MCAL User Manual for Port

32-bit TriCore^{TM} AURIX^{TM} TC3xx microcontroller

About this document

Scope and purpose

This User Manual is intended to enable users to integrate the Microcontroller Abstraction Layer (MCAL) software for the TriCore^{TM} AURIX^{TM} family of 32-bit microcontrollers.

This document describes responsibilities of integrator in-charge of integrating MCAL software with the basic software (BSW) stack. This document also provides detailed information on safety, configuration and functions along with examples of usage of significant features.

Note: Detailed information about package installation, safety and other generic information that are common across all modules are provided in MCAL User Manual General.

Intended audience

This document is intended for anyone using the Port module of the TC3xx MCAL software.

Document conventions

Table 1 **Conventions**

Convention	Explanation
Bold	Emphasizes heading levels, column headings, table and figure captions, screen names, windows, dialog boxes, menus, sub-menus
Italics	Denotes variable(s) and reference(s)
`Courier New`	Denotes APIs, functions, interrupt handlers, events, data types, error handlers, file/folder names, directories, command line inputs, code snippets
>	Indicates that a cascading sub-menu opens when you select a menu item
[cover parentID=<alpha numeric value>]	Used for traceability completeness. Reader should ignore these.

Reference documents

This User Manual should be read in conjunction with the following documents:
- AURIX^{TM} TC3xx MCAL User Manual General
- Specification of PORT Driver, AUTOSAR_SWS_PORT_Driver, AUTOSAR Release 4.2.2
- Specification of PORT Driver, AUTOSAR_SWS_PORT_Driver, AUTOSAR Release 4.4.0

Table of contents

Table of contents

Table of contents

1 Port driver

1.1 User information

1.1.1 Description

The PORT driver helps user in assigning of port pins to peripherals and configuring characteristics or features provided by the underlying hardware. Being limited in availability, these port pins are shared among several on-chip peripherals. However, at any given point in time, a port pin is assigned to and used exactly by one peripheral.

1.1.2 Hardware-software mapping

This section describes the system view of the PORT driver and peripherals administered by it.
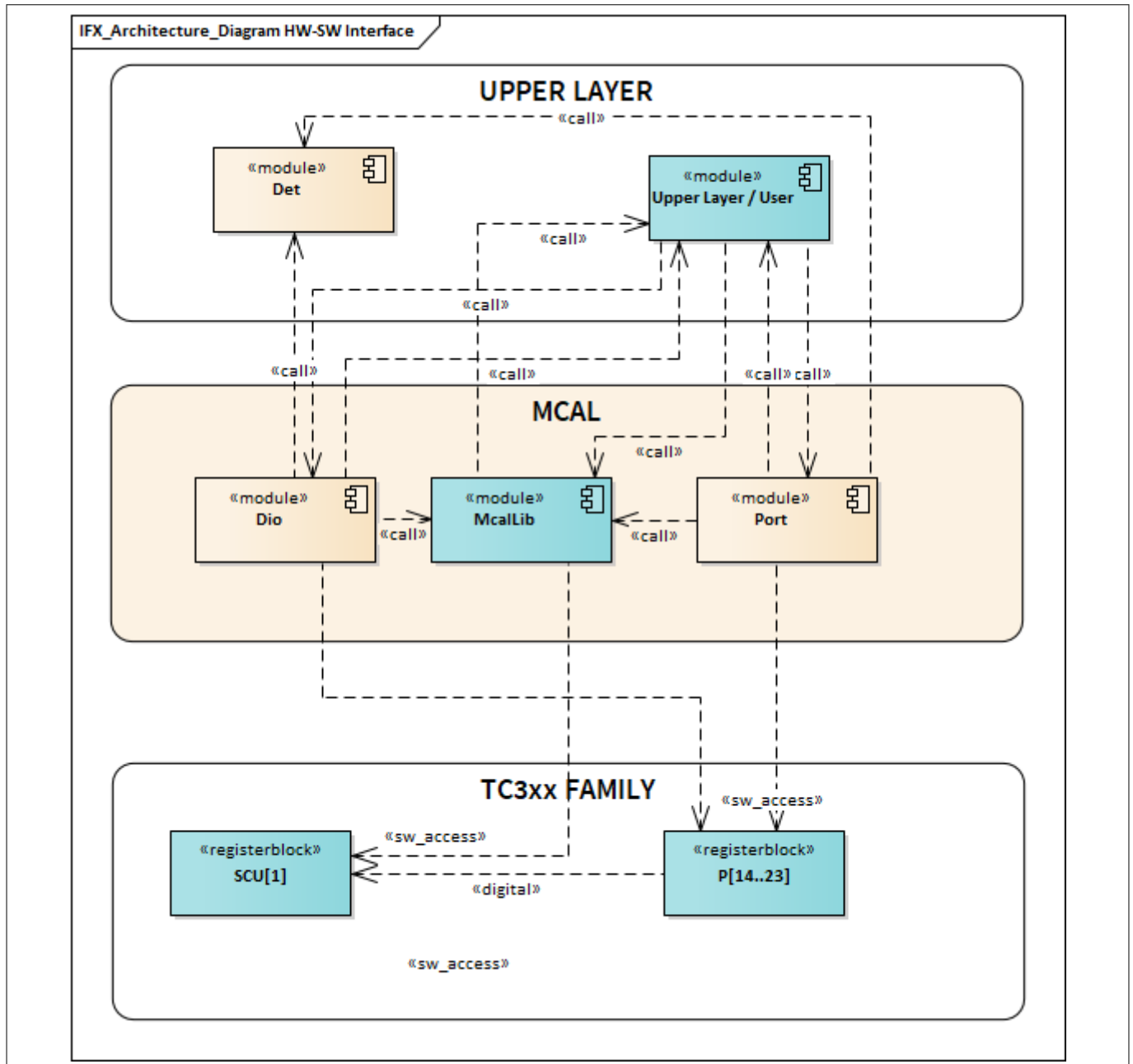
Figure 1   Mapping of hardware-software interfaces

1.1.2.1   PORT: primary hardware peripheral

Hardware functional features

The PORT driver uses the PORT peripheral for configuration of PORT driver. The key hardware functional features used by the driver are:

• Pin direction

• Alternate pin mode

• Pad driver strength

• Pad level

• Pull up/Pull down

• Digital/Analog mode

- Low voltage differential signalling (LVDS) characteristic
- Emergency stop

The unsupported features of the PORT are:

None.

Users of the hardware

Port pins are used by many MCAL drivers, but configuring the port pins is done through the PORT driver. The DIO driver also writes into the registers of port, hardware to toggle, set or clear a port pin. Both the drivers program the registers atomically to avoid corruption due to concurrent updates.

Hardware diagnostic features

Not applicable.

Hardware events

Not applicable.

1.1.2.2 SCU: dependent hardware peripheral

Hardware functional features

The PORT driver depends on the SCU IP for the clock, ENDINIT and reset functionalities. The driver requires the SPB clock signals for functioning.

Users of the hardware

The SCU IP supplies clock for all the peripherals and the MCU driver is responsible for configuring the clock tree. To avoid conflicts due to simultaneous writes, update to all the ENDINIT protected registers is performed using the MCALLIB APIs.

Hardware diagnostic features

The SMU alarms configured for the SCU IP are not monitored by the PORT driver.

Hardware events

Hardware events from the SCU are not used by the PORT driver.

1.1.3 File structure

1.1.3.1 C file structure

This section provides the details of the C files of the PORT driver.

