
User Manual

for MPC574XG PORT Driver

Document Number: UM35PORTASR4.2 Rev002 R1.0.0
Rev. 1.0





Contents

Section number	Title	Page
Chapter 1		
Revision History		
Chapter 2		
Introduction		
2.1	Supported Derivatives.....	9
2.2	Overview.....	10
2.3	About this Manual.....	10
2.4	Acronyms and Definitions.....	11
2.5	Reference List.....	11
Chapter 3		
Driver		
3.1	Requirements.....	13
3.2	Driver Design Summary.....	13
3.3	Hardware Resources.....	14
3.4	Deviation from Requirements.....	14
3.5	PORT Driver limitations.....	15
3.6	Driver usage and configuration tips.....	15
3.7	Runtime Errors.....	15
3.8	Software specification.....	16
3.8.1	Define Reference.....	16
3.8.1.1	Define PORT_E_DIRECTION_UNCHANGEABLE.....	16
3.8.1.2	Define PORT_E_MODE_UNCHANGEABLE.....	16
3.8.1.3	Define PORT_E_PARAM_CONFIG.....	17
3.8.1.4	Define PORT_E_PARAM_INVALID_MODE.....	17
3.8.1.5	Define PORT_E_PARAM_PIN.....	17
3.8.1.6	Define PORT_E_PARAM_POINTER.....	18
3.8.1.7	Define PORT_E_UNINIT.....	18
3.8.1.8	Define PORT_GETVERSIONINFO_ID.....	18

Section number	Title	Page
3.8.1.9	Define PORT_INIT_ID.....	19
3.8.1.10	Define PORT_SETPINDIRECTION_ID.....	19
3.8.1.11	Define PORT_SETPINMODE_ID.....	20
3.8.1.12	Define PORT_REFRESHPINDIRECTION_ID.....	20
3.8.1.13	Define PORT_GPIO_MODE.....	20
3.8.1.14	Define PORT_ALT1_FUNC_MODE.....	20
3.8.1.15	Define PORT_ALT2_FUNC_MODE.....	21
3.8.1.16	Define PORT_ALT3_FUNC_MODE.....	21
3.8.1.17	Define PORT_ALT4_FUNC_MODE.....	21
3.8.1.18	Define PORT_ALT5_FUNC_MODE.....	21
3.8.1.19	Define PORT_ALT6_FUNC_MODE.....	22
3.8.1.20	Define PORT_ALT7_FUNC_MODE.....	22
3.8.1.21	Define PORT_ONLY_OUTPUT_MODE.....	22
3.8.1.22	Define PORT_ONLY_INPUT_MODE.....	22
3.8.1.23	Define PORT_ANALOG_INPUT_MODE.....	23
3.8.1.24	Define PORT_INPUT1_MODE.....	23
3.8.1.25	Define PORT_INPUT2_MODE.....	23
3.8.1.26	Define PORT_INPUT3_MODE.....	23
3.8.1.27	Define PORT_INPUT4_MODE.....	24
3.8.1.28	Define PORT_INPUT5_MODE.....	24
3.8.1.29	Define PORT_INPUT6_MODE.....	24
3.8.1.30	Define PORT_INOUT1_MODE.....	24
3.8.1.31	Define PORT_INOUT2_MODE.....	25
3.8.1.32	Define PORT_INOUT3_MODE.....	25
3.8.1.33	Define PORT_INOUT4_MODE.....	25
3.8.1.34	Define PORT_INOUT5_MODE.....	25
3.8.1.35	Define PORT_INOUT6_MODE.....	25
3.8.1.36	Define PORT_INOUT7_MODE.....	26
3.8.2	Enum Reference.....	26

Section number	Title	Page
3.8.3	Function Reference.....	26
3.8.3.1	Function Port_Init.....	26
3.8.3.2	Function Port_SetPinDirection.....	27
3.8.3.3	Function Port_SetPinMode.....	27
3.8.3.4	Function Port_RefreshPortDirection.....	28
3.8.3.5	Function Port_GetVersionInfo.....	28
3.8.4	Structs Reference.....	29
3.8.4.1	Structure Port_ConfigType.....	29
3.8.4.2	Structure Port_InMuxSettingType.....	30
3.8.4.3	Structure Port_InoutSettingType.....	31
3.8.5	Types Reference.....	32
3.9	Symbolic Names Disclaimer.....	32

Chapter 4 Tresos Configuration Plug-in

4.1	Configuration elements of Port.....	33
4.2	IMPLEMENTATION_CONFIG_VARIANT.....	33
4.3	PortGeneral.....	34
4.3.1	PortDevErrorDetect (PortGeneral).....	34
4.3.2	PortSetPinDirectionApi (PortGeneral).....	35
4.3.3	PortSetPinModeApi (PortGeneral).....	35
4.3.4	PortVersionInfoApi (PortGeneral).....	35
4.3.5	PortSetPinModeDoesNotTouchGpioLevel (PortGeneral).....	36
4.3.6	PortEnableUserModeSupport (PortGeneral).....	36
4.4	Form CommonPublishedInformation.....	36
4.4.1	ArReleaseMajorVersion (CommonPublishedInformation).....	37
4.4.2	ArReleaseMinorVersion (CommonPublishedInformation).....	37
4.4.3	ArReleaseRevisionVersion (CommonPublishedInformation).....	38
4.4.4	ModuleId (CommonPublishedInformation).....	38
4.4.5	SwMajorVersion (CommonPublishedInformation).....	39

Section number	Title	Page
4.4.6	SwMinorVersion (CommonPublishedInformation).....	39
4.4.7	SwPatchVersion (CommonPublishedInformation).....	40
4.4.8	VendorApiInfix (CommonPublishedInformation).....	40
4.4.9	VendorId (CommonPublishedInformation).....	41
4.5	PortConfigSet.....	41
4.5.1	PortContainer.....	42
4.5.1.1	PortNumberOfPortPins (PortContainer).....	42
4.5.1.2	PortPin.....	43
4.5.1.2.1	PortPinWpe (PortPin).....	43
4.5.1.2.2	PortPinWps (PortPin).....	44
4.5.1.2.3	PortPinOde (PortPin).....	44
4.5.1.2.4	PortPin SafeMode (PortPin).....	44
4.5.1.2.5	PortPinWithReadback(PortPin).....	45
4.5.1.2.6	PortPin Hysteresis Control(PortPin).....	45
4.5.1.2.7	PortPinDirectionChangeable (PortPin).....	45
4.5.1.2.8	PortPinModeChangeable (PortPin).....	46
4.5.1.2.9	PortPinId (PortPin).....	46
4.5.1.2.10	PortPinPcr (PortPin).....	46
4.5.1.2.11	PortPinDirection (PortPin).....	47
4.5.1.2.12	PortPinInitialMode (PortPin).....	47
4.5.1.2.13	PortPinMode (PortPin).....	48
4.5.1.2.14	PortPinLevelValue (PortPin).....	48
4.5.1.2.15	PortPinSlewRate (PortPin).....	49
4.5.1.3	NotUsedPortPin.....	49
4.5.1.3.1	PortPinWpe (NotUsedPortPin).....	50
4.5.1.3.2	PortPinWps (NotUsedPortPin).....	50
4.5.1.3.3	PortPin Ode (NotUsedPortPin).....	50
4.5.1.3.4	PortPinDirection (NotUsedPortPin).....	50
4.5.1.3.5	PortPinLevelValue(NotUsedPortPin).....	51

Chapter 1

Revision History

Table 1-1. Revision History

Revision	Date	Author	Description
1.0	17-Feb-2017	Duc Ta (B53913)	Version for Calypso Release 1.0.0



Chapter 2

Introduction

This User Manual describes NXP Semiconductor AUTOSAR Port (Port) for MPC574XG .

