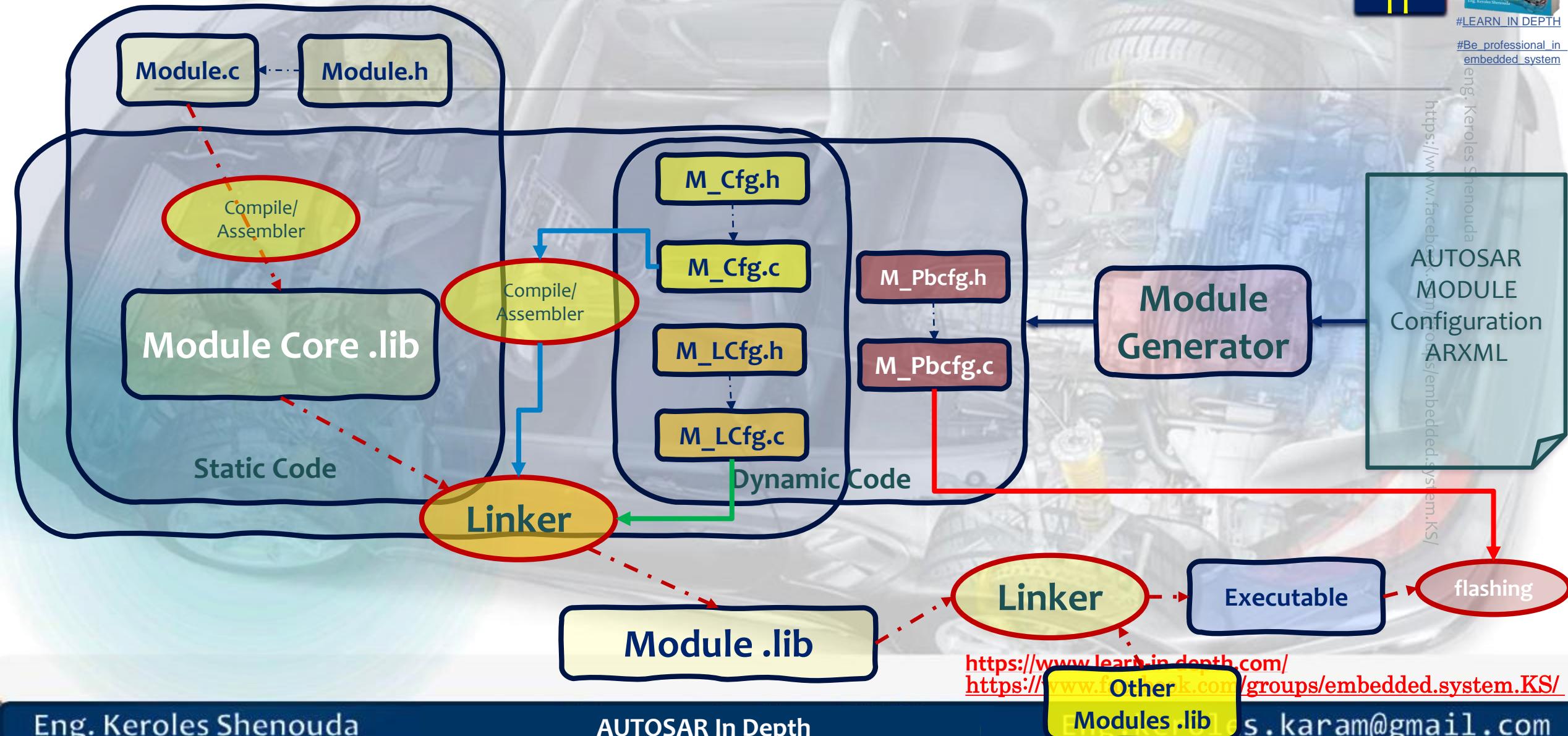
<https://www.learn-in-depth.com/>
<https://www.facebook.com/groups/embedded.system.KS/>