1 Port driver
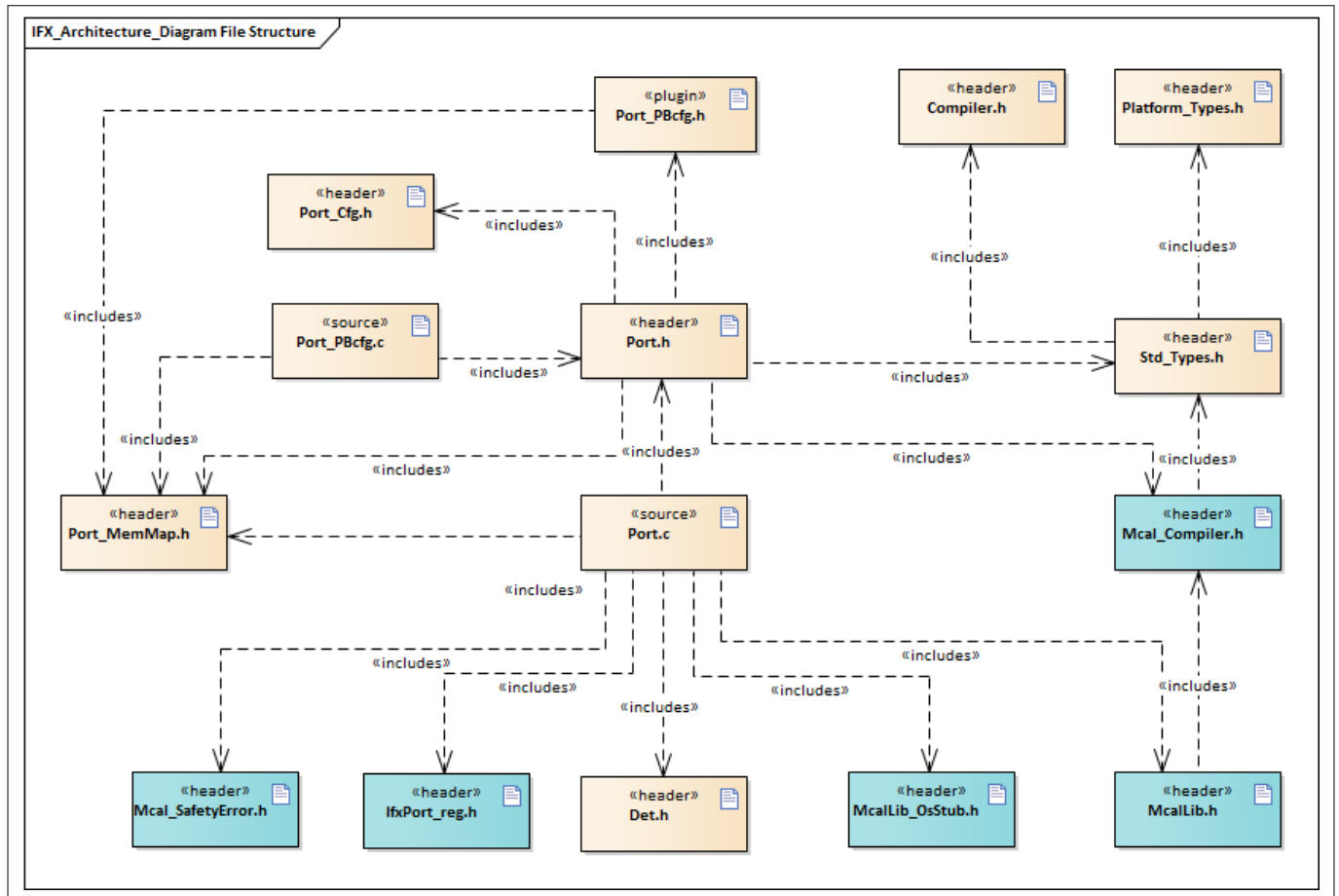
Figure 2   Port_File_Structure-1.png

Table 2   C file structure

File name	Description
Compiler.h	Provides abstraction from compiler-specific keywords
Det.h	Provides the exported interfaces of Development Error Tracer
IfxPort_reg.h	SFR header file for Port
McalLib.h	Static header file defining prototypes of data structure and APIs exported by the MCALLIB.
McalLib_OsStub.h	McalLib_OsStub.h provides macros to support user mode of Tricore. This shall be included by other drivers to call OS APIs.
Mcal_Compiler.h	Header file providing abstraction for TriCore™-intrinsic instruction.
Mcal_SafetyError.h	Header file containing the prototype of the API for reporting safety-related errors
Platform_Types.h	Platform-specific type declaration file as defined by AUTOSAR
Port.c	File (static) containing implementation of APIs
Port.h	Header file (static) defining prototypes of data structures and APIs
Port_Cfg.h	Header file (generated) containing constants and pre-processor macros
Port_MemMap.h	File (static) containing the memory section definitions used by the PORT driver

Table 2   **C file structure (continued)**

File name	Description
Port_PBcfg.c	File (generated) containing post-build configuration data structures
Port_PBcfg.h	File (generated) containing declaration of the post-build configuration data structures
Std_Types.h	Standard type declaration file as defined by AUTOSAR. It is independent of compiler or platform.

1.1.3.2   Code generator plugin files

This section provides the details of the code generation plugin files of the PORT driver.
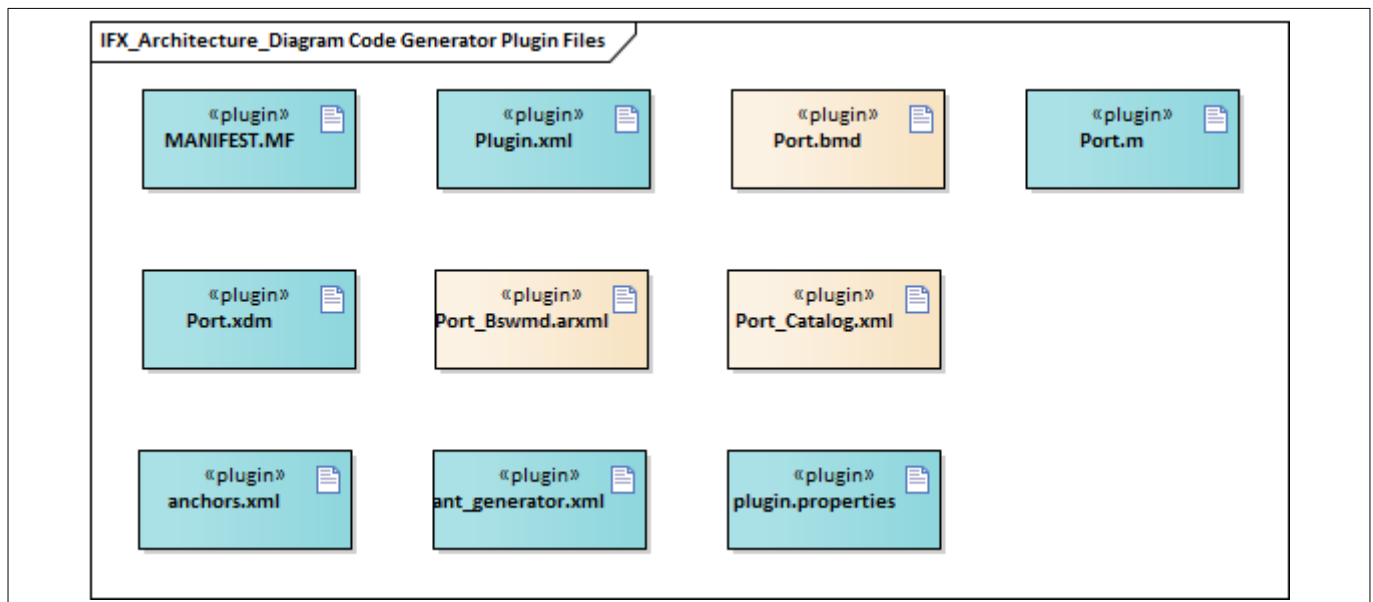
Figure 3   **Port_Code_Generator_Plugin_Files-1.png**

Table 3   **Code generator plugin files**

File name	Description
MANIFEST.MF	Tresos plugin support file containing the metadata for PORT driver
Plugin.xml	Tresos plugin support file for the PORT driver
Port.bmd	AUTOSAR format XML data model schema file (for each device)
Port.m	Code template macro file for PORT driver
Port.xdm	Tresos format XML data model schema file
Port_Bswmd.arxml	AUTOSAR format driver description file
Port_Catalog.xml	AUTOSAR format catalog file as per catalog_V3_0_0.ml.xsd
anchors.xml	Tresos anchors support file for the PORT driver
ant_generator.xml	Tresos support file to generate and rename multiple post-build configuration when using variation point

Table 3	**Code generator plugin files (continued)**
File name	**Description**
`plugin.properties`	Tresos plugin support file for the PORT driver

1.1.4 Integration hints

This section lists the key points that an integrator or the user of PORT driver must consider.

1.1.4.1 Integration with AUTOSAR stack

This section lists the modules, which are not part of MCAL, but are required to integrate the PORT driver.

- **EcuM**

  The ECU Manager module is a part of the AUTOSAR stack that manages common aspects of ECU. Specifically, in the context of the MCAL,the EcuM is used for initialization and de-initialization of the software driver.The EcuM module provided in the MCAL package is a stub code and needs to be replaced with a complete EcuM module during the integration phase.

- **Memory mapping**

  Memory mapping is a concept from AUTOSAR that allows relocation of text, variables, constants and configuration data to user specific memory regions.To achieve this, all the relocatable elements of the driver are encapsulated in different memory-section macros.The macros are defined in the `Port_MemMap.h`file.

  The `Port_MemMap.h` file is provided in the MCAL package as a stub code.Integrator must place appropriate compiler pragmas within the memory-section macros.The pragmas ensure that the elements are relocated to the correct memory region.A sample implementation listing the memory section macros is shown as follows.

```
#if defined PORT_START_SEC_VAR_CLEARED_ASIL_B_GLOBAL_32
/*your pragma here*/
#undef PORT_START_SEC_VAR_CLEARED_ASIL_B_GLOBAL_32

#elif defined PORT_STOP_SEC_VAR_CLEARED_ASIL_B_GLOBAL_32
/*your pragma here*/
#undef PORT_STOP_SEC_VAR_CLEARED_ASIL_B_GLOBAL_32

#elif defined PORT_START_SEC_CODE_ASIL_B_GLOBAL
/*your pragma here*/
#undef PORT_START_SEC_CODE_ASIL_B_GLOBAL

#elif defined PORT_STOP_SEC_CODE_ASIL_B_GLOBAL
/*your pragma here*/
#undef PORT_STOP_SEC_CODE_ASIL_B_GLOBAL
#endif

#if defined MEMMAP_ERROR
#error "Port_MemMap.h, wrong pragma command"
#endif
```

- **DET**

The DET module is a part of the AUTOSAR stack that handles all the development and runtime errors reported by the BSW modules.The PORT driver reports all the development errors to the DET module through the `Det_ReportError()` API.The user of the Port driver must process all the errors reported to the DET module through the `Det_ReportError()` API.

The `Det.h` and `Det.c` files are provided in the MCAL package as a stub code and needs to be replaced with a complete DET module during the integration phase.

- **DEM**

  The DEM module is not required for integrating the PORT driver.

- **SchM**

  The SchM is not required for integrating the PORT driver.

- **Safety error**

  The Port driver will report all the detected safety errors through the `Mcal_ReportSafetyError()` API.The driver performs only detection and reporting of the safety errors. Handling of the reported errors shall be done by the user.The `Mcal_ReportSafetyError()` API is provided in the `Mcal_SafetyError.c` and `Mcal_SafetyError.h` files as a stub code, and must be updated by the integrator to handle the reported errors.

  Note: All DET errors are also reported as safety errors (error code used is same as DET).

- **Notifications and callbacks**

  The Port driver does not provide any call-backs or notifications.

- **Operating system**

  The OS or the application must ensure correct type of service and interrupt priority is configured in the SR register.Enabling and disabling of interrupts must also be managed by the OS or application. The OS files provided by MCAL package are only an example code and must be updated by the integrator with the actual OS files for the desired function.

1.1.4.2 Multicore and Resource Manager

The PORT driver supports the multicore concept. All the APIs except `Port_Init` can be accessed from any core. The `Port_Init` API needs to be called only once from any core.

1.1.4.3 MCU support

The Port driver does not use any services provided by the MCU driver.

1.1.4.4 Port support

Port is a central module and supports initialization of port pin for all the other drivers.

1.1.4.5 DMA support

The PORT driver does not use any services provided by the DMA driver.

1.1.4.6 Interrupt connections

The PORT driver does not use any interrupt source.

1.1.4.7 Example usage

Initialization of PORT driver

Initialization of PORT driver is done by calling the `Port_Init` API .

```
/* Include Port.h to access configuration file*/
#include "Port.h"

/* MCU initializations */
Mcu_Init(&Mcu_Config);

Mcu_Init(&Mcu_Config);
(void)Mcu_InitClock( 0 );
while(Mcu_GetPllStatus() != MCU_PLL_LOCKED)
{
};

/* Port Initialization */
Port_Init(&Port_Config);
```

User must ensure that the `Port_Init` API is called before using any other API provided by PORT driver .
Development error is reported if user calls other PORT driver API before calling `Port_Init` .