AUTOSAR Port driver configuration parameters and deviations from the specification are described in Port Driver chapter of this document. AUTOSAR Port driver requirements and APIs are described in the AUTOSAR Port driver software specification document.

2.1 Supported Derivatives

The software described in this document is intended to be used with the following microcontroller devices of NXP Semiconductor .

Table 2-1. MPC574XG Derivatives

NXP Semiconductor	MPC5748G_LQFP176, MPC5748G_MAPBGA256, MPC5748G_MAPBGA324, MPC5747G_LQFP176, MPC5747G_MAPBGA256, MPC5747G_MAPBGA324, MPC5746G_LQFP176, MPC5746G_MAPBGA256, MPC5746G_MAPBGA324, MPC5748C_LQFP176, MPC5748C_MAPBGA256, MPC5748C_MAPBGA324, MPC5747C_LQFP176, MPC5747C_MAPBGA256, MPC5747C_MAPBGA324, MPC5746C_LQFP176, MPC5746C_MAPBGA256, MPC5746C_MAPBGA324, MPC5746C_MAPBGA100, MPC5745C_LQFP176, MPC5745C_MAPBGA256, MPC5745C_MAPBGA100, MPC5744C_LQFP176, MPC5744C_MAPBGA256,
-------------------	--

Table 2-1. MPC574XG Derivatives

	MPC5744C_MAPBGA100, MPC5746B_LQFP176, MPC5746B_MAPBGA256, MPC5746B_MAPBGA100, MPC5744B_LQFP176, MPC5744B_MAPBGA256, MPC5744B_MAPBGA100, MPC5745B_LQFP176, MPC5745B_MAPBGA256, MPC5745B_MAPBGA100
--	---

All of the above microcontroller devices are collectively named as MPC574XG .

2.2 Overview

AUTOSAR (AUTomotive Open System ARchitecture) is an industry partnership working to establish standards for software interfaces and software modules for automobile electronic control systems.

AUTOSAR

- paves the way for innovative electronic systems that further improve performance, safety and environmental friendliness.
- is a strong global partnership that creates one common standard: "Cooperate on standards, compete on implementation".
- is a key enabling technology to manage the growing electrics/electronics complexity. It aims to be prepared for the upcoming technologies and to improve cost-efficiency without making any compromise with respect to quality.
- facilitates the exchange and update of software and hardware over the service life of the vehicle.

2.3 About this Manual

This Technical Reference employs the following typographical conventions:

Boldface type: Bold is used for important terms, notes and warnings.

Italic font: Italic typeface is used for code snippets in the text. Note that C language modifiers such "const" or "volatile" are sometimes omitted to improve readability of the presented code.

Notes and warnings are shown as below:

Note

This is a note.

2.4 Acronyms and Definitions

Table 2-2. Acronyms and Definitions

Term	Definition
ADC	Analog to Digital Converter
API	Application Programming Interface
ASM	Assembler
AUTOSAR	Automotive Open System Architecture
BSMI	Basic Software Make file Interface
CAN	Controller Area Network
C/CPP	C and C++ Source Code
CS	Chip Select
CTU	Cross Trigger Unit
DEM	Diagnostic Event Manager
DET	Development Error Tracer
DMA	Direct Memory Access
ECU	Electronic Control Unit
FIFO	First In First Out
LSB	Least Significant Bit
MCU	Micro Controller Unit
MIDE	Multi Integrated Development Environment
MSB	Most Significant Bit
N/A	Not Applicable
RAM	Random Access Memory
SIU	Systems Integration Unit
SIUL	Systems Integration Unit Lite
SIUL2	Systems Integration Unit Lite version 2
SWS	Software Specification
VLE	Variable Length Encoding
XML	Extensible Markup Language

2.5 Reference List

Table 2-3. Reference List

#	Title	Version
1	AUTOSAR 4.2 Rev0002Port Driver Software Specification Document.	V3.2.0
2	MPC5748G Reference Manual	Rev. 5, 12/2016
3	MPC5746C Reference Manual	Rev. 4, 12/2016
4	MPC5748G_1N81M_Rev.2 (official document) (1N81M)	Jun-16
5	MPC5748G_1N81M_0N78S_Comparison_Summary_v2_0 (internal document) (1N81M, 0N78S)	31.10.2016
6	MPC5746C_1N06M_Rev.4 (official document) (1N06M)	Jul-16
7	MPC5746C_cut1.1_cut2.0_cut2.1_comparison_v0 (internal document) (1N06M, 0N84S, 1N84S)	14-Sep-16
8	C3M_cut2.1_new_errata_20170113 (internal document) (1N84S)	13-Jan-17

Chapter 3

Driver

3.1 Requirements

Requirements for this driver are detailed in the AUTOSAR 4.2 Rev0002Port Driver Software Specification document (See Table [Reference List](#)).

3.2 Driver Design Summary

This module provides the service for initializing the whole PORT structure of the microcontroller. Many ports and port pins can be assigned to various functionalities, e.g.

- General purpose I/O
- ADC
- SPI
- SCI
- PWM
- CAN
- LIN
- etc

For this reason, there is an overall configuration and initialization of this port structure. The configuration and mode of these port pins is microcontroller and ECU dependent.

Port initialisation data are written to each port as efficiently as possible. This PORT driver module completes the overall configuration and initialisation of the port structure which is used in the DIO driver module. Therefore, the DIO driver works on pins and ports which are configured by the PORT driver.

The PORT driver is initialised prior to use of the DIO functions. Otherwise DIO functions will exhibit undefined behaviour.

3.3 Hardware Resources

The hardware configured by the Port driver is SIUL2.

3.4 Deviation from Requirements

The driver deviates from the AUTOSAR Port Driver software specification in some places. Table identifies the AUTOSAR requirements that are not fully implemented, implemented differently, or out of scope for the Port driver. Table [Table 3-1](#) provides Status column description.

Table 3-1. Deviations Status Column Description

Term	Definition
N/A	Not available
N/T	Not testable
N/S	Out of scope
N/I	Not implemented
N/F	Not fully implemented

Below table identifies the AUTOSAR requirements that are not fully implemented, implemented differently, or out of scope for the driver.

Table 3-2. Driver Deviations Table

Requirement	Status	Description	Notes
ECUC_Port_001 28	N/A	Name : PortPinInitialMode {PORT_PIN_INITIAL_MODE} Description : Port pin mode from mode list for use with Port_Init() function. Range: PORT_PIN_MODE_ADC PORT_PIN_MODE_CAN PORT_PIN_MODE_DIO PORT_PIN_MODE_DIO_GPT PORT_PIN_MODE_DIO_WDG PORT_PIN_MODE_FLEXRAY PORT_PIN_MODE_ICUPort PORT_PIN_MODE_LINPort PORT_PIN_MODE_MEMPort PORT_PIN_MODE_PWMPort PORT_PIN_MODE_SPIPort	Currently implemented in a different mode in MCAL 4.0.
ECUC_Port_001 30	N/A	Name : PortPinMode {PORT_PIN_MODE} Description : Port pin mode from mode list. Note that more than one mode is allowed by	Replaced by requirement CPR- MCAL-781.port.

Table continues on the next page...