AUTOSAR Module Structure

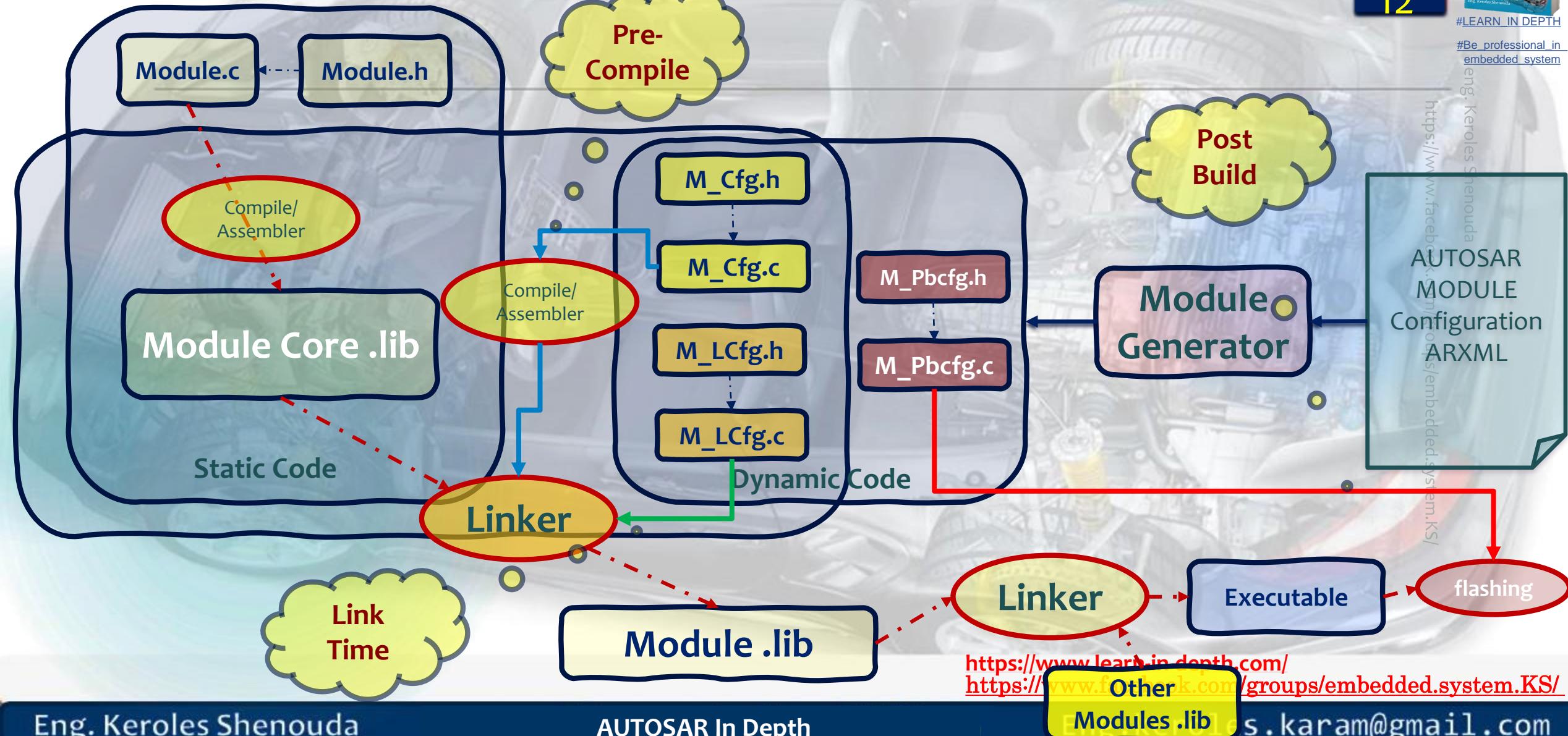


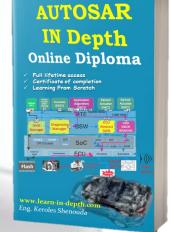
AUTOSAR Module Structure





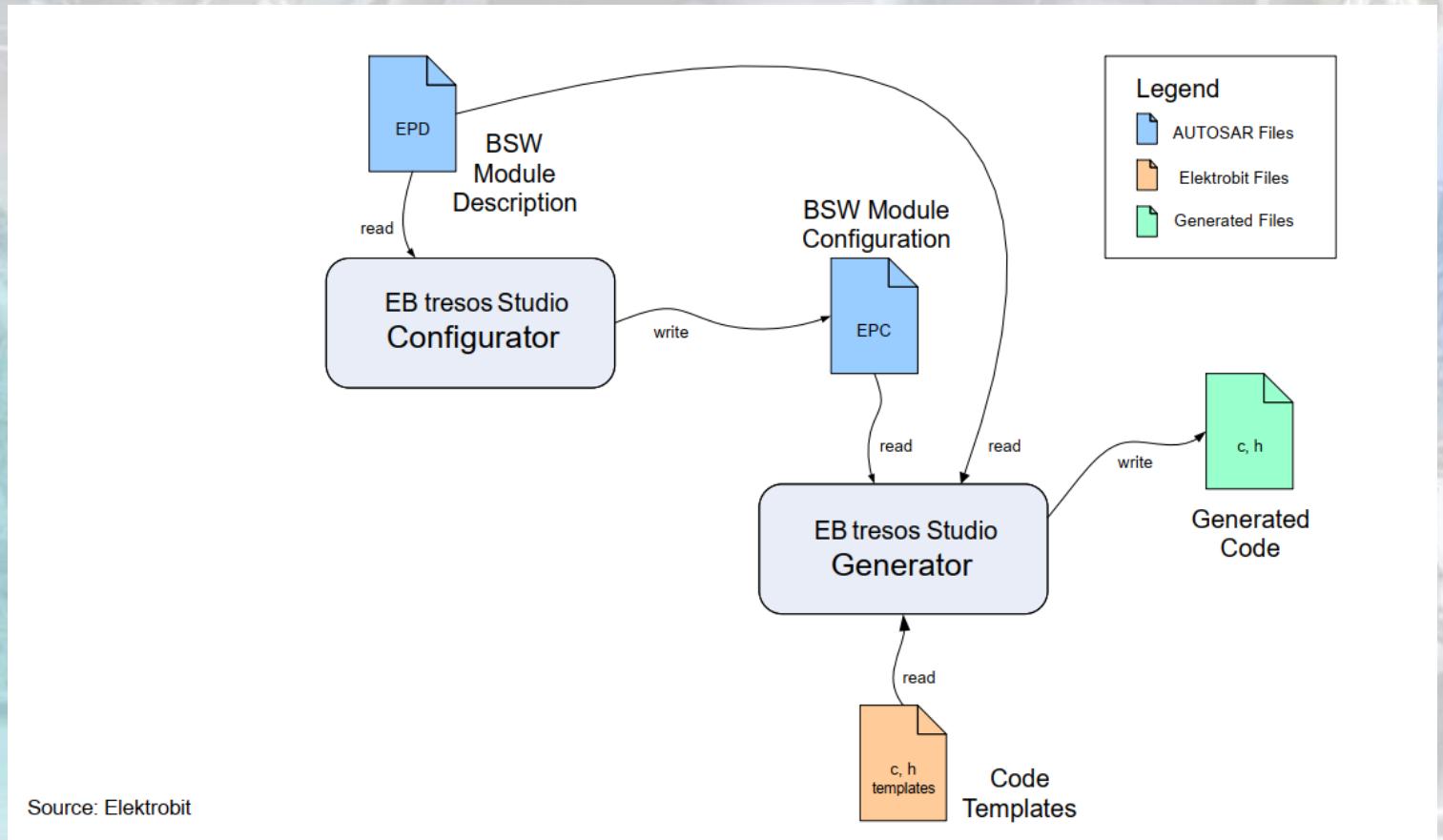
AUTOSAR Module Structure



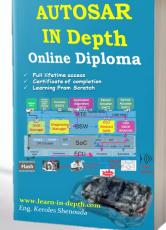


<https://www.facebook.com/groups/embedded.system.KS/>

Example



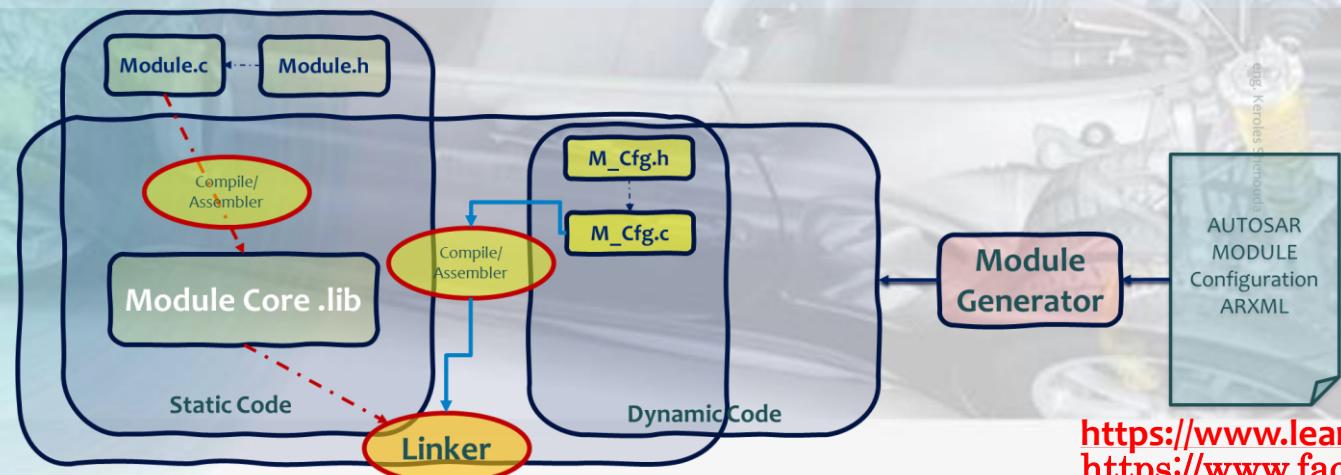
<https://www.learn-in-depth.com/>
<https://www.facebook.com/groups/embedded.system.KS/>



Configuration Classes in AUTOSAR

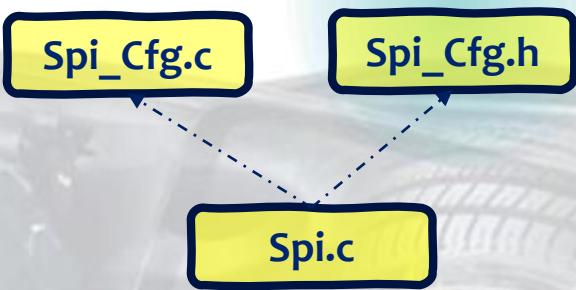
► Pre-compile time :

- ▶ Configuration is used to include or exclude parts of the source code which are not needed during runtime.
- ▶ Pre-compile time configurations are **static** where in the software modules will be effective, based on configuration, after compile time.
- ▶ It results in **optimization of code size and performance**.
- ▶ Pre-compile time configurations are done in file ***_Cfg.c** and ***_Cfg.h**, '*' specifies the module name.