Changing the direction of the pin

```
/* Port Initialization */
Port_Init(&Port_Config);

Port_SetPinDirection(PortConf_PortContainer_0_PORT_0_PIN_0, PORT_PIN_OUT);
```

The PORT driver provides the API to change the direction of the pin during runtime . User must enable
`PortPinDirectionChangeable` in the configuration tool for the required pin.The API is available only when
the `PortSetPinDirectionApi` is switched ON .

Refreshing direction of all configured pins

The PORT driver provides the API to refresh the direction of all the configured pins . The API does not refresh the
direction of pins, where pin direction is configured as changeable .

```
/* Port Initialization */
Port_Init(&Port_Config);

Port_RefreshPortDirection();
```

Set port pin mode to another alternate mode

1 Port driver

The PORT driver provides the API to set the port pin mode of referenced pin at runtime.This API is available only when the `PortSetPinModeApi` is switched ON .

```
/* Port Initialization */
Port_Init(&Port_Config);

Port_SetPinMode(PortConf_PortContainer_0_PORT_0_PIN_0,PORT_PIN_MODE_ALT3);
```

Port_InitCheck

The PORT driver provides `Port_InitCheck` to check the initialization value is correct after PORT is initialized . It should be called after `Port_Init` whether API returns E_OK or E_NOT_OK .

```
/* Port Initialization */
Port_Init(&Port_Config);

/*Complete other module initialization*/

result= Port_InitCheck (&Port_Config);
```

1.1.5 Key architectural considerations

1.1.5.1 Pin support for Ethernet

The PortPinOutputPadDriveStrength configuration parameter can be set to PORT_PIN_RGMII_DRIVER for port pins that are RFAST and keeping direction OUT and used by GETH .

1.1.5.2 LCK bit not checked before writing to PCSR Register by Port_Init API

All the pins must be configured and initialized under the purview of the PORT driver. If some pins are configured and initialized outside the PORT driver, the LCK bit must not be set before calling the Port_Init. If the LCK bit is set, Port_Init cannot configure the respective PCSR register.

1.2 Assumptions of Use (AoU)

The AoU for the PORT driver are as follows.

- **Configuration check**

User shall check that the generated configuration code is correct as per the GUI configuration.
[cover parentID PORT={4AF3131A-7647-47c5-9AE4-AFD673334476}]

- **InitCheck**

If configured, Port_InitCheck API shall be invoked by the application or user to validate successful initialization of the Port_Init API. The Port_InitCheck API shall be invoked before starting any functionality of the PORT driver.
[cover parentID PORT={2F1F7426-F632-45ed-B43B-D922AF98DC75}]

1.3 Reference information

1.3.1 Configuration interfaces
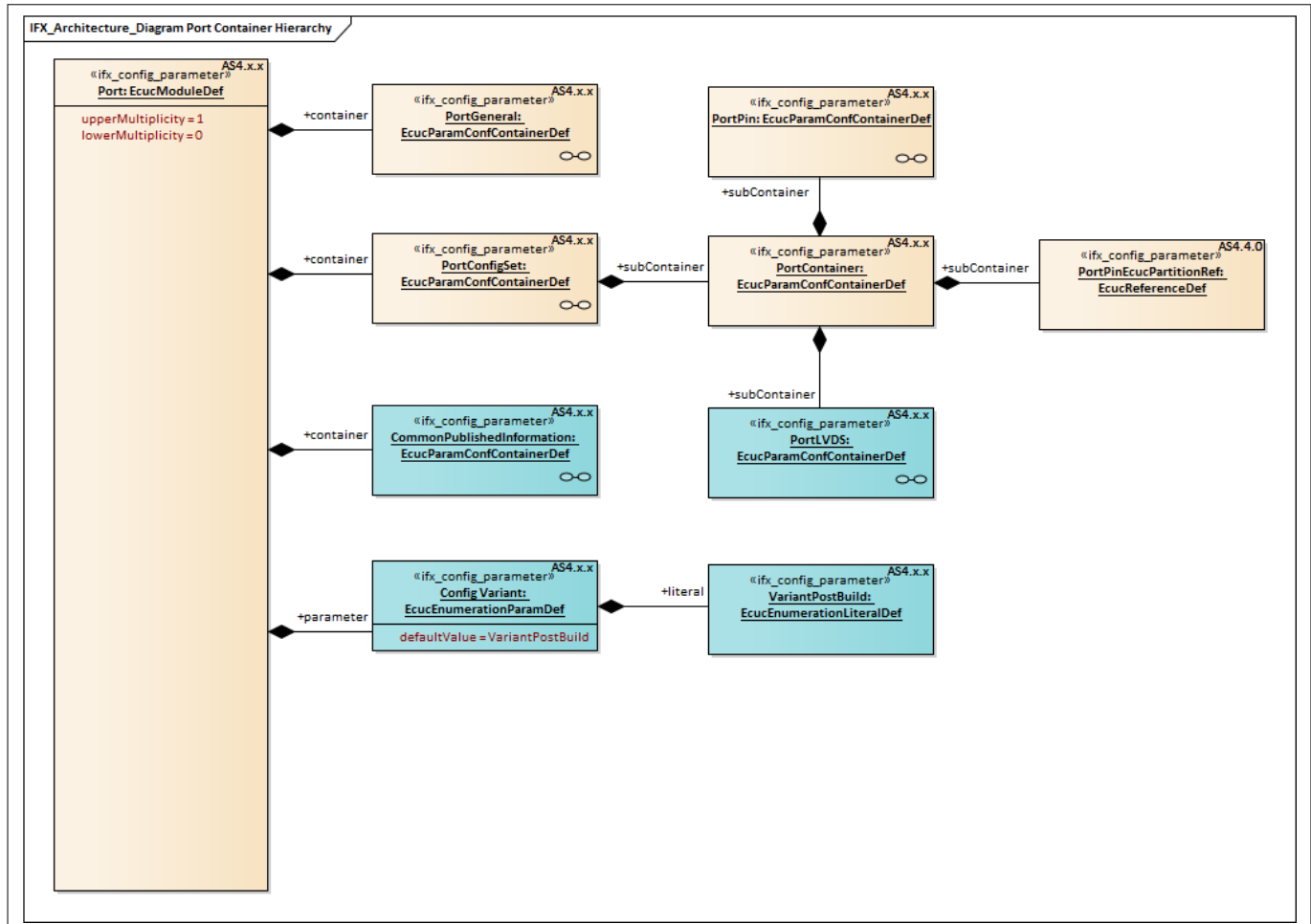
Figure 4 **Container hierarchy along with their configuration parameters**

1.3.1.1 Container: CommonPublishedInformation

This container holds all the published information of the PORT driver.

Post-Build Variant Multiplicity: -

Multiplicity Configuration Class: -

1.3.1.1.1 ArMajorVersion

Table 4 **Specification for ArMajorVersion**

Name	`ArMajorVersion`		
Description	Major version number of the AUTOSAR specification on which the implementation is based on.		
Multiplicity	1..1	**Type**	EcucIntegerParamDef
Range	0 - 255		

Table 4	Specification for ArMajorVersion (continued)		
Default value	4		
Post-build variant value	FALSE	**Post-build variant multiplicity**	-
Value configuration class	Published-Information	**Multiplicity configuration class**	-
Origin	IFX	**Scope**	LOCAL
Dependency	-		
Autosar Version	Applicable for Autosar versions 4.2.2 and 4.4.0.		

1.3.1.1.2   ArMinorVersion

Table 5	Specification for ArMinorVersion		
Name	ArMinorVersion		
Description	Minor version number of the AUTOSAR specification on which the implementation is based on.		
Multiplicity	1..1	**Type**	EcucIntegerParamDef
Range	0 - 255		
Default value	As per AUTOSAR version		
Post-build variant value	FALSE	**Post-build variant multiplicity**	-
Value configuration class	Published-Information	**Multiplicity configuration class**	-
Origin	IFX	**Scope**	LOCAL
Dependency	-		
Autosar Version	Applicable for Autosar versions 4.2.2 and 4.4.0.		

1.3.1.1.3   ArPatchVersion

Table 6	Specification for ArPatchVersion		
Name	ArPatchVersion		
Description	Patch version number of the AUTOSAR specification on which the implementation is based on.		
Multiplicity	1..1	**Type**	EcucIntegerParamDef
Range	0 - 255		
Default value	As per AUTOSAR version		

Table 6	Specification for ArPatchVersion (continued)		
Post-build variant value	FALSE	**Post-build variant multiplicity**	-
Value configuration class	Published-Information	**Multiplicity configuration class**	-
Origin	IFX	**Scope**	LOCAL
Dependency	-		
Autosar Version	Applicable for Autosar versions 4.2.2 and 4.4.0.		

1.3.1.1.4 ModuleId

Table 7	Specification for ModuleId		
Name	`ModuleId`		
Description	Module ID of the PORT driver.		
Multiplicity	1..1	**Type**	EcucIntegerParamDef
Range	0 - 65535		
Default value	124		
Post-build variant value	FALSE	**Post-build variant multiplicity**	-
Value configuration class	Published-Information	**Multiplicity configuration class**	-
Origin	IFX	**Scope**	LOCAL
Dependency	-		
Autosar Version	Applicable for Autosar versions 4.2.2 and 4.4.0.		

1.3.1.1.5 Release

Table 8	Specification for Release		
Name	`Release`		
Description	Specifies the derivate for which the configuration project is created.		
Multiplicity	1..1	**Type**	EcucStringParamDef
Range	String		
Default value	As per hardware derivative		
Post-build variant value	FALSE	**Post-build variant multiplicity**	-

Table 8   Specification for Release (continued)

Value configuration class	Published-Information	Multiplicity configuration class	-
Origin	IFX	**Scope**	LOCAL
Dependency	-		
Autosar Version	Applicable for Autosar versions 4.2.2 and 4.4.0.		

1.3.1.1.6   SwMajorVersion

Table 9   Specification for SwMajorVersion

Name	`SwMajorVersion`		
Description	Specifies the major version of the driver software.		
Multiplicity	1..1	**Type**	EcucIntegerParamDef
Range	0 - 255		
Default value	As per Driver		
Post-build variant value	FALSE	**Post-build variant multiplicity**	-
Value configuration class	Published-Information	**Multiplicity configuration class**	-
Origin	IFX	**Scope**	LOCAL
Dependency	-		
Autosar Version	Applicable for Autosar versions 4.2.2 and 4.4.0.		