Table 3-2. Driver Deviations Table (continued)

Requirement	Status	Description	Notes
		default. That way it is e.g. possible to combine DIO with another mode such as ICU. More HW specific literals may be added by module implementor. E.g. SPI may be further broken down into SpiCsHw, SpiCsSw, etc. if necessary. Range : PORT_PIN_MODE_ADC PORT_PIN_MODE_CAN PORT_PIN_MODE_DIO PORT_PIN_MODE_DIO_GPT PORT_PIN_MODE_DIO_WDG PORT_PIN_MODE_FLEXRAY PORT_PIN_MODE_ICUPort PORT_PIN_MODE_LINPort PORT_PIN_MODE_MEMPort PORT_PIN_MODE_PWMPort PORT_PIN_MODE_SPIPort	
SWS_Port_00205	N/A	Port_Lcfg.c shall include Port_MemMap.h and Port.h	Currently no support for link-time configuration is provided.
SWS_Port_00220	N/A	The type Port_PinDirectionType shall be of enumeration type having range as PORT_PIN_IN and PORT_PIN_OUT.	Replaced by SMCAL_SW066.port

As a deviation from standard:

Port_PBcfg_**VariantNo**.c files will contain the definition for all parameters that are variant aware, independent of the configuration class that will be selected (PC, LT, PB).

Port_Cfg.c file will contain the definition for all parameters that are not variant aware.

3.5 PORT Driver limitations

None.

3.6 Driver usage and configuration tips

None

3.7 Runtime Errors

This driver doesn't generate any runtime error.

3.8 Software specification

The following sections contains driver software specifications.

3.8.1 Define Reference

Constants supported by the driver are as per AUTOSAR Port Driver software specification Version 4.2 Rev0002 .

3.8.1.1 Define PORT_E_DIRECTION_UNCHANGEABLE

Port Pin Direction not configured as changeable.

Details:

Det Error value, returned by Port_SetPinDirection if the passed PortPin have unchangeable direction.

**Table 3-3. Define PORT_E_DIRECTION_UNCHANGEABLE
Description**

Name	PORT_E_DIRECTION_UNCHANGEABLE
Initializer	(uint8)0x0B

3.8.1.2 Define PORT_E_MODE_UNCHANGEABLE

API `Port_SetPinMode()` service called when mode is unchangeable.

Details:

Det Error value, returned by Port_SetPinMode function if the passed PortPin have a unchangeable Mode.

**Table 3-4. Define PORT_E_MODE_UNCHANGEABLE
Description**

Name	PORT_E_MODE_UNCHANGEABLE
-------------	--------------------------

Table continues on the next page...

Table 3-4. Define PORT_E_MODE_UNCHANGEABLE Description (continued)

Initializer	(uint8)0x0E
--------------------	-------------

3.8.1.3 Define PORT_E_PARAM_CONFIG

API `Port_Init()` service called with wrong parameter.

Details:

Det Error value, returned by `Port_Init` function if `Port_Init` is called with wrong parameter.

Table 3-5. Define PORT_E_PARAM_CONFIG Description

Name	PORT_E_PARAM_CONFIG
Initializer	(uint8)0x0C

3.8.1.4 Define PORT_E_PARAM_INVALID_MODE

API `Port_SetPinMode()` service called when mode is invalid.

Details:

Det Error value, returned by `Port_SetPinMode` function if the passed `PortPinMode` is invalid.

Table 3-6. Define PORT_E_PARAM_INVALID_MODE Description

Name	PORT_E_PARAM_INVALID_MODE
Initializer	(uint8)0x0D

3.8.1.5 Define PORT_E_PARAM_PIN

Invalid Port Pin ID requested.

Details:

Det Error value, returned by Port_SetPinDirection and Port_SetPinMode if a wrong PortPin ID is passed.

Table 3-7. Define PORT_E_PARAM_PIN Description

Name	PORT_E_PARAM_PIN
Initializer	(uint8)0x0A

3.8.1.6 Define PORT_E_PARAM_POINTER

API service called with NULL Pointer Parameter.

Details:

Det Error value, returned by Port_GetVersionInfo function if API is called with NULL Pointer Parameter.

Table 3-8. Define PORT_E_PARAM_POINTER Description

Name	PORT_E_PARAM_POINTER
Initializer	(uint8)0x10

3.8.1.7 Define PORT_E_UNINIT

API service called without module initialization.

Details:

Det Error value, returned by a function if API service called prior to module initialization.

Table 3-9. Define PORT_E_UNINIT Description

Name	PORT_E_UNINIT
Initializer	(uint8)0x0F

3.8.1.8 Define PORT_GETVERSIONINFO_ID

API service ID for PORT get version info function.

Details:

Parameters used when raising an error/exception.

Table 3-10. Define PORT_GETVERSIONINFO_ID Description

Name	PORT_GETVERSIONINFO_ID
Initializer	(uint8)0x03

3.8.1.9 Define PORT_INIT_ID

API service ID for PORT Init function.

Details:

Parameters used when raising an error/exception.

Table 3-11. Define PORT_INIT_ID Description

Name	PORT_INIT_ID
Initializer	(uint8)0x00

3.8.1.10 Define PORT_SETPINDIRECTION_ID

API service ID for PORT set pin direction function.

Details:

Parameters used when raising an error/exception.

Table 3-12. Define PORT_SETPINDIRECTION_ID Description

Name	PORT_SETPINDIRECTION_ID
Initializer	(uint8)0x01

3.8.1.11 Define PORT_SETPINMODE_ID

API service ID for PORT set pin mode.

Details:

Parameters used when raising an error/exception.

Table 3-13. Define PORT_SETPINMODE_ID Description

Name	PORT_SETPINMODE_ID
Initializer	(uint8)0x04

3.8.1.12 Define PORT_REFRESHPINDIRECTION_ID

API service ID for PORT refresh pin direction function.

Details:

Parameters used when raising an error/exception.

Table 3-14. Define PORT_REFRESHPINDIRECTION_ID Description

Name	PORT_REFRESHPINDIRECTION_ID
Initializer	(uint8)0x02

3.8.1.13 Define PORT_GPIO_MODE

Port GPIO Mode.

Table 3-15. Define PORT_GPIO_MODE Description

Name	PORT_GPIO_MODE
Initializer	((Port_PinModeType)0)

3.8.1.14 Define PORT_ALT1_FUNC_MODE

Port Alternate 1 Mode.

Table 3-16. Define PORT_ALT1_FUNC_MODE Description

Name	PORT_ALT1_FUNC_MODE
Initializer	((Port_PinModeType)1)

3.8.1.15 Define PORT_ALT2_FUNC_MODE

Port Alternate 2 Mode.

Table 3-17. Define PORT_ALT2_FUNC_MODE Description

Name	PORT_ALT2_FUNC_MODE
Initializer	((Port_PinModeType)2)

3.8.1.16 Define PORT_ALT3_FUNC_MODE

Port Alternate 3 Mode.

Table 3-18. Define PORT_ALT3_FUNC_MODE Description

Name	PORT_ALT3_FUNC_MODE
Initializer	((Port_PinModeType)3)

3.8.1.17 Define PORT_ALT4_FUNC_MODE

Port Alternate 4 Mode.

Table 3-19. Define PORT_ALT4_FUNC_MODE Description

Name	PORT_ALT4_FUNC_MODE
Initializer	((Port_PinModeType)4)

3.8.1.18 Define PORT_ALT5_FUNC_MODE

Port Alternate 5 Mode.

Table 3-20. Define PORT_ALT5_FUNC_MODE Description

Name	PORT_ALT5_FUNC_MODE
Initializer	((Port_PinModeType)5)

3.8.1.19 Define PORT_ALT6_FUNC_MODE

Port Alternate 6 Mode.

Table 3-21. Define PORT_ALT6_FUNC_MODE Description

Name	PORT_ALT6_FUNC_MODE
Initializer	((Port_PinModeType)6)

3.8.1.20 Define PORT_ALT7_FUNC_MODE

Port Alternate 7 Mode.