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

Pre-Compile



- ▶ Advantage :
 - ▶ **Save runtime overhead**, complete the configuration in the pre-compilation stage
- ▶ Disadvantage :
 - ▶ Parameter configuration is achieved through macros, and the **configured C must be provided in source code form, not in the form of a library or intermediate file**
 - ▶ The parameter type changes need to **recompile** the software

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

Pre-Compile

- ▶ An example
 - ▶ enable and disable a function

```
const uint8 myconstant = 1U; Spi_Cfg.c #define SPI_DEV_ERROR_DETECT ON
```

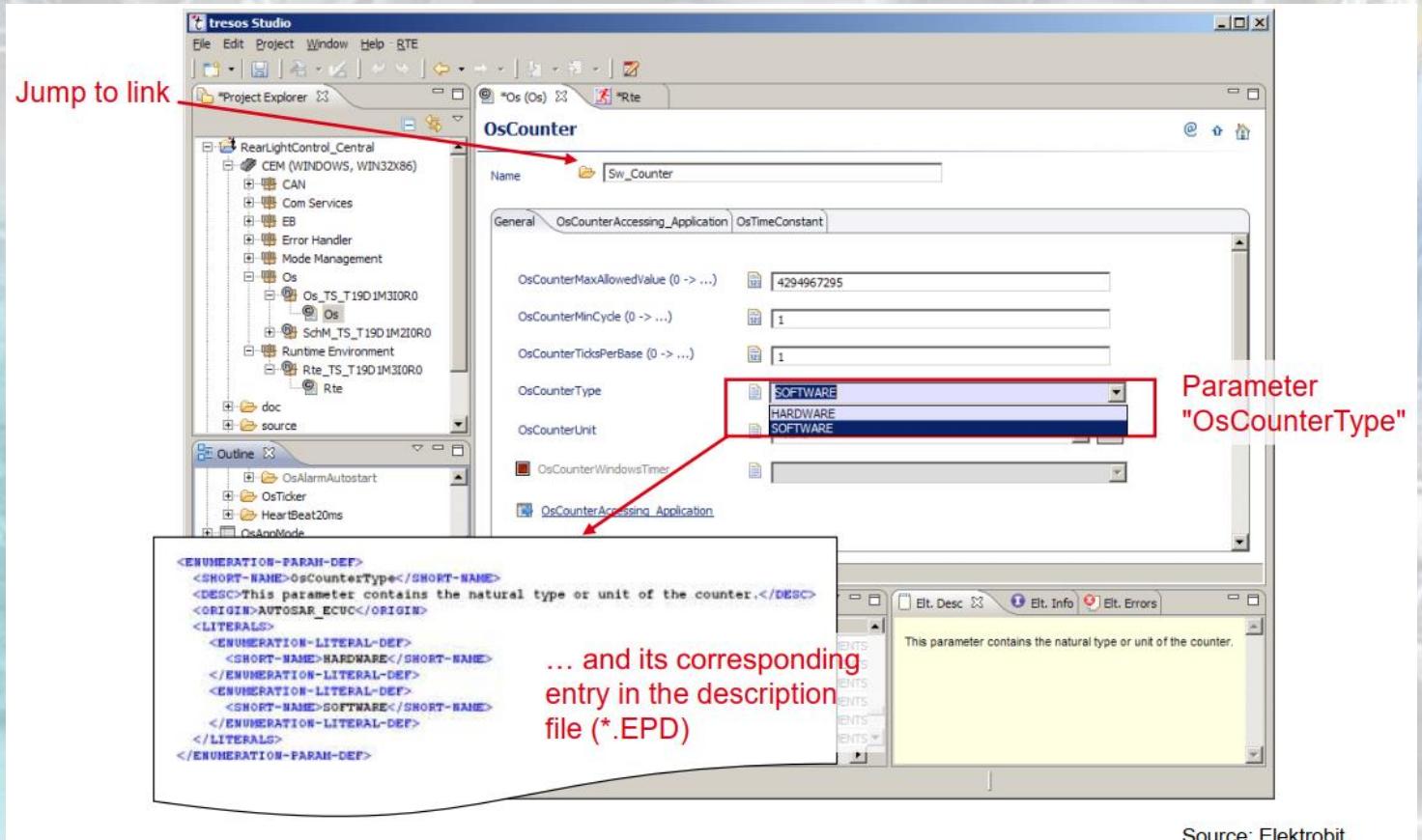
Spi_Cfg.h

Spi.c

```
#include "Spi_Cfg.h" /* for importing the configuration parameters */
extern const uint8 myconstant;
#if (SPI_DEV_ERROR_DETECT == ON)
Det_ReportError(Spi_ModuleId, 0U, 3U, SPI_E_PARAM_LENGTH);
/* only one instance available */
#endif
```

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

Pre-Compile Example 2



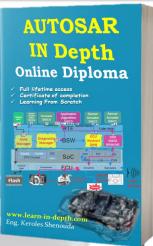
Source: Elektrobit

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

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

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

Eng. Keroles Shenouda



#LEARN_IN_DEPTH
#Be_professional_in_embedded_system

Eng. Keroles Shenouda

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

Pre-Compile Example 2

The screenshot shows the AUTOSAR Configuration tool interface for the 'Dio' component. The component name is 'Dio' and it is selected in the tree view. The 'DioConfig' tab is active. A dropdown menu labeled 'VariantPreCompile' is open, with the 'VariantPreCompile' option highlighted. Below this, the 'DioGeneral' configuration section is expanded, showing the following settings:

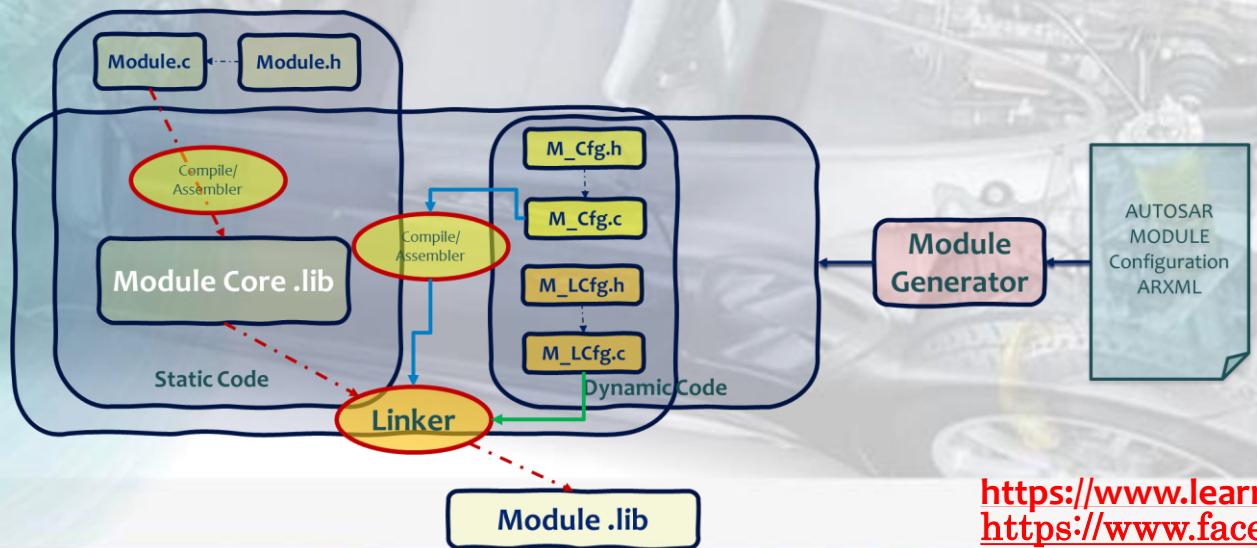
API	Enabled	Configurable
Dio Development Error Detect	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Dio Reverse Port Bits	<input type="checkbox"/>	<input checked="" type="checkbox"/>
Dio Version Info Api	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Dio Flip Channel Api	<input checked="" type="checkbox"/>	<input type="checkbox"/>

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

Configuration Classes in AUTOSAR

▶ Link time :

- ▶ This type of configuration is used when the configuration files are available as object code.
- ▶ The object code of the software receives parts of the configuration from other object code file or it is defined by linker options.
- ▶ Configuration is selected after compilation and before linking.
- ▶ Configuration available in separate files are referred as external constants.
- ▶ Link time configurations are done in files *_Lcfg.c and *_Lcfg.h, '*' specifies the module name.



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

Link-time Example

```
#include "Dem_Cfg.h" /* for providing access to event symbols */
const Dem_EventIdType Fls_WriteFailed[2] =
{DemConf_DemEventParameter_FLS_E_WRITE_FAILED_1,
DemConf_DemEventParameter_FLS_E_WRITE_FAILED_2};
```

Fls_LCfg.c

```
typedef uint16 Dem_EventIdType; /* total number of events = 380
=> uint16 required */
#define DemConf_DemEventParameter_FLS_E_ERASE_FAILED_0 1U
#define DemConf_DemEventParameter_FLS_E_ERASE_FAILED_1 2U
#define DemConf_DemEventParameter_FLS_E_WRITE_FAILED_1 3U
#define DemConf_DemEventParameter_FLS_E_WRITE_FAILED_2 4U
#define DemConf_DemEventParameter_NVM_E_REQ_FAILED 5U
#define DemConf_DemEventParameter_CANSM_E_BUS_OFF 6U
```

Dem_Cfg.h

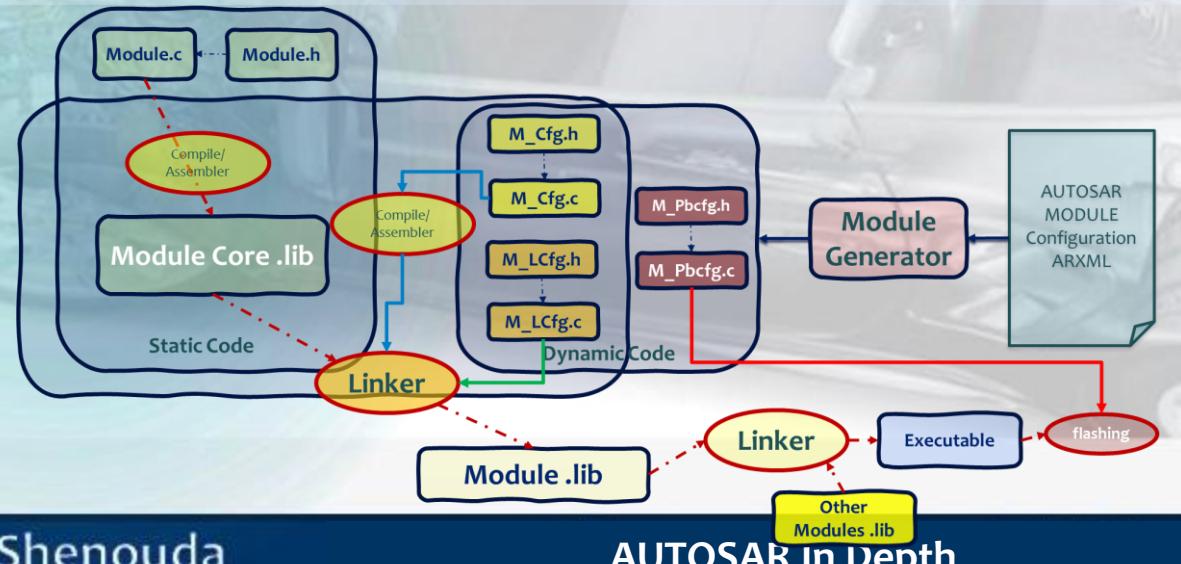
Fls.c (available as object code)

```
#include "Dem.h" /* for reporting production errors */
extern const Dem_EventIdType Fls_WriteFailed[];
Dem_SetEventStatus(Fls_WriteFailed[instance], DEM_EVENT_STATUS_FAILED);
```

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

Configuration Classes in AUTOSAR

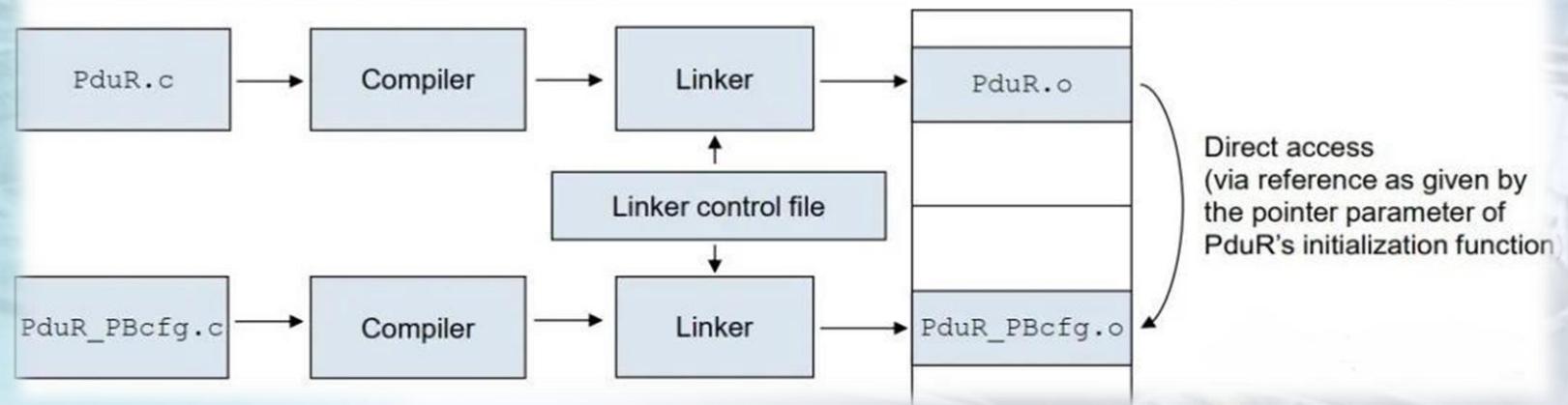
- ▶ Post-build time :
- ▶ Configuration of software module is possible after building the complete software.
- ▶ A reference to the configuration is available and the actual configuration is available during ECU flashing.
- ▶ This increases the re-usability such that the same ECU can be reused in a different car by providing a different set of ECU configuration.
- ▶ Post-build time configurations are done in files *_PBcfg.c and *_PBcfg.h, '*' specifies the module name.