1.3.1.1.7   SwMinorVersion

Table 10   Specification for SwMinorVersion

Name	`SwMinorVersion`		
Description	Specifies the minor version of the driver software.		
Multiplicity	1..1	**Type**	EcucIntegerParamDef
Range	0 - 255		
Default value	As per Driver		
Post-build variant value	FALSE	**Post-build variant multiplicity**	-
Value configuration class	Published-Information	**Multiplicity configuration class**	-
Origin	IFX	**Scope**	LOCAL

Table 10	Specification for SwMinorVersion (continued)		
Dependency	-		
Autosar Version	Applicable for Autosar versions 4.2.2 and 4.4.0.		

1.3.1.1.8 SwPatchVersion

Table 11	Specification for SwPatchVersion		
Name	SwPatchVersion		
Description	Specifies the patch version of the driver software.		
Multiplicity	1..1	**Type**	EcucIntegerParamDef
Range	0 - 255		
Default value	As per Driver		
Post-build variant value	FALSE	**Post-build variant multiplicity**	-
Value configuration class	Published-Information	**Multiplicity configuration class**	-
Origin	IFX	**Scope**	LOCAL
Dependency	-		
Autosar Version	Applicable for Autosar versions 4.2.2 and 4.4.0.		

1.3.1.1.9 VendorId

Table 12	Specification for VendorId		
Name	VendorId		
Description	Vendor ID for Infineon.		
Multiplicity	1..1	**Type**	EcucIntegerParamDef
Range	0 - 65535		
Default value	17		
Post-build variant value	FALSE	**Post-build variant multiplicity**	-
Value configuration class	Published-Information	**Multiplicity configuration class**	-
Origin	IFX	**Scope**	LOCAL
Dependency	-		
Autosar Version	Applicable for Autosar versions 4.2.2 and 4.4.0.		

1.3.1.2   Container: Port

This is the parent container for all configuration parameters of the PORT driver.

Post-Build Variant Multiplicity: -

Multiplicity Configuration Class: -

1.3.1.2.1   Config Variant

Table 13   Specification for Config Variant

Name	Config Variant		
Description	Selects the configuration variant for the PORT driver.		
Multiplicity	1..1	**Type**	EcucEnumerationParamDef
Range	VariantPostBuild: Post Build Support.		
Default value	VariantPostBuild		
Post-build variant value	FALSE	**Post-build variant multiplicity**	-
Value configuration class	Pre-Compile	**Multiplicity configuration class**	-
Origin	IFX	**Scope**	LOCAL
Dependency	-		
Autosar Version	Applicable for Autosar versions 4.2.2 and 4.4.0.		

1.3.1.3   Container: PortConfigSet

This container contains the configuration parameters and sub containers of the PORT driver.

Post-Build Variant Multiplicity: -
Multiplicity Configuration Class: -

1.3.1.4   Container: PortContainer

This container holds the configuration parameters related to all the port pins.
Post-Build Variant Multiplicity: -
Multiplicity Configuration Class: -

1.3.1.4.1   PortNumber

Table 14   Specification for PortNumber

Name	PortNumber
Description	Specifies the port number currently being configured.
	Note: The values configured here are device dependent and retrieved from device property files. It cannot be edited by the user.

Table 14 **Specification for PortNumber (continued)**

Multiplicity	1..1	Type	EcucIntegerParamDef
Range	0 - max port available		
Default value	Device dependent. Starting value is retrieved from device property files.		
Post-build variant value	FALSE	Post-build variant multiplicity	-
Value configuration class	Pre-Compile	Multiplicity configuration class	-
Origin	IFX	Scope	LOCAL
Dependency	-		
Autosar Version	Applicable for Autosar versions 4.2.2 and 4.4.0.		

1.3.1.4.2 PortNumberOfPortPins

Table 15 **Specification for PortNumberOfPortPins**

Name	PortNumberOfPortPins		
Description	Specifies the number of port pins available for the selected port. *Note: The values configured here are device dependent and retrieved from device property files. It cannot be edited by the user.*		
Multiplicity	1..1	Type	EcucIntegerParamDef
Range	1 - 16		
Default value	Device dependent. Starting value is retrieved from device property files.		
Post-build variant value	FALSE	Post-build variant multiplicity	-
Value configuration class	Pre-Compile	Multiplicity configuration class	-
Origin	AUTOSAR_ECUC	Scope	LOCAL
Dependency	-		
Autosar Version	Applicable for Autosar versions 4.2.2 and 4.4.0.		

1.3.1.4.3 PortPinEcucPartitionRef

Table 16 **Specification for PortPinEcucPartitionRef**

Name	PortPinEcucPartitionRef		
Description	Parameter support is added only for AUTOSAR schema compliance, this parameter is not used in code generation logic, and hence this parameter is made editable false.		
Multiplicity	1..1	Type	EcucReferenceDef

Table 16   Specification for PortPinEcucPartitionRef (continued)

Range	Reference to Node: PortPinEcucPartitionRef		
Default value	NULL		
Post-build variant value	FALSE	Post-build variant multiplicity	-
Value configuration class	Pre-Compile	Multiplicity configuration class	-
Origin	AUTOSAR_ECUC	Scope	ECU
Dependency	-		
Autosar Version	Applicable for Autosar version 4.4.0.		

1.3.1.5   Container: PortGeneral

The container holds the driver's configuration parameters.

Post-Build Variant Multiplicity: -

Multiplicity Configuration Class: -

1.3.1.5.1   PortDevErrorDetect

Table 17   Specification for PortDevErrorDetect

Name	`PortDevErrorDetect`		
Description	Enables / Disables the detection and reporting of Development Error Detection. TRUE: DET is enabled. FALSE: DET is disabled.		
Multiplicity	1..1	Type	EcucBooleanParamDef
Range	TRUE FALSE		
Default value	FALSE		
Post-build variant value	FALSE	Post-build variant multiplicity	-
Value configuration class	Pre-Compile	Multiplicity configuration class	-
Origin	AUTOSAR_ECUC	Scope	LOCAL
Dependency	-		
Autosar Version	Applicable for Autosar versions 4.2.2 and 4.4.0.		

1.3.1.5.2 PortEcucPartitionRef

Table 18   Specification for PortEcucPartitionRef

Name	PortEcucPartitionRef		
Description	Parameter support is added only for AUTOSAR schema compliance, this parameter is not used in code generation logic, and hence this parameter is made editable false.		
Multiplicity	1..1	Type	EcucReferenceDef
Range	Reference to Node: EcucPartition, PortPinEcucPartitionRef		
Default value	NULL		
Post-build variant value	None	Post-build variant multiplicity	-
Value configuration class	None	Multiplicity configuration class	-
Origin	AUTOSAR_ECUC	Scope	LOCAL
Dependency	-		
Autosar Version	Applicable for Autosar version 4.4.0.		

1.3.1.5.3 PortInitApiMode

Table 19   Specification for PortInitApiMode

Name	PortInitApiMode		
Description	Specifies the privilege mode in which the initialization API will operate. *Note: The driver accesses the SFRs, it is more efficient to operate the PORT driver in supervisor mode. Hence, the default mode of operation is supervisor.*		
Multiplicity	1..1	Type	EcucEnumerationParamDef
Range	PORT_MCAL_SUPERVISOR: The selected privilege mode is SUPERVISOR PORT_MCAL_USER1: The selected privilege mode is USER1		
Default value	PORT_MCAL_SUPERVISOR		
Post-build variant value	FALSE	Post-build variant multiplicity	-
Value configuration class	Pre-Compile	Multiplicity configuration class	-
Origin	IFX	Scope	LOCAL
Dependency	-		
Autosar Version	Applicable for Autosar versions 4.2.2 and 4.4.0.		

1.3.1.5.4 PortInitCheckApi

Table 20 **Specification for PortInitCheckApi**

Name	`PortInitCheckApi`		
Description	Enables/disables the Port_InitCheck API. TRUE: PortInitCheckApi API is enabled. FALSE: PortInitCheckApi API is disabled. *Note: The detection of safety related errors is enabled by default to ensure that safety issues are addressed during the product lifecycle.*		
Multiplicity	1..1	**Type**	EcucBooleanParamDef
Range	TRUE FALSE		
Default value	FALSE		
Post-build variant value	FALSE	**Post-build variant multiplicity**	-
Value configuration class	Pre-Compile	**Multiplicity configuration class**	-
Origin	IFX	**Scope**	LOCAL
Dependency	-		
Autosar Version	Applicable for Autosar versions 4.2.2 and 4.4.0.		

1.3.1.5.5 PortSafetyEnable

Table 21 **Specification for PortSafetyEnable**

Name	`PortSafetyEnable`		
Description	Specifies whether the safety checks mandated by safety standards are enabled or disabled. TRUE: Safety checks are enabled. FALSE: Safety checks are disabled. *Note: The detection of safety related errors is enabled by default to ensure that safety issues are addressed during the product lifecycle.*		
Multiplicity	1..1	**Type**	EcucBooleanParamDef
Range	TRUE FALSE		
Default value	TRUE		
Post-build variant value	FALSE	**Post-build variant multiplicity**	-

Table 21 **Specification for PortSafetyEnable (continued)**

Value configuration class	Pre-Compile	Multiplicity configuration class	-
Origin	IFX	**Scope**	LOCAL
Dependency	-		
Autosar Version	Applicable for Autosar versions 4.2.2 and 4.4.0.		

1.3.1.5.6 PortSetPinDirectionApi

Table 22 **Specification for PortSetPinDirectionApi**

Name	`PortSetPinDirectionApi`		
Description	Pre-processor switch for enabling the API Port_SetPinDirection () which sets the port pin direction of the referenced pin during runtime. TRUE: Port_SetPinDirection API is available. FALSE: Port_SetPinDirection API is not available. *Note: The APIs is disabled by default to minimize the executable code size.*		
Multiplicity	1..1	**Type**	EcucBooleanParamDef
Range	TRUE FALSE		
Default value	FALSE		
Post-build variant value	FALSE	**Post-build variant multiplicity**	-
Value configuration class	Pre-Compile	**Multiplicity configuration class**	-
Origin	AUTOSAR_ECUC	**Scope**	LOCAL
Dependency	-		
Autosar Version	Applicable for Autosar versions 4.2.2 and 4.4.0.		