Table 3-22. Define PORT_ALT7_FUNC_MODE Description

Name	PORT_ALT7_FUNC_MODE
Initializer	((Port_PinModeType)7)

3.8.1.21 Define PORT_ONLY_OUTPUT_MODE

Port Only Output.

Table 3-23. Define PORT_ONLY_OUTPUT_MODE Description

Name	PORT_ONLY_OUTPUT_MODE
Initializer	(Port_PinModeType)8

3.8.1.22 Define PORT_ONLY_INPUT_MODE

Port Only Input.

Table 3-24. Define PORT_ONLY_INPUT_MODE Description

Name	PORT_ONLY_INPUT_MODE
-------------	----------------------

Table continues on the next page...

**Table 3-24. Define PORT_ONLY_INPUT_MODE Description
(continued)**

Initializer	((Port_PinModeType)9)
--------------------	-----------------------

3.8.1.23 Define PORT_ANALOG_INPUT_MODE

Port Analog input Mode.

Table 3-25. Define PORT_ANALOG_INPUT_MODE Description

Name	PORT_ANALOG_INPUT_MODE
Initializer	(Port_PinModeType)10

3.8.1.24 Define PORT_INPUT1_MODE

Port Input 1 Mode.

Table 3-26. Define PORT_INPUT1_MODE Description

Name	PORT_INPUT1_MODE
Initializer	(Port_PinModeType)11

3.8.1.25 Define PORT_INPUT2_MODE

Port Input 2 Mode.

Table 3-27. Define PORT_INPUT2_MODE Description

Name	PORT_INPUT2_MODE
Initializer	(Port_PinModeType)12

3.8.1.26 Define PORT_INPUT3_MODE

Port Input 3 Mode.

Table 3-28. Define PORT_INPUT3_MODE Description

Name	PORT_INPUT3_MODE
-------------	------------------

Table continues on the next page...

Table 3-28. Define PORT_INPUT3_MODE Description (continued)

Initializer	(Port_PinModeType)13
--------------------	----------------------

3.8.1.27 Define PORT_INPUT4_MODE

Port Input 4 Mode.

Table 3-29. Define PORT_INPUT4_MODE Description

Name	PORT_INPUT4_MODE
Initializer	(Port_PinModeType)14

3.8.1.28 Define PORT_INPUT5_MODE

Port Input 5 Mode.

Table 3-30. Define PORT_INPUT5_MODE Description

Name	PORT_INPUT5_MODE
Initializer	(Port_PinModeType)15

3.8.1.29 Define PORT_INPUT6_MODE

Port Input 6 Mode.

Table 3-31. Define PORT_INPUT6_MODE Description

Name	PORT_INPUT6_MODE
Initializer	(Port_PinModeType)16

3.8.1.30 Define PORT_INOUT1_MODE

Port Inout 1 Mode.

Table 3-32. Define PORT_INOUT1_MODE Description

Name	PORT_INOUT1_MODE
Initializer	(Port_PinModeType)17

3.8.1.31 Define PORT_INOUT2_MODE

Port Inout 2 Mode.

Table 3-33. Define PORT_INOUT2_MODE Description

Name	PORT_INOUT2_MODE
Initializer	(Port_PinModeType)18

3.8.1.32 Define PORT_INOUT3_MODE

Port Inout 3 Mode.

Table 3-34. Define PORT_INOUT3_MODE Description

Name	PORT_INOUT3_MODE
Initializer	(Port_PinModeType)19

3.8.1.33 Define PORT_INOUT4_MODE

Port Inout 4 Mode.

Table 3-35. Define PORT_INOUT4_MODE Description

Name	PORT_INOUT4_MODE
Initializer	(Port_PinModeType)20

3.8.1.34 Define PORT_INOUT5_MODE

Port Inout 5 Mode.

Table 3-36. Define PORT_INOUT5_MODE Description

Name	PORT_INOUT5_MODE
Initializer	(Port_PinModeType)21

3.8.1.35 Define PORT_INOUT6_MODE

Port Inout 6 Mode.

Table 3-37. Define PORT_INOUT6_MODE Description

Name	PORT_INOUT6_MODE
Initializer	(Port_PinModeType)22

3.8.1.36 Define PORT_INOUT7_MODE

Port Inout 7 Mode.

Table 3-38. Define PORT_INOUT7_MODE Description

Name	PORT_INOUT7_MODE
Initializer	(Port_PinModeType)23

3.8.2 Enum Reference

Enumeration of all constants supported by the driver are as per AUTOSAR Port Driver software specification Version 4.2 Rev0002 .

3.8.3 Function Reference

Functions of all functions supported by the driver are as per AUTOSAR Port Driver software specification Version 4.2 Rev0002 .

3.8.3.1 Function Port_Init

Initializes the Port Driver module.

Details:

The function `Port_Init()` will initialize ALL ports and port pins with the configuration set pointed to by the parameter `ConfigPtr`. It always requires an input as a valid pointer.

Pre:

Function `Port_Init()` should not have been called before.

Post: `Port_Init()` must be called before all other Port Driver module's functions otherwise no operation can occur on the MCU ports and port pins.

Prototype: `void Port_Init(const Port_ConfigType *ConfigPtr);`

Table 3-39. Port_Init Arguments

Type	Name	Direction	Description
<code>const Port_ConfigType*</code>	<code>ConfigPtr</code>	input	A pointer to the structure which contains initialization parameters.

3.8.3.2 Function Port_SetPinDirection

Sets the port pin direction.

Details:

The function `Port_SetPinDirection()` will set the port pin direction during runtime.

Pre: `Port_Init()` must have been called first. In order to change the pin direction the `PortPinDirectionChangeable` flag must have been set to `TRUE`.

Prototype: `void Port_SetPinDirection(Port_PinType Pin, Port_PinDirectionType Direction);`

Table 3-40. Port_SetPinDirection Arguments

Type	Name	Direction	Description
<code>Port_PinType</code>	<code>Pin</code>	input	Pin ID number.
<code>Port_PinDirectionType</code>	<code>Direction</code>	input	Port Pin direction.

3.8.3.3 Function Port_SetPinMode

Sets the port pin mode.

Details:

The function `Port_SetPinMode()` will set the port pin mode of the referenced pin during runtime.

Pre: `Port_Init()` must have been called first.

Prototype: `void Port_SetPinMode(Port_PinType Pin, Port_PinModeType Mode);`

Table 3-41. Port_SetPinMode Arguments

Type	Name	Direction	Description
Port_PinType	Pin	input	Pin ID number.
Port_PinModeType	Mode	input	New Port Pin mode to be set on port pin.

3.8.3.4 Function Port_RefreshPortDirection

Refreshes port direction.

Details:

This function will refresh the direction of all configured ports to the configured direction. The PORT driver will exclude from refreshing those port pins that are configured as "pin direction changeable during runtime".

Pre: `Port_Init()` must have been called first.

Prototype: `void Port_RefreshPortDirection(void);`

3.8.3.5 Function Port_GetVersionInfo

Returns the version information of this module.

Details:

The function `Port_GetVersionInfo()` will return the version information of this module. The version information includes:

- Module Id,
- Vendor Id,
- Vendor specific version numbers.

Pre: None

Prototype: void Port_GetVersionInfo(Std_VersionInfoType *versioninfo);

Table 3-42. Port_GetVersionInfo Arguments

Type	Name	Direction	Description
Std_VersionInfoType *	versioninfo	input, output	Pointer to where to store the version information of this module.

3.8.4 Structs Reference

Data structures supported by the driver are as per AUTOSAR Port Driver software specification Version 4.2 Rev0002 .