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

Post-build time

- ▶ Post-build allows ECU configuration changes without the need to rebuild the ECU SW.
- ▶ However, not all parameters support post-build time configuration (have post-build class in a post-build BSW module variant).
- ▶ In the context of post-build, it is possible to distinguish between **post-build loadable** and **post-build selectable**:



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

Post-build (Suitable scene)

- ▶ Data configuration suitable for **ECU adaptation**, such as some calibration and calibration parameters after the production line (**these parameters are not known during the development period in advance**. Generally, the minor differences in the hardware of each machine cause each machine The parameters are different), or the serial number and version configuration of each machine belong to the Post-build time configuration.
- ▶ **The reusability of ECUs across different car versions** (same application, different configurations).
 - ▶ For example, compared with the ECU of a luxury car, the ECU of a low-profile car transmits fewer signals on the bus, and other configurations are the same.

<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/>

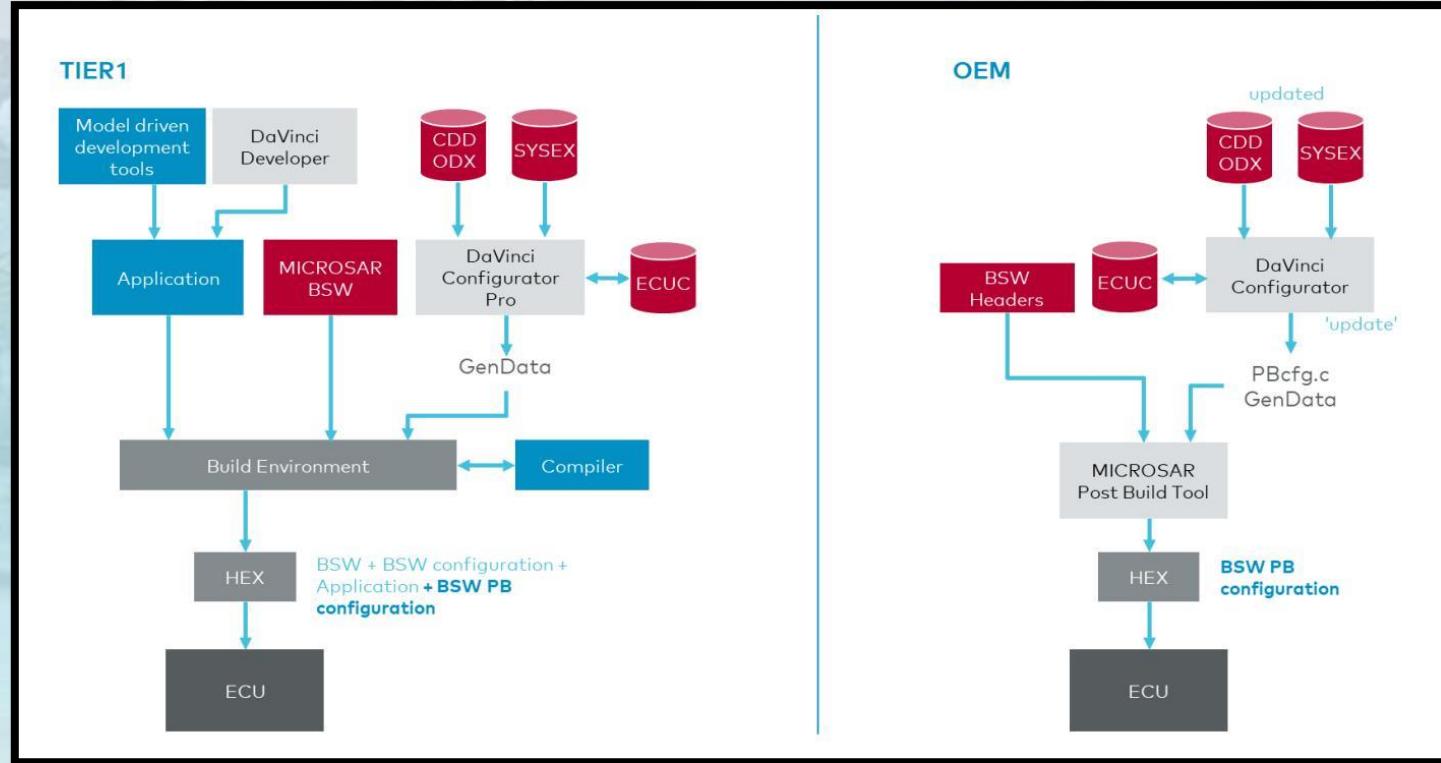
Post-build overview

- ▶ It is to separately put the configured parameters in a memory area,
- ▶ this area can be individually flashed (through the flash erasing tool, through the Bootloader to modify the parameters),
- ▶ one advantage of this is that the OEM does not need to know the code logic.

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

Post-build workflow

- ▶ the Post-build time is modified after the binary is obtained from the link.
- ▶ The following is a Post-build workflow given by Vector:

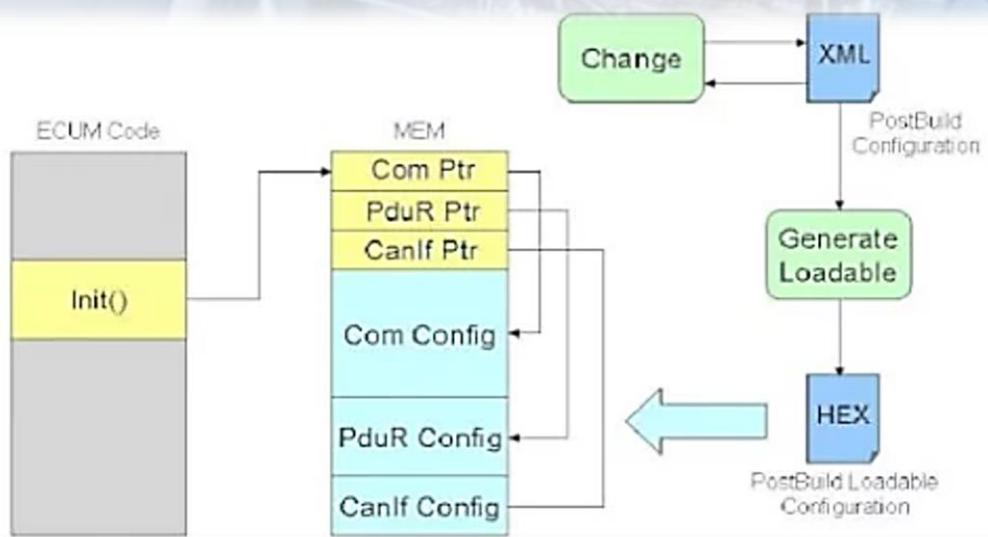


www.learn-in-depth.com/
<https://www.facebook.com/groups/embedded.system.KS/>

Post-build is divided into two types

► Post-Build Loadable:

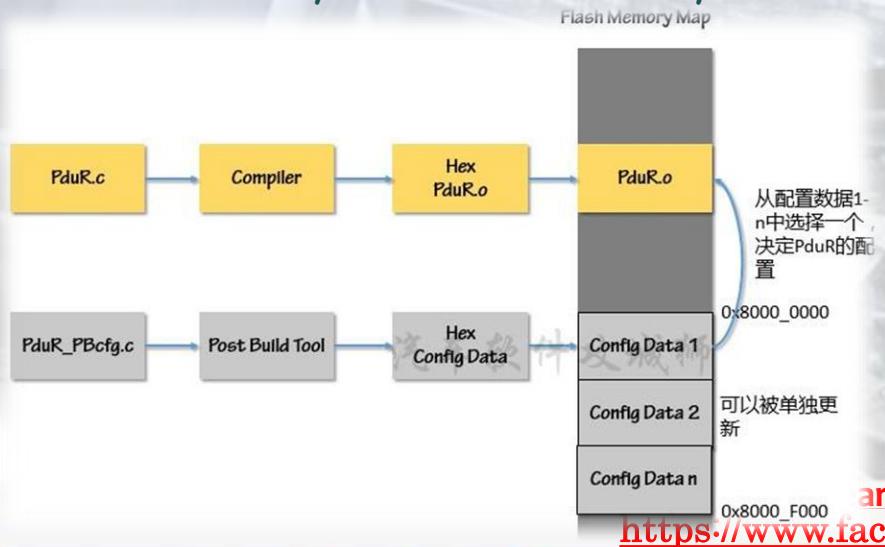
- ▶ There is only one copy of the configuration parameters, which are stored separately in a fixed area (specific address) of the flash.
- ▶ The parameter configuration can be modified and can be updated separately.
- ▶ replaceable configuration is located in a separate flash sector.



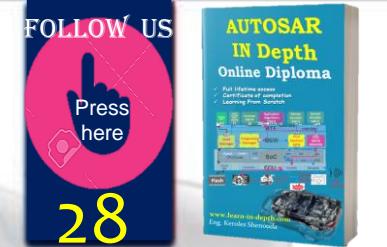
Post-build is divided into two types

► Post-Build Selectable:

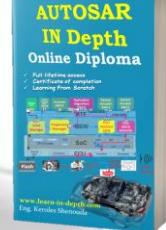
- ECU is configured with **several configurations linked together** in an ECU SW and **one of them is chosen at start-up**
 - (for example software made to control both left and right door modules where the right configuration is chosen in runtime depending on the pin encoding in the HW connector).
- In this configuration set, 'n' configuration sets might be available and one complete set can be selected during ECU flashing. These are mostly available as array of structures wherein one set is selected during ECU flashing



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



28



#LEARN_IN_DEPTH
#Be_professional_in_embedded_system

eng. Keroles Shenouda

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

Post-build initialization

- ▶ The post-build initialization is done in **the ECU State Manager**.
- ▶ In principle, configuring post-build using EcuM Fixed or Flexible require some manual implementation of callouts

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

post-build selectable Example1

Example:

```
FUNC(P2CONST(EcuM_ConfigType_t, ECUM_VAR, ECUM_APPL_CONST),  
ECUM_CODE) EcuM_DeterminePbConfiguration( void )  
{ /* Integrator code - start */  
    EcuM_CpnfigType* myPB = 0;  
    if (ECU_is_Left_Door) {  
        myPB = &myLeftPBConfiguration;  
    } else {  
        myPB = &myRightPBConfiguration;  
    }  
    /* Integration code - end */  
} /* EcuM_DeterminePbConfiguration() */
```

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

post-build Example

voidEcuM_AL_DriverInitOne(constEcuM_ConfigType* ConfigPtr) – This function is called next in order to initialize all modules that have to be initialized before the OS is started (pre-compile, link-time, and post-build configuration selectable or loadable).

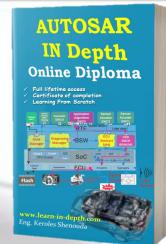
The ConfigPtr is the pointer that is set in the DeterminePBConfiguration callout (see above).

Example:

```
FUNC(void, ECUM_CODE) EcuM_AL_DriverInitOne(P2CONST(EcuM_ConfigType_t, ECUM_VAR,
ECUM_APPL_CONST), EcuM_CfgPtr )
{
    /* Automatically generated init calls */
    Mcu_Init(EcuM_CfgPtr->EcuMMModuleConfigurationRef->EcuMMcuCfgPtr);
    WdgM_Init(EcuM_CfgPtr->EcuMMModuleConfigurationRef->EcuMWdgMCfgPtr);

    /* Post build loadable init calls */
    Com_Init(PBLoadableROM[0]);
    PduR_Init(PBLoadableROM[1]);
} /* End of EcuM_AL_DriverInitOne() */
```

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



#LEARN_IN_DEPTH
#Be_professional_in_embedded_system

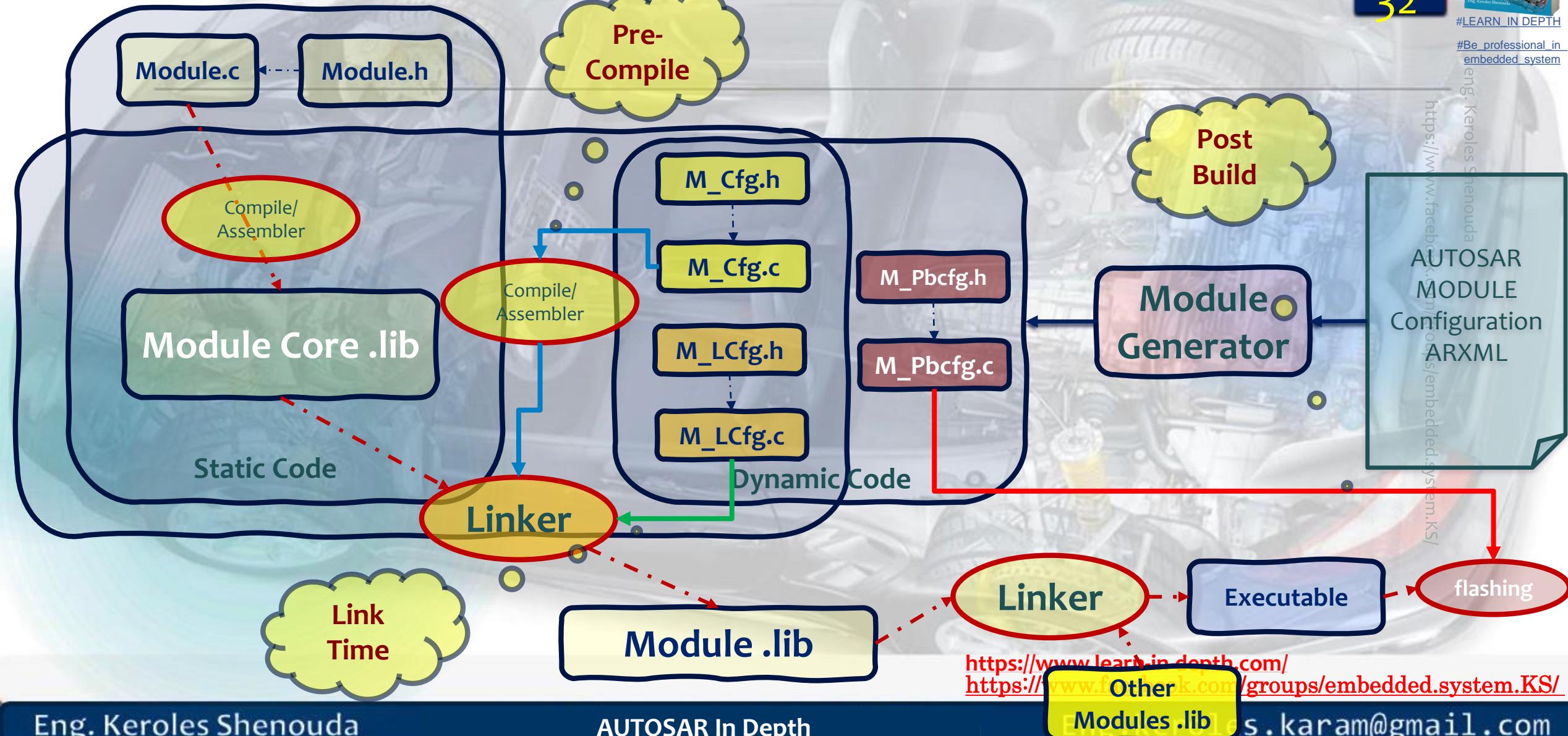
https://www.facebook.com/groups/embedded.system.KS/