1.3.1.5.7 PortSetPinModeApi

Table 23 **Specification for PortSetPinModeApi**

Name	`PortSetPinModeApi`
Description	Pre-processor switch to enable/disable the use of the API Port_SetPinMode(). TRUE: Port_SetPinMode() API is available. FALSE: Port_SetPinMode() API is not available. *Note: The API is disabled by default to minimize the executable code size.*

Table 23   Specification for PortSetPinModeApi (continued)

Multiplicity	1..1	**Type**	EcucBooleanParamDef
Range	TRUE FALSE		
Default value	FALSE		
Post-build variant value	FALSE	**Post-build variant multiplicity**	-
Value configuration class	Pre-Compile	**Multiplicity configuration class**	-
Origin	AUTOSAR_ECUC	**Scope**	LOCAL
Dependency	-		
Autosar Version	Applicable for Autosar versions 4.2.2 and 4.4.0.		

1.3.1.5.8   PortVersionInfoApi

Table 24   Specification for PortVersionInfoApi

Name	`PortVersionInfoApi`
Description	Pre-processor switch to enable / disable the API to read out the driver version information. TRUE: Port_GetVersionInfo API is enabled. FALSE: Port_GetVersionInfo API is disabled. *Note: The APIs is disabled by default to minimize the executable code size.*

Multiplicity	1..1	**Type**	EcucBooleanParamDef
Range	TRUE FALSE		
Default value	FALSE		
Post-build variant value	FALSE	**Post-build variant multiplicity**	-
Value configuration class	Pre-Compile	**Multiplicity configuration class**	-
Origin	AUTOSAR_ECUC	**Scope**	LOCAL
Dependency	-		
Autosar Version	Applicable for Autosar versions 4.2.2 and 4.4.0.		

1.3.1.6   Container: PortLVDS

This container holds all the configuration parameters for LVDS port pin pairs.

Post-Build Variant Multiplicity: -

Multiplicity Configuration Class: -

1.3.1.6.1 PortLVDSMode

Table 25   Specification for PortLVDSMode

Name	PortLVDSMode		
Description	Specifies the frequency mode for the Rx pads. *Note: The default value of this parameter is set to the reset value of the corresponding SFR.*		
Multiplicity	1..1	Type	EcucEnumerationParamDef
Range	LVDSH: High frequency mode is selected LVDSM: Reduced frequency mode is selected		
Default value	LVDSH		
Post-build variant value	TRUE	Post-build variant multiplicity	-
Value configuration class	Post-Build	Multiplicity configuration class	-
Origin	IFX	Scope	LOCAL
Dependency	-		
Autosar Version	Applicable for Autosar versions 4.2.2 and 4.4.0.		

1.3.1.6.2 PortLVDSPadSupply

Table 26   Specification for PortLVDSPadSupply

Name	PortLVDSPadSupply		
Description	Specifies the supply voltage for both Tx and Rx pads. *Note: The default value of this parameter is set to the reset value of the corresponding SFR.*		
Multiplicity	1..1	Type	EcucEnumerationParamDef
Range	V3_3: 3.3 V is selected V5_0: 5.0 V is selected		
Default value	V3_3		
Post-build variant value	TRUE	Post-build variant multiplicity	-
Value configuration class	Post-Build	Multiplicity configuration class	-
Origin	IFX	Scope	LOCAL

1 Port driver

Table 26   **Specification for PortLVDSPadSupply (continued)**

Dependency	-
Autosar Version	Applicable for Autosar versions 4.2.2 and 4.4.0.

1.3.1.6.3   PortLVDSPinPair

Table 27   **Specification for PortLVDSPinPair**

Name	`PortLVDSPinPair`		
Description	Publishes the port pin pair which supports the LVDS feature. The value is automatically retrieved by the tool from the device property files. *Note: This parameter cannot be edited by the user.*		
Multiplicity	1..1	**Type**	EcucStringParamDef
Range	String		
Default value	Pin_x_Pin_y [x and y indicate the pin pair for LVDS]		
Post-build variant value	FALSE	**Post-build variant multiplicity**	-
Value configuration class	Post-Build	**Multiplicity configuration class**	-
Origin	IFX	**Scope**	LOCAL
Dependency	-		
Autosar Version	Applicable for Autosar versions 4.2.2 and 4.4.0.		

1.3.1.6.4   PortLVDSRxEnController

Table 28   **Specification for PortLVDSRxEnController**

Name	`PortLVDSRxEnController`		
Description	Specifies the controller of the LVDS enable/disable function. It is applicable only for the Rx LVDS pair. *Note: The default value of this parameter is set to the reset value of the corresponding SFR.*		
Multiplicity	1..1	**Type**	EcucEnumerationParamDef
Range	HSCT_CONTROLLED: If HSCT_CONTROLLED is selected as the controller, the user must enable/disable the LVDS in registers related to HSCT module. PORT_CONTROLLED: If PORT_CONTROLLED is selected, then LVDS enable/disable is performed by the means of PORT registers.		
Default value	PORT_CONTROLLED		
Post-build variant value	TRUE	**Post-build variant multiplicity**	-

1 Port driver

Table 28	Specification for PortLVDSRxEnController (continued)		
Value configuration class	Post-Build	**Multiplicity configuration class**	-
Origin	IFX	**Scope**	LOCAL
Dependency	-		
Autosar Version	Applicable for Autosar versions 4.2.2 and 4.4.0.		

1.3.1.6.5   PortLVDSRxPathEnable

Table 29	Specification for PortLVDSRxPathEnable		
Name	`PortLVDSRxPathEnable`		
Description	Specifies whether the pin pair is in LVDS mode or CMOS mode. TRUE: Enables the LVDS transceiver and disables the CMOS mode FALSE: Disables the LVDS transceiver and enables the CMOS mode The parameter is applicable for only Rx LVDS pair and when PortLVDSRxEnController is holding a value PORT_CONTROLLED. *Note: The default value of this parameter is set to the reset value of the corresponding SFR.*		
Multiplicity	1..1	**Type**	EcucBooleanParamDef
Range	TRUE FALSE		
Default value	FALSE		
Post-build variant value	TRUE	**Post-build variant multiplicity**	-
Value configuration class	Post-Build	**Multiplicity configuration class**	-
Origin	IFX	**Scope**	LOCAL
Dependency	-		
Autosar Version	Applicable for Autosar versions 4.2.2 and 4.4.0.		

1.3.1.6.6   PortLVDSRxTerminationMode

Table 30	Specification for PortLVDSRxTerminationMode		
Name	`PortLVDSRxTerminationMode`		
Description	Selects the terminal load resistor for the port pin. This parameter is applicable only for the Rx LVDS pair. *Note: The default value of this parameter is set to the reset value of the corresponding SFR.*		

Table 30   Specification for PortLVDSRxTerminationMode (continued)

Multiplicity	1..1	Type	EcucEnumerationParamDef
Range	EXTERNAL_TERMINATION: External load resistor is selected (external termination on the PCB). INTERNAL_TERMINATION: Internal load resistor of a 100 ohms is selected.		
Default value	INTERNAL_TERMINATION		
Post-build variant value	TRUE	**Post-build variant multiplicity**	-
Value configuration class	Post-Build	**Multiplicity configuration class**	-
Origin	IFX	**Scope**	LOCAL
Dependency	-		
Autosar Version	Applicable for Autosar versions 4.2.2 and 4.4.0.		

1.3.1.6.7   PortLVDSTxEnController

Table 31   Specification for PortLVDSTxEnController

Name	`PortLVDSTxEnController`		
Description	Specifies the controller of the LVDS enable/disable function. It is applicable only for Tx LVDS pair. *Note: The default value of this parameter is set to the reset value of the corresponding SFR.*		
Multiplicity	1..1	**Type**	EcucEnumerationParamDef
Range	HSCT_CONTROLLED: HSCT_CONTROLLED is selected as the controller, the user should take care of enabling/disabling the LVDS in registers related to HSCT module. PORT_CONTROLLED: PORT_CONTROLLED is selected, then LVDS enable/disable can be done by means of PORT registers.		
Default value	PORT_CONTROLLED		
Post-build variant value	TRUE	**Post-build variant multiplicity**	-
Value configuration class	Post-Build	**Multiplicity configuration class**	-
Origin	IFX	**Scope**	LOCAL
Dependency	-		
Autosar Version	Applicable for Autosar versions 4.2.2 and 4.4.0.		

1.3.1.6.8　PortLVDSTxPathEnable

Table 32　　　**Specification for PortLVDSTxPathEnable**

Name	PortLVDSTxPathEnable		
Description	Specifies whether the pin pair is in LVDS mode or CMOS mode. TRUE: Enables the LVDS transceiver FALSE: Disable the LVDS transceiver The parameter is applicable for TX LVDS pair and only when PortLVDSTxEnController is holding a value PORT_CONTROLLED. *Note: The default value of this parameter is set to the reset value of the corresponding SFR.*		
Multiplicity	1..1	Type	EcucBooleanParamDef
Range	TRUE FALSE		
Default value	FALSE		
Post-build variant value	TRUE	Post-build variant multiplicity	-
Value configuration class	Post-Build	Multiplicity configuration class	-
Origin	IFX	Scope	LOCAL
Dependency	-		
Autosar Version	Applicable for Autosar versions 4.2.2 and 4.4.0.		

1.3.1.6.9　PortLVDSTxPowerDownPullDown

Table 33　　　**Specification for PortLVDSTxPowerDownPullDown**

Name	PortLVDSTxPowerDownPullDown		
Description	Specifies the state of the Transmit Pull down resistor. DISABLE: Disables the Tx Power down - pull down resistor ENABLE: Enables the Tx Power down - pull down resistor This parameter is applicable only for the transmitting LVDS pair. *Note: The default value of this parameter is set to the reset value of the corresponding SFR.*		
Multiplicity	1..1	Type	EcucEnumerationParamDef
Range	DISABLE: Disables the Tx Power down - pull down resistor ENABLE: Enables the Tx Power down - pull down resistor		
Default value	DISABLE		
Post-build variant value	TRUE	Post-build variant multiplicity	-

Table 33	**Specification for PortLVDSTxPowerDownPullDown (continued)**		
Value configuration class	Post-Build	**Multiplicity configuration class**	-
Origin	IFX	**Scope**	LOCAL
Dependency	-		
Autosar Version	Applicable for Autosar versions 4.2.2 and 4.4.0.		