3.8.4.1 Structure Port_ConfigType

Structure needed by Port_Init().

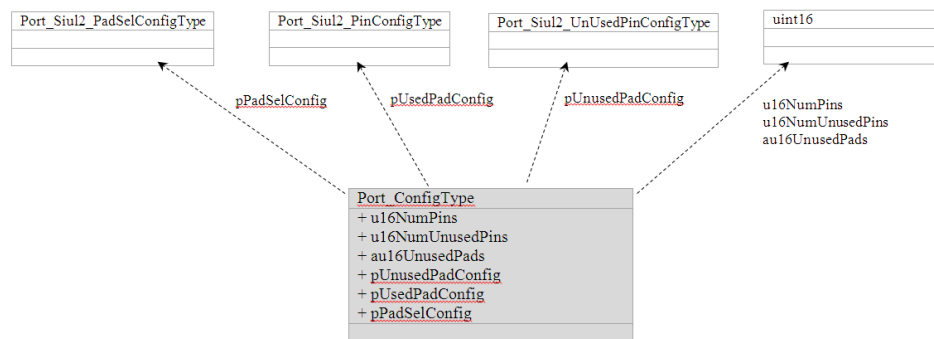


Figure 3-1. Struct Port_ConfigType

Details:

The structure `Port_ConfigType` is a type for the external data structure containing the initialization data for the PORT Driver.

Note

The user must use the symbolic names defined in the configuration tool.

Declaration:

```
typedef struct
{
    VAR(uint16,AUTOMATIC) u16NumPins;
    VAR(uint16,AUTOMATIC) u16NumUnusedPins;
    P2CONST(uint16,AUTOMATIC,PORT_APPL_CONST) au16UnusedPads;
    P2CONST(Port_Siul2_UnUsedPinConfigType,AUTOMATIC,PORT_APPL_CONST) pUnusedPadConfig;
    P2CONST(Port_Siul2_PinConfigType,AUTOMATIC,PORT_APPL_CONST) pUsedPadConfig;
    P2CONST(Port_Siul2_PadSelConfigType,AUTOMATIC,PORT_APPL_CONST) pPadSelConfig;
} Port_ConfigType;
```

Table 3-43. Structure Port_ConfigType member description

Member	Description
u16NumPins	Number of used pads.
u16NumUnusedPins	Number of unused pads.
au16UnusedPads	Unused pad id's array.
pUnusedPadConfig	Unused pad configuration.
pUsedPadConfig	Used pads data configuration.
pPadSelConfig	Used pads INMUX configuration.

3.8.4.2 Structure Port_InMuxSettingType

Structure needed for settings for a INMUX register.

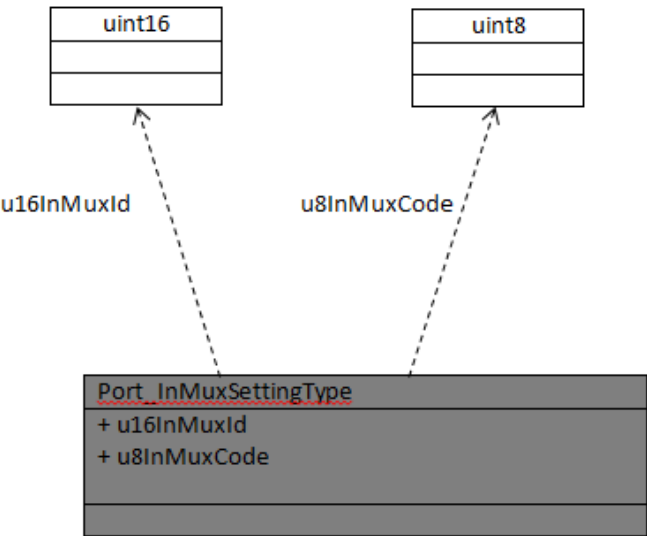


Figure 3-2. Struct Port_InMuxSettingType

Details:

The structure `Port_InMuxSettingType` manages ExtraFunctions PinMode.

Declaration:

```
typedef struct
{
    VAR(uint16,AUTOMATIC) u16InMuxId;
    VAR(uint8, AUTOMATIC) u8InMuxCode;
} Port_InMuxSettingType;
```

Table 3-44. Structure Port_InMuxSettingType member description

Member	Description
u16InMuxId	The id of the INMUX register to be configured.
u8InMuxCode	The value of the INMUX reg for signal allocation.

3.8.4.3 Structure Port_InoutSettingType

Structure needed for settings for INOUT functionality.

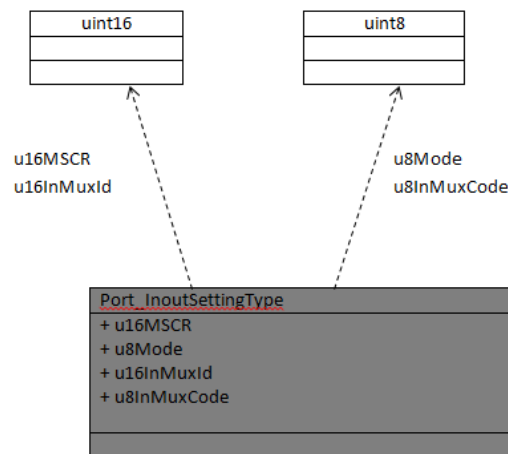


Figure 3-3. Struct Port_InoutSettingType

Details:

The structure `Port_InoutSettingType` manages Inout PinModes.

Declaration:

```
typedef struct
{
    VAR(uint16,AUTOMATIC) u16MSCR;
    VAR(uint8,AUTOMATIC) u8Mode;
    VAR(uint16,AUTOMATIC) u16InMuxId;
```

Symbolic Names Disclaimer

```
VAR(uint8, AUTOMATIC) u8InMuxCode;  
} Port_InoutSettingType;
```

Table 3-45. Structure Port_InoutSettingType member description

Member	Description
u16MSCR	The id of the MSCR register to be configured.
u8Mode	The mode to be configured.
u16InMuxId	The id of the INMUX register to be configured.
u8InMuxCode	The value of the INMUX reg for signal allocation.

3.8.5 Types Reference

Types supported by the driver are as per AUTOSAR Port Driver software specification Version 4.2 Rev0002 .

3.9 Symbolic Names Disclaimer

All containers having the symbolic name tag set as true in the Autosar schema will generate defines like:

```
#define <Container_Short_Name> <Container_ID>
```

For this reason it is forbidden to duplicate the name of such containers across the MCAL configuration, or to use names that may trigger other compile issues (e.g. match existing #ifdefs arguments).

Chapter 4

Tresos Configuration Plug-in

This chapter describes the Tresos configuration plug-in for the Port Driver. The most of the parameters are described below.

4.1 Configuration elements of Port

Included forms :

- IMPLEMENTATION_CONFIG_VARIANT
- PortGeneral
- PortConfigSet
- CommonPublishedInformation

Table 4-1. Revision table

Revision	Date
4.1.0	2010-12-03

4.2 IMPLEMENTATION_CONFIG_VARIANT

VariantPreCompile: Only precompile time configuration parameters. Only one set of parameters. VariantPostBuild: Mix of precompile and postbuild time configuration parameters. More sets of parameters. If Config Variant = VariantPreCompile, the files Port_Cfg.h and Port_Cfg.c should be used. If Config Variant = VariantPostBuild, the files Port_Cfg.h and Port_PBcfg.c should be used.