The screenshot shows the configuration of a port named "Port". The "VariantPostBuild" variant is selected. The "PortGeneral" section is expanded, showing the configuration of four APIs:

- Port Development Error Detect: Enabled (checked)
- Port SetPinDirection Api: Enabled (checked)
- Port SetPinMode Api: Enabled (checked)
- Port VersionInfo Api: Enabled (checked)

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

AUTOSAR Module Structure





33

eng. Keroles Shenouda

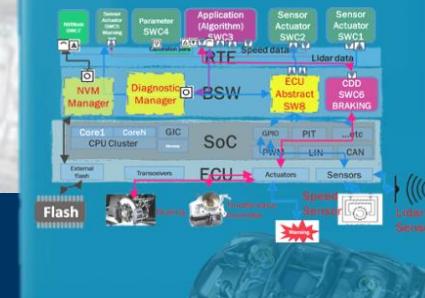
#LEARN_IN_DEPTH
#Be_professional_in
embedded_system

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

AUTOSAR		Specification of C Implementation Rules V 1.0.1	
Document Title		Specification of C Implementation Rules	
Document Owner		AUTOSAR GbR	
Document Responsibility		AUTOSAR GbR	
Document Version		1.0.1	
Document Status		Final	
Document Change History			
Date	Version	Changed by	Change Description
27.06.2006	1.0.1	AUTOSAR Administration	Layout Adaptations
09.05.2006	1.0.0	AUTOSAR Administration	Initial Release

AUTOSAR IN Depth Online Diploma

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



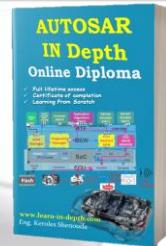
www.learn-in-depth.com
Eng. Keroles Shenouda

LEARN-IN-DEPTH
Be professional in
embedded system

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



34

#LEARN_IN_DEPTH
#Be_professional_in_embedded_system

eng. Keroles Shenouda

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

8.4.2 Pre compile time implementation

This is the simplest option and can result in the most efficient implementation. This is because the code generator can produce code and data that is optimised for the configuration being generated.

The entire module has to be delivered as source code. In this example there are four source files:

xx.c This is part of the implementation of the module and, therefore, part of the delivery of the module and would not normally be changed by the user of the module.

xx_structs.h These are data structure declarations that the module's implementation uses. This is part of the delivery of the module and, therefore, would not normally be changed by the user of the module.

xx_cfg.h This file is generated automatically by the code generator from the AUTOSAR XML. It just contains macro definitions. The only changes to this file can be made by altering the XML and regenerating the file.

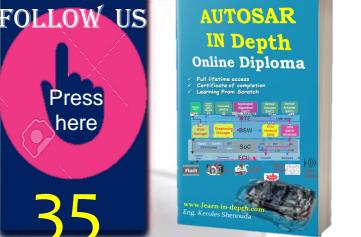
26 of 44

AUTOSAR_SWS_C_ImplementationRules

- AUTOSAR confidential -

xx_cfg.c This file is generated automatically by the code generator from the AUTOSAR XML. It contains definitions of data used by xx.c.

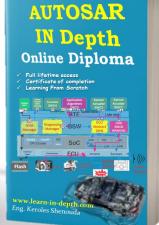
<https://www.learn-in-depth.com/>
<https://www.facebook.com/groups/embedded.system.KS/>



35

Eng. Keroles Shenouda

#LEARN IN DEPTH
#Be professional in
embedded system



```
xx_structs.h
/* Data structures that are part of the implementation
 * and should not be changed.
 */
struct buffer {
    /* size of data in this buffer */
    unsigned int size;
    /* pointer to initial value for data in buffer */
    const unsigned char *init_data;
    /* pointer to ram for buffer */
    unsigned char *ram_data;
};

/* flag to state whether or not this module has been initialised.*/
extern unsigned char module_initialised;

extern const struct buffer my_configuration[];

void xx_memcpy(unsigned int size,
              const unsigned char *src,
              unsigned char *dest);
```

xx.c

```
#include "xx_structs.h"
#include "xx_cfg.h"

void f(void) {
    unsigned int count;

#ifdef DEVELOPMENT_ERROR_CHECKING
    if(!module_initialised) {
        /* do something with the DET */
        return;
    }
#endif

    /* set CAN Baud rate */
    *BAUD_RATE_REGISTER = BAUD_RATE;

    /* initialise the buffers in RAM from data in ROM*/
    for(count=0; count<NBUFFS; count++) {
#ifdef CACHING_IS_BETTER
        /* this code caches the pointer */
        struct buffer *tmp = &my_configuration[count];
        xx_memcpy(tmp->size, tmp->init_data, tmp->ram_data);
#else
        /* Caching is not better */
        xx_memcpy(my_configuration[count]->size,
                  my_configuration[count]->init_data,
                  my_configuration[count]->ram_data);
#endif /* CACHING_IS_BETTER */
    }
}
```

xx_cfg.h

```
/* AUTOMATICALLY GENERATED FILE - DO NOT EDIT */

/* These defines may be automatically generated depending
 * upon the module implementation
 */
#define BAUD_RATE (0xf0)
#define BAUD_RATE_REGISTER ((unsigned char *) (0xfe00))

/* DET is turned on in the XML. This define would be omitted
 * if DET is not required
 */
#define DEVELOPMENT_ERROR_CHECKING

/* number of entries in my_configuration */
#define NBUFFS ((unsigned int)2)

/* Pointer caching is better on this target */
#define CACHING_IS_BETTER
```

xx_cfg.c

```
/* AUTOMATICALLY GENERATED FILE - DO NOT EDIT */

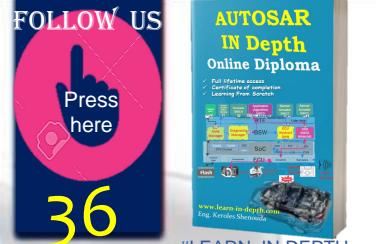
#include "xx_structs.h"

/* RAM buffers for different variables */
unsigned char buffer_bob[4];
unsigned char buffer_bill[3];

/* ROM initial values for buffers */
const unsigned char initial_bob[] = {0x0, 0x23, 0x12, 0xe4};
const unsigned char initial_bill[] = {0xdf, 0x3d, 0x12};

/* ROM generated configuration information for module */
const struct buffer my_configuration[] = {
    {4, initial_bob, buffer_bob},           /* buffer for bob */
    {3, initial_bill, buffer_bill},          /* buffer for bill */
};
```