1.3.1.7 Container: PortPin

Configuration of the individual port pins.

Post-Build Variant Multiplicity: -

Multiplicity Configuration Class: -

1.3.1.7.1 PortPinControllerSelect

Table 34	**Specification for PortPinControllerSelect**		
Name	`PortPinControllerSelect`		
Description	The parameter enables / disables whether SCR controls the port pins configuration and data. *Note: The default value of this parameter is set to the reset value of the corresponding SFR.*		
Multiplicity	1..1	**Type**	EcucEnumerationParamDef
Range	DISABLE: Tricore is selected for data and control of the pin. ENABLE: SCR/VADC/GETH/SMU selected for data and control of the pin.		
Default value	DISABLE		
Post-build variant value	TRUE	**Post-build variant multiplicity**	-
Value configuration class	Post-Build	**Multiplicity configuration class**	-
Origin	IFX	**Scope**	LOCAL
Dependency	PortNumber		
Autosar Version	Applicable for Autosar versions 4.2.2 and 4.4.0.		

1.3.1.7.2 PortPinDirection

Table 35	**Specification for PortPinDirection**	
Name	`PortPinDirection`	
Description	Specifies the direction for the port pin.	

Table 35   **Specification for PortPinDirection (continued)**

	Note: The configuration of this parameter is not valid for analog input ports and the default value is set to the reset value of the corresponding SFR.		
Multiplicity	1..1	**Type**	EcucEnumerationParamDef
Range	PORT_PIN_IN: Port Pin direction set as input PORT_PIN_OUT: Port Pin direction set as output		
Default value	PORT_PIN_IN		
Post-build variant value	TRUE	**Post-build variant multiplicity**	-
Value configuration class	Post-Build	**Multiplicity configuration class**	-
Origin	AUTOSAR_ECUC	**Scope**	LOCAL
Dependency	PortNumber		
Autosar Version	Applicable for Autosar versions 4.2.2 and 4.4.0.		

1.3.1.7.3   PortPinDirectionChangeable

Table 36   **Specification for PortPinDirectionChangeable**

Name	PortPinDirectionChangeable		
Description	Specifies whether the pin direction can be changed at runtime for the current port pin. The configuration of this parameter is not valid for analog or digital input ports. *Note: The optional features are disabled by default to minimize the executable code size.*		
Multiplicity	1..1	**Type**	EcucBooleanParamDef
Range	TRUE FALSE		
Default value	FALSE		
Post-build variant value	TRUE	**Post-build variant multiplicity**	-
Value configuration class	Post-Build	**Multiplicity configuration class**	-
Origin	AUTOSAR_ECUC	**Scope**	LOCAL
Dependency	PortNumber, PortSetPinDirectionApi		
Autosar Version	Applicable for Autosar versions 4.2.2 and 4.4.0.		

1.3.1.7.4   PortPinEmergencyStop

Table 37   Specification for PortPinEmergencyStop

Name	`PortPinEmergencyStop`		
Description	With this feature enabled, a pin configured as output pin will be automatically be reconfigured as an input, when an emergency condition is encountered. FALSE: Disable emergency stop function for the selected pin TRUE: Enables emergency stop function for the selected pin *Note: The default value of this parameter is set to the reset value of the corresponding SFR.*		
Multiplicity	1..1	Type	EcucBooleanParamDef
Range	TRUE FALSE		
Default value	FALSE		
Post-build variant value	TRUE	Post-build variant multiplicity	-
Value configuration class	Post-Build	Multiplicity configuration class	-
Origin	IFX	Scope	LOCAL
Dependency	PortPinDirection		
Autosar Version	Applicable for Autosar versions 4.2.2 and 4.4.0.		

1.3.1.7.5   PortPinEnableAnalogInputOnly

Table 38   Specification for PortPinEnableAnalogInputOnly

Name	`PortPinEnableAnalogInputOnly`		
Description	Certain digital PORT pads additionally support analog input functionality. It is possible to completely disable the digital functions of the pad in order to use the pad for analog input. *Note: The default value of this parameter is set to the reset value of the corresponding SFR.*		
Multiplicity	1..1	Type	EcucEnumerationParamDef
Range	PORT_PIN_ANALOG_INPUT_DISABLE: Analog Input functionality is disabled PORT_PIN_ANALOG_INPUT_ENABLE: Analog Input functionality is enabled.		
Default value	PORT_PIN_ANALOG_INPUT_DISABLE		
Post-build variant value	TRUE	Post-build variant multiplicity	-
Value configuration class	Post-Build	Multiplicity configuration class	-
Origin	IFX	Scope	LOCAL

1 Port driver

Table 38	Specification for PortPinEnableAnalogInputOnly (continued)
Dependency	PortNumber
Autosar Version	Applicable for Autosar versions 4.2.2 and 4.4.0.

1.3.1.7.6   PortPinId

Table 39	Specification for PortPinId		
Name	`PortPinId`		
Description	Publish an Id for each port pins. *Note: The default value of this parameter is to be retained. This read only parameter is derived from property files.*		
Multiplicity	1..1	**Type**	EcucIntegerParamDef
Range	0x0000 - max port pin ID		
Default value	Device Dependent. Starting value is retrieved from device property files.		
Post-build variant value	FALSE	**Post-build variant multiplicity**	-
Value configuration class	Pre-Compile	**Multiplicity configuration class**	-
Origin	AUTOSAR_ECUC	**Scope**	LOCAL
Dependency	-		
Autosar Version	Applicable for Autosar versions 4.2.2 and 4.4.0.		

1.3.1.7.7   PortPinInitialMode

Table 40	Specification for PortPinInitialMode		
Name	`PortPinInitialMode`		
Description	The parameter allows to configure the operating mode i.e. the different alternate functionality of each pin. The selected operating mode is programmed to registers during initialization. *Note: This parameter is editable only if the port pin direction is set to PORT_PIN_OUT and the default value is set to the reset value of the corresponding SFR. The tooltip for PortPinIntialMode gives the list of all available ALT mode.*		
Multiplicity	1..1	**Type**	EcucEnumerationParamDef
Range	ALT1: ALT1 mode ALT2: ALT2 mode ALT3: ALT3 mode ALT4: ALT4 mode		

1 Port driver

Table 40	Specification for PortPinInitialMode (continued)		
	ALT5: ALT5 mode ALT6: ALT6 mode ALT7: ALT7 mode GPIO: GPIO mode		
Default value	GPIO		
Post-build variant value	TRUE	**Post-build variant multiplicity**	-
Value configuration class	Post-Build	**Multiplicity configuration class**	-
Origin	AUTOSAR_ECUC	**Scope**	LOCAL
Dependency	PortPinDirection		
Autosar Version	Applicable for Autosar versions 4.2.2 and 4.4.0.		

1.3.1.7.8 PortPinInputPadLevel

Table 41	Specification for PortPinInputPadLevel		
Name	`PortPinInputPadLevel`		
Description	The parameter allows to configure the voltage level for the selected port pin. *Note: The default value of this parameter is set to the reset value of the corresponding SFR.*		
Multiplicity	1..1	**Type**	EcucEnumerationParamDef
Range	PORT _INPUT_LEVEL_TTL_3_3V: TTL level for 3.3V PORT _INPUT_LEVEL_TTL_5_0V: TTL level for 5V. PORT_INPUT_LEVEL_CMOS_AUTOMOTIVE: CMOS automotive level PORT_RGMII_INPUT: Selection only for pads with RGMII input buffer for GETH		
Default value	PORT_INPUT_LEVEL_CMOS_AUTOMOTIVE		
Post-build variant value	TRUE	**Post-build variant multiplicity**	-
Value configuration class	Post-Build	**Multiplicity configuration class**	-
Origin	IFX	**Scope**	LOCAL
Dependency	PortPinDirection		
Autosar Version	Applicable for Autosar versions 4.2.2 and 4.4.0.		

1.3.1.7.9   PortPinInputPullResistor

Table 42   **Specification for PortPinInputPullResistor**

Name	`PortPinInputPullResistor`		
Description	The parameter allows to configure the internal Pull resistor [up/down] for the selected port pin. *Note: The default value of this parameter is set to the reset value of the corresponding SFR.*		
Multiplicity	1..1	Type	EcucEnumerationParamDef
Range	PORT_PIN_IN_NO_PULL: Input pull is not connected and pin operates in tristate mode. PORT_PIN_IN_PULL_DOWN: Pull-down is connected. PORT_PIN_IN_PULL_UP: Pull-up is connected.		
Default value	PORT_PIN_IN_PULL_UP		
Post-build variant value	TRUE	Post-build variant multiplicity	-
Value configuration class	Post-Build	Multiplicity configuration class	-
Origin	IFX	Scope	LOCAL
Dependency	PortPinDirectionChangeable, PortPinDirection		
Autosar Version	Applicable for Autosar versions 4.2.2 and 4.4.0.		

1.3.1.7.10   PortPinLevelValue

Table 43   **Specification for PortPinLevelValue**

Name	`PortPinLevelValue`		
Description	The parameter is to configure the initial pin level for each pin. This level is set to the output latch of the port pin during initialization of the port driver, irrespective of the direction configured to the pin. *Note: For analog or digital input ports, the configuration of this parameter is not valid and the default value is set to the reset value of the corresponding SFR.*		
Multiplicity	1..1	Type	EcucEnumerationParamDef
Range	PORT_PIN_LEVEL_HIGH: Port Pin level is High. PORT_PIN_LEVEL_LOW: Port Pin level is LOW.		
Default value	PORT_PIN_LEVEL_LOW		
Post-build variant value	TRUE	Post-build variant multiplicity	-
Value configuration class	Post-Build	Multiplicity configuration class	-

Table 43	Specification for PortPinLevelValue (continued)				
Origin	AUTOSAR_ECUC	**Scope**		LOCAL	
Dependency	PortPinDirection				
Autosar Version	Applicable for Autosar versions 4.2.2 and 4.4.0.				