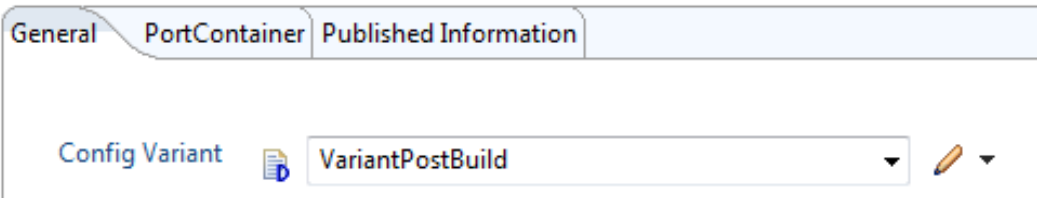


Figure 4-1. Tresos Plugin snapshot for IMPLEMENTATION_CONFIG_VARIANT

Table 4-2. Attribute IMPLEMENTATION_CONFIG_VARIANT detailed description

Property	Value
Label	Config Variant
Default	VariantPostBuild
Range	VariantPostBuild VariantPreCompile

4.3 PortGeneral

Module wide configuration parameters of the PORT driver.

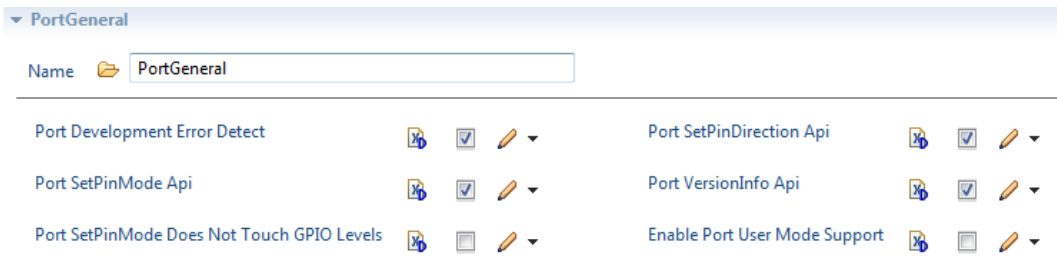


Figure 4-2. Tresos Plugin snapshot for PortGeneral form.

4.3.1 PortDevErrorDetect (PortGeneral)

Switches the Development Error Detection and Notification ON or OFF.

Table 4-3. Attribute PortDevErrorDetect (PortGeneral) detailed description

Property	Value
Label	Port Development Error Detect
Type	BOOLEAN
Origin	AUTOSAR_ECUC
Symbolic Name	false
Default	true

4.3.2 PortSetPinDirectionApi (PortGeneral)

Pre-processor switch to enable/disable the use of the function Port_SetPinDirection().

Table 4-4. Attribute PortSetPinDirectionApi (PortGeneral) detailed description

Property	Value
Label	Port SetPinDirection Api
Type	BOOLEAN
Origin	AUTOSAR_ECUC
Symbolic Name	false
Default	true

4.3.3 PortSetPinModeApi (PortGeneral)

Pre-processor switch to enable/disable the use of the function Port_SetPinMode().

Table 4-5. Attribute PortSetPinModeApi (PortGeneral) detailed description

Property	Value
Label	Port SetPinMode Api
Type	BOOLEAN
Origin	AUTOSAR_ECUC
Symbolic Name	false
Default	true

4.3.4 PortVersionInfoApi (PortGeneral)

Pre-processor switch to enable/disable the API to read out the modules version information.

Table 4-6. Attribute PortVersionInfoApi (PortGeneral) detailed description

Property	Value
Label	Port VersionInfo Api
Type	BOOLEAN
Origin	AUTOSAR_ECUC
Symbolic Name	false
Default	true

4.3.5 PortSetPinModeDoesNotTouchGpioLevel (PortGeneral)

Pre-processor switch. When not checked, the function Port_SetPinMode() will set the output level of the pin to the value configured in the PortPinLevelValue combo when called at run time to change mode of a pin from alternate function to GPIO. When checked, the function Port_SetPinMode() will not touch the output level of the pin when called at run time to change mode of a pin from alternate function to GPIO

Table 4-7. Attribute PortSetPinModeDoesNotTouchGpioLevel (PortGeneral) detailed description

Property	Value
Label	Port VersionInfo Api
Type	BOOLEAN
Origin	NXP
Symbolic Name	false
Default	false

4.3.6 PortEnableUserModeSupport (PortGeneral)

When this parameter is enabled, the Port module will adapt to run from User Mode, configuring REG_PROT for SIUL2 IP so that the registers under protection can be accessed from user mode by setting UAA bit in REG_PROT_GCR to 1. For more information, please see chapter 'User Mode Support' in IM. Note: Implementation Specific Parameter.

Table 4-8. Attribute PortEnableUserModeSupport (PortGeneral) detailed description

Property	Value
Label	Enable Port User Mode Support
Type	BOOLEAN
Origin	NXP
Symbolic Name	false
Default	false

4.4 Form CommonPublishedInformation

Common container, aggregated by all modules. It contains published information about vendor and versions.

CommonPublishedInformation	
Name	Value
AUTOSAR Major Version	4
AUTOSAR Minor Version	2
AUTOSAR Revision Version	2
Module Id.	124
Software Major Version	1
Software Minor Version	0
Software Patch Version	0
VendorApiInfix	
VendorId	43

Figure 4-3. Tresos Plugin snapshot for CommonPublishedInformation form.

4.4.1 ArReleaseMajorVersion (CommonPublishedInformation)

Major version number of AUTOSAR specification on which the appropriate implementation is based on.

Table 4-9. Attribute ArReleaseMajorVersion (CommonPublishedInformation) detailed description

Property	Value
Label	AUTOSAR Major Version
Type	INTEGER_LABEL
Origin	Custom
Symbolic Name	false
Default	4
Invalid	Range <div>>=4</div> <div><=4</div>

4.4.2 ArReleaseMinorVersion (CommonPublishedInformation)

Minor version number of AUTOSAR specification on which the appropriate implementation is based on.

Table 4-10. Attribute ArReleaseMinorVersion (CommonPublishedInformation) detailed description

Property	Value
Label	AUTOSAR Minor Version
Type	INTEGER_LABEL
Origin	Custom
Symbolic Name	false
Default	2
Invalid	Range <div> <div>>=2</div> <div><=2</div> </div>

4.4.3 ArReleaseRevisionVersion (CommonPublishedInformation)

Revision version number of AUTOSAR specification on which the appropriate implementation is based on.

Table 4-11. Attribute ArReleaseRevisionVersion (CommonPublishedInformation) detailed description

Property	Value
Label	AUTOSAR Release Revision Version
Type	INTEGER_LABEL
Origin	Custom
Symbolic Name	false
Default	2
Invalid	Range <div> <div>>=2</div> <div><=2</div> </div>

4.4.4 ModuleId (CommonPublishedInformation)

Module ID of this module from Module List.

Table 4-12. Attribute ModuleId (CommonPublishedInformation) detailed description

Property	Value
Label	Module Id
Type	INTEGER_LABEL
Origin	Custom
Symbolic Name	false
Default	124
Invalid	Range <div> <div>>=124</div> <div><=124</div> </div>

4.4.5 SwMajorVersion (CommonPublishedInformation)

Major version number of the vendor specific implementation of the module. The numbering is vendor specific.

Table 4-13. Attribute SwMajorVersion (CommonPublishedInformation) detailed description

Property	Value
Label	Software Major Version
Type	INTEGER_LABEL
Origin	Custom
Symbolic Name	false
Default	1
Invalid	Range <div> <div>>=1</div> <div><=1</div> </div>

4.4.6 SwMinorVersion (CommonPublishedInformation)

Minor version number of the vendor specific implementation of the module. The numbering is vendor specific.