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



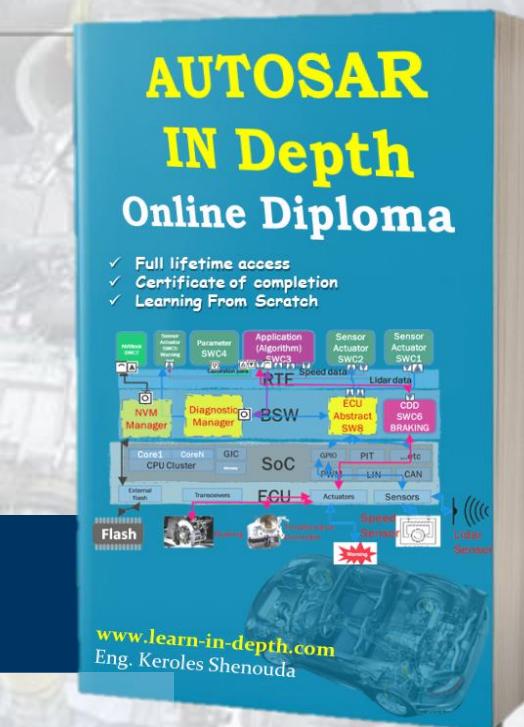
36

eng. Keroles Shenouda

#LEARN_IN_DEPTH
#Be_professional_in
embedded_system

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

AUTOSAR		Specification of C Implementation Rules V 1.0.1	
Document Title		Specification of C Implementation Rules	
Document Owner		AUTOSAR GbR	
Document Responsibility		AUTOSAR GbR	
Document Version		1.0.1	
Document Status		Final	
Document Change History			
Date	Version	Changed by	Change Description
27.06.2006	1.0.1	AUTOSAR Administration	Layout Adaptations
09.05.2006	1.0.0	AUTOSAR Administration	Initial Release



LEARN-IN-DEPTH
Be professional in
embedded system

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

Link time implementation

Three files are needed for this: two .c files and a header file that ensures consistency between the .c files. Brief examples are shown below.

link_time.h

```
extern const int fred;
```

user.c

```
#include "link_time.h"
int get_fred(void) {
    return fred;
}
```

init.c

```
#include "link_time.h"
const int fred = 0x1234;
```

The file user.c uses fred but does not know in advance what its value is. The file is compiled by the supplier of the module and, therefore, cannot be changed by the user of the module. The file link_time.h is also supplied as part of the module, because it is necessary to compile a file in the user's domain, and cannot be changed by the user.

The file init.c is generated and compiled by the user. When user.o and init.o are linked the reference by user.c to fred is resolved so that the function get_fred() can work correctly.

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



42

#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/>

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/>



44

#LEARN_IN_DEPTH
#Be_professional_in_embedded_system

eng. Keroles Shenouda

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/>



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

Eng. Keroles Shenouda

#LEARN_IN_DEPTH
#Be_professional_in
embedded_system

45

References

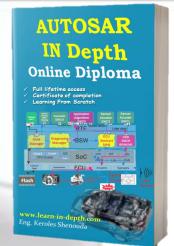
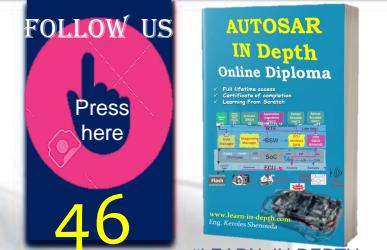
- ▶ Introduction to AUTOSAR, Stephen Waldron, **Vector webinar**
Wednesday 7th May 2014

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

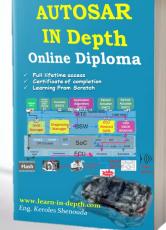
References

- ▶ 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/>



48

#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/>



49

#LEARN_IN_DEPTH
#Be_professional_in_embedded_system

Eng. Keroles Shenouda

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

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_layersoftwarearchitecture.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/>



51

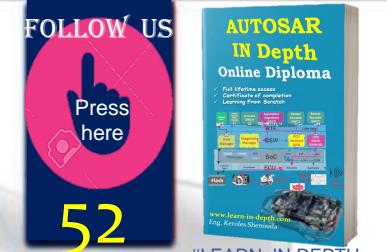
#LEARN_IN_DEPTH
#Be_professional_in_embedded_system

Embedded_system_KS

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/>



52

#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=ZmtlLnVObS5teXxyaWR6dWFuLXMtd2Vic2I0ZXxneDo2ODU0NzIKM2JkOTg4MjRk>
- ▶ <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

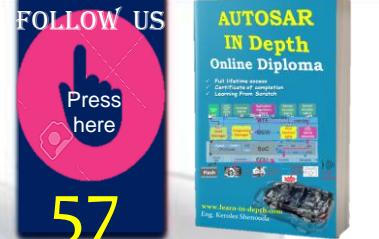
- ▶ Tiva™ TM4C123GH6PM Microcontroller DATA SHEET
- ▶ Interrupts and Exceptions COMS W6998 Spring 2010
- ▶ THE AVR MICROCONTROLLER. AND EMBEDDED SYSTEMS Using Assembly and C. Muhammad Ali Mazidi.
- ▶ <http://embedded-lab.com/blog/tinkering-ti-msp430f5529/27/>
- ▶ [How to use JIRA \(AR\)](#)

<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/>



57

#LEARN_IN_DEPTH
#Be_professional_in_embedded_system

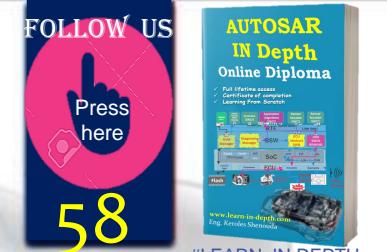
eng. Keroles Shenouda

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

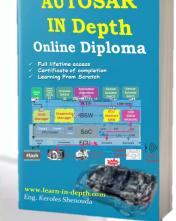
References

- ▶ <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/>



58



www.learn-in-depth.com

Eng. Keroles Shenouda

#LEARN_IN_DEPTH

#Be_professional_in

embedded_system

A large, three-dimensional, stylized text graphic reading "Thank You" in a bold, rounded font. The letters are primarily light blue with a white fill and a thick black outline. A thin brown string with a small metal loop hangs from the top center of the letter "k", giving it the appearance of a hanging tag or a gift message.

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