1.3.1.7.11 PortPinMode

Table 44	Specification for PortPinMode		
Name	`PortPinMode`		
Description	The parameter allows to configure all the possible alternate modes for the current port pin. *Note: Refer to PortPinIntialMode tooltip for the ALT modes available. The default value is PORT_PIN_MODE_ALL which are the bits set according to the modes that are available for that port pin.*		
Multiplicity	1..8	**Type**	EcucEnumerationParamDef
Range	PORT_PIN_MODE_ALL: All the alternate modes are supported. PORT_PIN_MODE_ALT1: Alternate mode 1 is supported. PORT_PIN_MODE_ALT2: Alternate mode 2 is supported. PORT_PIN_MODE_ALT3: Alternate mode 3 is supported. PORT_PIN_MODE_ALT4: Alternate mode 4 is supported. PORT_PIN_MODE_ALT5: Alternate mode 5 is supported. PORT_PIN_MODE_ALT6: Alternate mode 6 is supported. PORT_PIN_MODE_ALT7: Alternate mode 7 is supported. PORT_PIN_MODE_GPIO: Alternate mode GPIO is supported.		
Default value	PORT_PIN_MODE_ALL		
Post-build variant value	TRUE	**Post-build variant multiplicity**	TRUE
Value configuration class	Post-Build	**Multiplicity configuration class**	Post-Build
Origin	AUTOSAR_ECUC	**Scope**	LOCAL
Dependency	PortPinModeChangeable, PortPinDirection, PortSetPinModeApi		
Autosar Version	Applicable for Autosar versions 4.2.2 and 4.4.0.		

1.3.1.7.12 PortPinModeChangeable

Table 45	Specification for PortPinModeChangeable
Name	`PortPinModeChangeable`

1 Port driver

Table 45	Specification for PortPinModeChangeable (continued)		
Description	The parameter allows to configure whether the mode for a port pin is allowed to be changed at run-time. The configuration of this parameter is not valid for analog or digital input ports. FALSE: Mode is not changeable at runtime for the selected pin. TRUE: Mode is changeable at runtime for the selected pin. *Note: The optional features are disabled by default to minimize the executable code size.*		
Multiplicity	1..1	Type	EcucBooleanParamDef
Range	TRUE FALSE		
Default value	FALSE		
Post-build variant value	TRUE	Post-build variant multiplicity	-
Value configuration class	Post-Build	Multiplicity configuration class	-
Origin	AUTOSAR_ECUC	Scope	LOCAL
Dependency	PortPinDirectionChangeable, PortNumber, PortSetPinModeApi		
Autosar Version	Applicable for Autosar versions 4.2.2 and 4.4.0.		

1.3.1.7.13 PortPinOutputPadDriveStrength

Table 46	Specification for PortPinOutputPadDriveStrength		
Name	`PortPinOutputPadDriveStrength`		
Description	This parameter configures the output drive strength and slew rate for each pin. *Note: The default value of this parameter is set to the reset value of the corresponding SFR.*		
Multiplicity	1..1	Type	EcucEnumerationParamDef
Range	PORT_PIN_DEFAULT_DRIVER: Default driver strength. PORT_PIN_MEDIUM_DRIVER: Medium driver strength applicable for RFast, Fast and Slow pads PORT_PIN_MEDIUM_DRIVER_SHARP_EDGE: Medium driver strength with sharp edge applicable only for slow pads. PORT_PIN_RGMII_DRIVER: RGMII driver applicable only for RFast pads PORT_PIN_STRONG_DRIVER_MEDIUM_EDGE: This option is selectable for RFast and Fast pads. PORT_PIN_STRONG_DRIVER_SHARP_EDGE: Strong driver strength with sharp edge applicable for RFast and Fast pads.		
Default value	PORT_PIN_DEFAULT_DRIVER		
Post-build variant value	TRUE	Post-build variant multiplicity	-

Table 46   Specification for PortPinOutputPadDriveStrength (continued)

Value configuration class	Post-Build	Multiplicity configuration class	-
Origin	IFX	**Scope**	LOCAL
Dependency	PortPinDirectionChangeable, PortPinDirection		
Autosar Version	Applicable for Autosar versions 4.2.2 and 4.4.0.		

1.3.1.7.14   PortPinOutputPinDriveMode

Table 47   Specification for PortPinOutputPinDriveMode

Name	`PortPinOutputPinDriveMode`		
Description	This parameter allows to configure between open-drain or push pull. *Note: The default value of this parameter is set to the reset value of the corresponding SFR.*		
Multiplicity	1..1	**Type**	EcucEnumerationParamDef
Range	PORT_PIN_OUT_OPENDRAIN: PORT_PIN_OUT_OPENDRAIN: Open drain configuration is selected for the pin. PORT_PIN_OUT_PUSHPULL: PORT_PIN_OUT_PUSHPULL: Push pull configuration is selected for the pin.		
Default value	PORT_PIN_OUT_PUSHPULL		
Post-build variant value	TRUE	**Post-build variant multiplicity**	-
Value configuration class	Post-Build	**Multiplicity configuration class**	-
Origin	IFX	**Scope**	LOCAL
Dependency	PortPinDirectionChangeable, PortPinDirection		
Autosar Version	Applicable for Autosar versions 4.2.2 and 4.4.0.		

1.3.1.7.15   PortPinSymbolicName

Table 48   Specification for PortPinSymbolicName

Name	`PortPinSymbolicName`		
Description	This parameter is a user defined name for the port pin under consideration. The user of the PORT driver can use the enumerator to identify a Port pin pair rather than using an absolute number.		
Multiplicity	1..1	**Type**	EcucStringParamDef
Range	String		

Table 48   Specification for PortPinSymbolicName (continued)

Default value	PORT_x_PIN_Y		
Post-build variant value	FALSE	**Post-build variant multiplicity**	-
Value configuration class	Pre-Compile	**Multiplicity configuration class**	-
Origin	IFX	**Scope**	LOCAL
Dependency	PortPinId		
Autosar Version	Applicable for Autosar versions 4.2.2 and 4.4.0.		

1.3.2   Functions - Type definitions

This section lists all the data type of the PORT driver.

1.3.2.1   Port_ConfigType

Table 49   Specification for Port_ConfigType

Syntax	`Port_ConfigType`	
Type	Structure	
File	`Port.h`	
Description	Data type for the post-build configuration structure	
Source	AUTOSAR	
Autosar Version	Applicable for Autosar versions 4.2.2 and 4.4.0.	

1.3.2.2   Port_PinDirectionType

Table 50   Specification for Port_PinDirectionType

Syntax	`Port_PinDirectionType`	
Type	Enumeration	
File	`Port.h`	
Range	0 - PORT_PIN_IN	Sets port pin as input.(0x00)
	128 - PORT_PIN_OUT	Sets port pin as output.(0x80)
Description	The type defines PORT pin direction	
Source	AUTOSAR	
Autosar Version	Applicable for Autosar versions 4.2.2 and 4.4.0.	

1.3.2.3   Port_PinModeType

Table 51   Specification for Port_PinModeType

Syntax	`Port_PinModeType`	
Type	uint8	
File	`Port.h`	
Range	PORT_INPUT_NO_PULL	To support input characteristic no input pull.
	PORT_INPUT_PULL_DOWN	To support input characteristic pull-down.
	PORT_INPUT_PULL_UP	To support input characteristic pull-up.
	PORT_OUTPUT_OPEN_DRAIN	To support output characteristic open drain.
	PORT_OUTPUT_PUSH_PULL	To support output characteristic push pull.
	PORT_PIN_MODE_ALT1	Alternate mode 1
	PORT_PIN_MODE_ALT2	Alternate mode 2.
	PORT_PIN_MODE_ALT3	Alternate mode 3.
	PORT_PIN_MODE_ALT4	Alternate mode 4.
	PORT_PIN_MODE_ALT5	Alternate mode 5.
	PORT_PIN_MODE_ALT6	Alternate mode 6.
	PORT_PIN_MODE_ALT7	Alternate mode 7.
	PORT_PIN_MODE_GPIO	GPIO mode.
Description	The type defines PORT pin mode and/or input/output pin characteristics. User can pass the pin mode along with the required pin characteristics (Using Bitwise OR). User can also pass only the pin mode or the pin characteristics as required.	
Source	AUTOSAR	
Autosar Version	Applicable for Autosar versions 4.2.2 and 4.4.0.	

1.3.2.4   Port_PinType

Table 52   Specification for Port_PinType

Syntax	`Port_PinType`	
Type	uint16	
File	`Port.h`	
Range	0-Number of available port pins	
Description	This type defines numeric ID for port pins.	
Source	AUTOSAR	
Autosar Version	Applicable for Autosar versions 4.2.2 and 4.4.0.	

1.3.3 Functions - APIs

This section lists all the APIs of the PORT driver.

1.3.3.1 Port_GetVersionInfo

Table 53 **Specification for** `Port_GetVersionInfo` **API**

Syntax	`void Port_GetVersionInfo`
	`(`
	`    Std_VersionInfoType * const versioninfo`
	`)`	

Service ID	0x03	
Sync/Async	Synchronous	
ASIL Level	B	
Re-entrancy	Reentrant	
Parameters (in)	-	-
Parameters (out)	versioninfo	Pointer to where to store the version information of the driver.
Parameters (in - out)	-	-
Return	void	-
Description	Returns the version information of PORT driver.	
Source	AUTOSAR	
Error handling	PORT_E_PARAM_POINTER	
Configuration dependencies	PortVersionInfoApi	
User hints	-	
SFR accessed	-	
Autosar Version	Applicable for Autosar versions 4.2.2 and 4.4.0.	