Table 4-14. Attribute SwMinorVersion (CommonPublishedInformation) detailed description

Property	Value
Label	Software Minor Version
Type	INTEGER_LABEL
Origin	Custom
Symbolic Name	false

Table continues on the next page...

Table 4-14. Attribute SwMinorVersion (CommonPublishedInformation) detailed description (continued)

Property	Value
Default	0
Invalid	Range >=0 <=0

4.4.7 SwPatchVersion (CommonPublishedInformation)

Patch level version number of the vendor specific implementation of the module. The numbering is vendor specific.

Table 4-15. Attribute SwPatchVersion (CommonPublishedInformation) detailed description

Property	Value
Label	Software Patch Version
Type	INTEGER_LABEL
Origin	Custom
Symbolic Name	false
Default	0
Invalid	Range >=0 <=0

4.4.8 VendorApiInfix (CommonPublishedInformation)

In driver modules which can be instantiated several times on a single ECU, BSW00347 requires that the name of APIs is extended by the VendorId and a vendor specific name. This parameter is used to specify the vendor specific name. In total, the implementation specific name is generated as follows:

<ModuleName>_>VendorId>_<VendorApiInfix><Api name from SWS>. E.g. assuming that the VendorId of the implementor is 123 and the implementer chose a VendorApiInfix of "v11r456" a api name Can_Write defined in the SWS will translate to Can_123_v11r456Write. This parameter is mandatory for all modules with upper multiplicity > 1. It shall not be used for modules with upper multiplicity =1.

Table 4-16. Attribute VendorApiInfix (CommonPublishedInformation) detailed description

Property	Value
Label	Vendor Api Infix
Type	STRING_LABEL
Origin	Custom
Symbolic Name	false
Default	
Enable	false

4.4.9 VendorId (CommonPublishedInformation)

Vendor ID of the dedicated implementation of this module according to the AUTOSAR vendor list.

Table 4-17. Attribute VendorId (CommonPublishedInformation) detailed description

Property	Value
Label	Vendor Id
Type	INTEGER_LABEL
Origin	Custom
Symbolic Name	false
Default	43
Invalid	Range >=43 <=43

4.5 PortConfigSet

This container contains a configuration of the PORT driver / SIUL2 module. This container is a MultipleConfigurationContainer, i.e. this container and its subcontainers exit once per configuration set. It contains 2 sub-containers: General, for the notused pins configuration, and PortContainer for the used pins configurations. NOTE:"User should configure only ONE instance of PortConfigSet container and not MULTIPLE."

Includes:

- [PortContainer](#)
- [NotUsedPortPin](#)

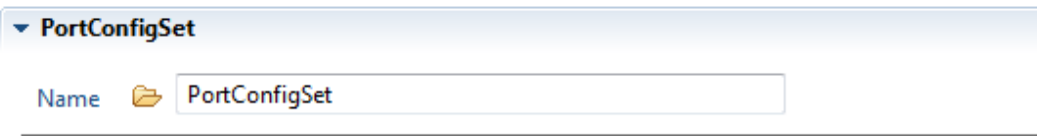


Figure 4-4. Tresos Plugin snapshot for PortConfigSet.

4.5.1 PortContainer

Container collecting the PortPins.

Is included by : PortConfigSet

Includes :

- PortPin

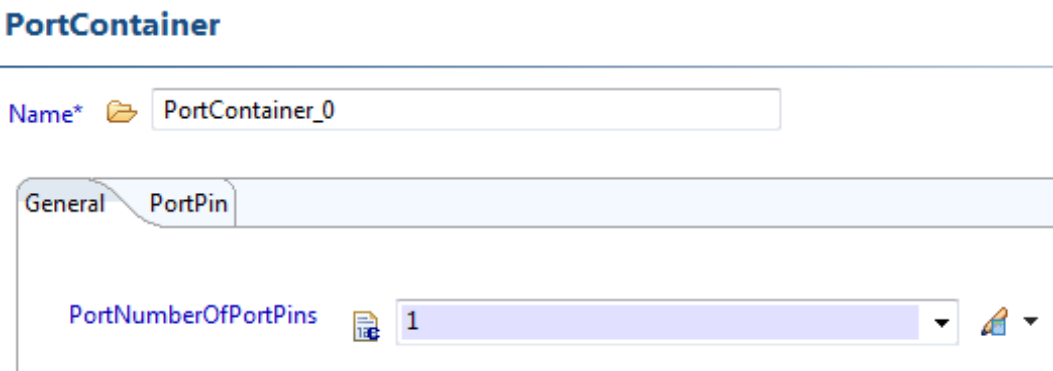


Figure 4-5. Tresos Plugin snapshot for PortContainer form.

4.5.1.1 PortNumberOfPortPins (PortContainer)

The number of specified PortPins in this PortContainer.

Table 4-18. Attribute PortNumberOfPortPins (PortContainer) detailed description

Property	Value
Label	PortNumberOfPortPins
Type	INTEGER
Origin	AUTOSAR_ECUC
Symbolic Name	false
Invalid	Range >=1 <=199

4.5.1.2 PortPin

Configuration of the individual port pins.

Is included by : [PortContainer](#)

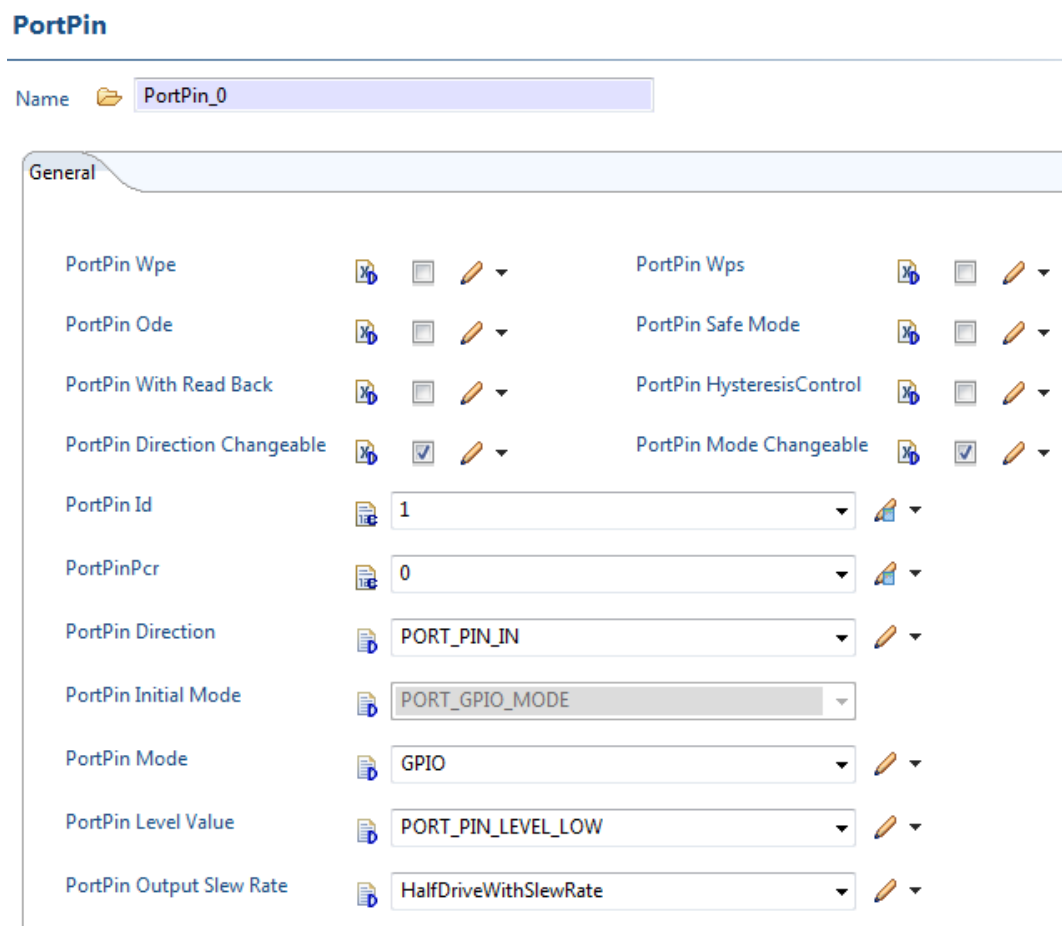


Figure 4-6. Tresos Plugin snapshot for PortPin

4.5.1.2.1 PortPinWpe (PortPin)

Enable Weak Pull Up/Down for the configured Pin. Checked box means any of the Weak Pull Up or Weak Pull Down configurations is enabled for the pin. This is an implementation specific parameter.