1.3.3.2 Port_Init

Table 54 **Specification for** `Port_Init` **API**

Syntax	`void Port_Init`
	`(`
	`    const Port_ConfigType * const ConfigPtr`
	`)`

Service ID	0x00
Sync/Async	Synchronous

Table 54	Specification for `Port_Init` API (continued)	
ASIL Level	B	
Re-entrancy	Non Reentrant	
Parameters (in)	ConfigPtr	Pointer to configuration set.
Parameters (out)	-	-
Parameters (in - out)	-	-
Return	void	-
Description	Initializes the PORT driver as per the configuration set passed.	
Source	AUTOSAR	
Error handling	PORT_E_INIT_FAILED	
Configuration dependencies	-	
User hints	-	
SFR accessed	CPU_COMPAT(w), CPU_SYSCON(w), CPU_TPS_EXTIM_CLASS_EN(w), CPU_TPS_EXTIM_ENTRY_LVAL(w), CPU_TPS_EXTIM_EXIT_LVAL(w), P_ESR(w), P_IOCR0(w), P_IOCR12(w), P_IOCR4(w), P_IOCR8(w), P_LPCR(w), P_OUT(w), P_PCSR(w), P_PDISC(w), P_PDR0(w), P_PDR1(w), SCU_CCUCON0(r), SCU_EICON0(rw), SCU_OSCCON(r), SCU_SEICON0(rw), SCU_SYSPLLCON0(r), SCU_SYSPLLCON1(r), STM_TIM0(r)	
	Note : The list includes all the SFRs accessed in the context of the API. It lists the SFRs accessed by the driver and called interfaces from other drivers. During runtime, the SFRs accessed from this list may vary based on configuration and execution context.	
Autosar Version	Applicable for Autosar versions 4.2.2 and 4.4.0.	

1.3.3.3 Port_InitCheck

Table 55	Specification for `Port_InitCheck` API	

Syntax	`Std_ReturnType  Port_InitCheck`
	`(`
	`    const Port_ConfigType ConfigPtr`
	`)`	

Service ID	0x05	
Sync/Async	Synchronous	
ASIL Level	B	
Re-entrancy	Non Reentrant	
Parameters (in)	ConfigPtr	Pointer to configuration set

1 Port driver

Table 55 **Specification for `Port_InitCheck` API (continued)**

Parameters (out)	-	-
Parameters (in - out)	-	-
Return	Std_ReturnType	E_OK: Driver is initialized correctly.
		E_NOT_OK: Driver is not initialized correctly.
Description	Performs the initialization check for the PORT driver.	
Source	IFX	
Error handling	-	
Configuration dependencies	-	
User hints	-	
SFR accessed	P_ESR(r), P_IOCR0(r), P_IOCR12(r), P_IOCR4(r), P_IOCR8(r), P_LPCR(r), P_OUT(r), P_PCSR(r), P_PDISC(r), P_PDR0(r), P_PDR1(r)	
	Note : The list includes all the SFRs accessed in the context of the API. It lists the SFRs accessed by the driver and called interfaces from other drivers. During runtime, the SFRs accessed from this list may vary based on configuration and execution context.	
Autosar Version	Applicable for Autosar versions 4.2.2 and 4.4.0.	

1.3.3.4 Port_RefreshPortDirection

Table 56 **Specification for `Port_RefreshPortDirection` API**

| Syntax | ```
void  Port_RefreshPortDirection
(
  void
)
``` | |
|---|---|---|
Service ID	0x02	
Sync/Async	Synchronous	
ASIL Level	B	
Re-entrancy	Non Reentrant	
Parameters (in)	-	-
Parameters (out)	-	-
Parameters (in - out)	-	-
Return	void	-
Description	Refreshes port direction.	

Table 56	Specification for `Port_RefreshPortDirection` API (continued)
Source	AUTOSAR
Error handling	PORT_E_UNINIT
Configuration dependencies	-
User hints	-
SFR accessed	P_IOCR0(w), P_IOCR12(w), P_IOCR4(w), P_IOCR8(w)
	Note : The list includes all the SFRs accessed in the context of the API. It lists the SFRs accessed by the driver and called interfaces from other drivers. During runtime, the SFRs accessed from this list may vary based on configuration and execution context.
Autosar Version	Applicable for Autosar versions 4.2.2 and 4.4.0.

1.3.3.5 Port_SetPinDirection

Table 57	Specification for `Port_SetPinDirection` API	

Syntax	`void  Port_SetPinDirection`
	`(`
	`    const Port_PinType Pin,`
	`    const Port_PinDirectionType Direction`
	`)`	

Service ID	0x01	
Sync/Async	Synchronous	
ASIL Level	B	
Re-entrancy	Reentrant for different pins independent of port.	
Parameters (in)	Pin	Port pin ID
	Direction	Port pin direction
Parameters (out)	-	-
Parameters (in - out)	-	-
Return	void	-
Description	The API sets the PORT pin to the specified direction. Default Configuration: PORT_PIN_IN =PORT_PIN_IN_PULL_UP PORT_PIN_OUT=PORT_PIN_OUT_PUSHPULL	
Source	AUTOSAR	
Error handling	PORT_E_UNINIT , PORT_E_DIRECTION_UNCHANGEABLE, PORT_E_PARAM_PIN, PORT_E_PARAM_INVALID_DIRECTION	

Table 57	Specification for `Port_SetPinDirection` API (continued)
Configuration dependencies	PortSetPinDirectionApi
User hints	-
SFR accessed	P_IOCR0(rw), P_IOCR12(rw), P_IOCR4(rw), P_IOCR8(rw) *Note : The list includes all the SFRs accessed in the context of the API. It lists the SFRs accessed by the driver and called interfaces from other drivers. During runtime, the SFRs accessed from this list may vary based on configuration and execution context.*
Autosar Version	Applicable for Autosar versions 4.2.2 and 4.4.0.

1.3.3.6 Port_SetPinMode

Table 58	Specification for `Port_SetPinMode` API	

Syntax	`void  Port_SetPinMode`
	`(`
	`    const Port_PinType Pin,`
	`    const Port_PinModeType Mode`
	`)`	

Service ID	0x04	
Sync/Async	Synchronous	
ASIL Level	B	
Re-entrancy	Reentrant for different pins independent of port.	
Parameters (in)	Pin Mode	Port pin ID number whose mode has to be set Port pin mode to be set
Parameters (out)	-	-
Parameters (in - out)	-	-
Return	void	-
Description	The API sets the port pin mode during runtime as per the mode passed. For setting/configuring the mode along with the electrical characteristics, user shall pass the pin mode macro (Bitwise OR) with the electrical characteristics macro as an input parameter	
Source	AUTOSAR	
Error handling	PORT_E_UNINIT , PORT_E_PARAM_INVALID_MODE , PORT_E_MODE_UNCHANGEABLE , PORT_E_PARAM_PIN, PORT_E_IMPLAUSIBLE_MODE	
Configuration dependencies	PortSetPinModeApi	
User hints	-	
SFR accessed	P_IOCR0(rw), P_IOCR12(rw), P_IOCR4(rw), P_IOCR8(rw)	

Table 58	Specification for `Port_SetPinMode` API (continued)
	Note : The list includes all the SFRs accessed in the context of the API. It lists the SFRs accessed by the driver and called interfaces from other drivers. During runtime, the SFRs accessed from this list may vary based on configuration and execution context.
Autosar Version	Applicable for Autosar versions 4.2.2 and 4.4.0.

1.3.4 Notifications and Callbacks

The PORT driver does not provide any notification or callbacks.

1.3.5 Scheduled functions

The PORT driver does not provide any scheduled functions.

1.3.6 Interrupt service routines

The PORT driver does not provide any interrupt handlers.

1.3.7 Callout

The driver does not support any callout functions.

1.3.8 Errors Handling

Error Name: Description	Source	Error ID (AS422)	Type (AS422)	Error ID (AS440)	Type (AS440)
PORT_E_IMPLAUSIBLE_MODE : This error code is reported if output characteristics are passed for an input pin or input characteristics for an output pin.	IFX	0x33	SAFETY	0x33	SAFETY
PORT_E_DIRECTION_UNCHANGEABLE: This error code is reported if a port pin for which direction is not changeable is passed as a parameter.	AUTOSAR	0x0B	DET_SAFETY	0x0B	DET_SAFETY
PORT_E_INIT_FAILED : This error code is reported if initialization is invoked with incorrect configuration pointer.	AUTOSAR	0x0C	DET_SAFETY	0x0C	DET_SAFETY
PORT_E_MODE_UNCHANGEABLE : This error code is reported if a port pin for which the mode is not changeable is passed as a parameter.	AUTOSAR	0x0E	DET_SAFETY	0x0E	DET_SAFETY

Error Name: Description	Source	Error ID (AS422)	Type (AS422)	Error ID (AS440)	Type (AS440)
PORT_E_PARAM_INVALID_DIRECTION: This error code is reported if the passed direction is invalid.	IFX	0x32	SAFETY	0x32	SAFETY
PORT_E_PARAM_INVALID_MODE : This error code is reported if the passed mode and passed input/output characteristic value is invalid.	AUTOSAR	0x0D	DET_SAFETY	0x0D	DET_SAFETY
PORT_E_PARAM_PIN: This error code is reported if the port pin ID passed is invalid.	AUTOSAR	0x0A	DET_SAFETY	0x0A	DET_SAFETY
PORT_E_PARAM_POINTER : This error code is reported if a null pointer is passed as a parameter.	AUTOSAR	0x10	DET_SAFETY	0x10	DET_SAFETY
PORT_E_UNINIT : This error code is reported if any API is invoked prior to the driver initialization.	AUTOSAR	0x0F	DET_SAFETY	0x0F	DET_SAFETY

1.3.9   Deviations and limitations

This section describes the deviations and limitations of the PORT driver.

1.3.9.1   Deviations

This section describes the deviation of the PORT driver.

1.3.9.1.1   Software specification deviations

The PORT driver does not have any deviations.

1.3.9.1.2   AMDC Violations

This section describes the violations reported by the vector AMDC checker tool with respect to AUTOSAR.

Table 59   Violations reported by AMDC checker tool for A207

AMDC Rule	A207
Description	Minimum value of parameter 'Port/PortConfigSet/ PortContainer/PortPin/PortPinId' in VSMD (0) may not be smaller than minimum value defined in StMD (1). [Port.bmd]

1.3.9.1.3 VSMD Violations

The PORT driver does not have any VSMD violations.

1.3.9.2 Limitations

The PORT driver does not have any limitations.

Revision history

Table 60   **Revision history**

Date	Version	Description
2020-11-18	2.0	Document Released
2020-11-18	1.1	• SFR access information updated
2020-08-14	1.0	Document Released
2020-08-04	0.1	• Initial Version • Port driver chapter moved from MC-ISAR_TC3xx_UM_Basic to this document • Dynamic change of port output setting between push-pull and open-drain; port input setting between no-pull, pull-down, pull-up added for Port_SetPinMode() API. • Added PortPinEcucPartitionRef and PortEcucPartitionRef for AS440 • AMDC Violation added

Trademarks

All referenced product or service names and trademarks are the property of their respective owners.