Table 4-19. Attribute PortPinWpe (PortPin) detailed description

Property	Value
Label	PortPin Wpe
Type	BOOLEAN

Table continues on the next page...

Table 4-19. Attribute PortPinWpe (PortPin) detailed description (continued)

Property	Value
Origin	Custom
Symbolic Name	false
Default	false

4.5.1.2.2 PortPinWps (PortPin)

Select Weak Pull Up/Down for the configured Pin. Checked box means the Weak Pull Up configuration is set. Unchecked box means the Weak Pull Down configuration is set. This is an implementation specific parameter.

Table 4-20. Attribute PortPinWps (PortPin) detailed description

Property	Value
Label	PortPin Wps
Type	BOOLEAN
Origin	Custom
Symbolic Name	false
Default	false

4.5.1.2.3 PortPinOde (PortPin)

Enable Open Drain Output for the configured Pin. Checked box means the Open Drain configuration is set. This is an implementation specific parameter.

Table 4-21. Attribute PortPinOde (PortPin) detailed description

Property	Value
Label	PortPin Ode
Type	BOOLEAN
Origin	Custom
Symbolic Name	false
Default	false

4.5.1.2.4 PortPin SafeMode (PortPin)

Enable SafeMode capability for the configured Pin. This is an implementation specific parameter.

Table 4-22. Attribute PortPin Safe Mode (PortPin) detailed description

Property	Value
Label	PortPin Safe mode
Type	BOOLEAN
Origin	Custom
Symbolic Name	false
Default	false

4.5.1.2.5 PortPinWithReadback(PortPin)

Enable Readback capability for the configured Pin. This is an implementation specific parameter.

Table 4-23. Attribute PortPinWithReadback(PortPin) detailed description

Property	Value
Label	PortPin Readback
Type	BOOLEAN
Origin	Custom
Symbolic Name	false
Default	false

4.5.1.2.6 PortPin Hysteresis Control(PortPin)

Enable/Disable Input Hysteresis for the configured Pin. This is an implementation specific parameter.

Table 4-24. Attribute PortPin Hysteresis Control(PortPin) detailed description

Property	Value
Label	PortPin Input Hysteresis
Type	BOOLEAN
Origin	Custom
Symbolic Name	false
Default	false

4.5.1.2.7 PortPinDirectionChangeable (PortPin)

Enable/Disable the changeability for the configured Pin. Checked box means the Direction Changeability is enabled. This is an implementation specific parameter.

Table 4-25. Attribute PortPinDirectionChangeable (PortPin) detailed description

Property	Value
Label	PortPin Direction Changeable
Type	BOOLEAN
Origin	AUTOSAR_ECUC
Symbolic Name	false
Default	true

4.5.1.2.8 PortPinModeChangeable (PortPin)

Parameter to indicate if the mode of a port pin is changeable during runtime. True: Port Pin mode changeable allowed. False: Port Pin mode changeable not permitted

Table 4-26. Attribute PortPinModeChangeable (PortPin) detailed description

Property	Value
Label	PortPin Mode Changeable
Type	BOOLEAN
Origin	AUTOSAR_ECUC
Symbolic Name	false
Default	true

4.5.1.2.9 PortPinId (PortPin)

Pin Id of the port pin. This value will be assigned to the symbolic name derived from the port pin container short name.

Table 4-27. Attribute PortPinId (PortPin) detailed description

Property	Value
Label	PortPin Id
Type	INTEGER
Origin	AUTOSAR_ECUC
Symbolic Name	true
Invalid	Range >=1 <=264

4.5.1.2.10 PortPinPcr (PortPin)

Pin Id of the port pin. This value will be assigned to the symbolic name derived from the port pin container short name. Used to specify the PCR (Port Configuration Register) for the configured pin.

Table 4-28. Attribute PortPinPcr (PortPin) detailed description

Property	Value
Label	PortPinPcr
Origin	Custom
Symbolic Name	false
Invalid	Range ≥ 0 ≤ 263

4.5.1.2.11 PortPinDirection (PortPin)

Selects the initial direction of the pin (IN or OUT). If the direction is not changeable, the value configured here is fixed. The pin direction can be set only for the GPIO pins. For the Alternative Function modes the OUT pin direction is hw selected. If the IN direction is needed too, it can be set at runtime. NOTE: To set the IN direction take care, please, that all the possible module inputs, possible as Alternative Functions for the pad mode, are hw connected together, if IN direction is enabled, to the pad.

Table 4-29. Attribute PortPinDirection (PortPin) detailed description

Property	Value
Label	PortPin Direction
Type	ENUMERATION
Origin	AUTOSAR_ECUC
Symbolic Name	false
Default	PORT_PIN_IN
Range	PORT_PIN_IN PORT_PIN_OUT PORT_PIN_INOUT

4.5.1.2.12 PortPinInitialMode (PortPin)

Port pin mode from mode list for use with Port_Init() function. NOTE: This parameter is not used in the current implementation and is retained as per std AUTOSAR_EcucParamDef.arxml file.

Table 4-30. Attribute PortPinInitialMode (PortPin) detailed description

Property	Value
Label	PortPin Initial Mode
Type	ENUMERATION
Origin	AUTOSAR_ECUC
Symbolic Name	false
Default	PORT_GPIO_MODE
Enable	false
Range	PORT_GPIO_MODE PORT_ALT1_FUNC_MODE PORT_ALT2_FUNC_MODE PORT_ALT3_FUNC_MODE PORT_ALT4_FUNC_MODE

4.5.1.2.13 PortPinMode (PortPin)

Selects the PORT pin mode from the modes list. For the Alternative Function modes (not a GPIO mode) the OUT direction is hw selected for that pin. NOTE: To set the IN direction take care, please, that all the possible module inputs, possible as Alternative Functions for the pad mode, are hw connected together, if IN direction is enabled, to the pad.

Table 4-31. Attribute PortPinMode (PortPin) detailed description

Property	Value
Label	PortPin Mode
Type	ENUMERATION
Origin	AUTOSAR_ECUC
Symbolic Name	false
Default	GPIO

4.5.1.2.14 PortPinLevelValue (PortPin)

Port Pin Level value from Port pin list.

Table 4-32. Attribute PortPinLevelValue (PortPin) detailed description

Property	Value
Label	PortPin Level Value
Type	ENUMERATION
Origin	AUTOSAR_ECUC
Symbolic Name	false

Table continues on the next page...

Table 4-32. Attribute PortPinLevelValue (PortPin) detailed description (continued)

Property	Value
Default	PORT_PIN_LEVEL_LOW
Range	PORT_PIN_LEVEL_HIGH PORT_PIN_LEVEL_LOW

4.5.1.2.15 PortPinSlewRate (PortPin)

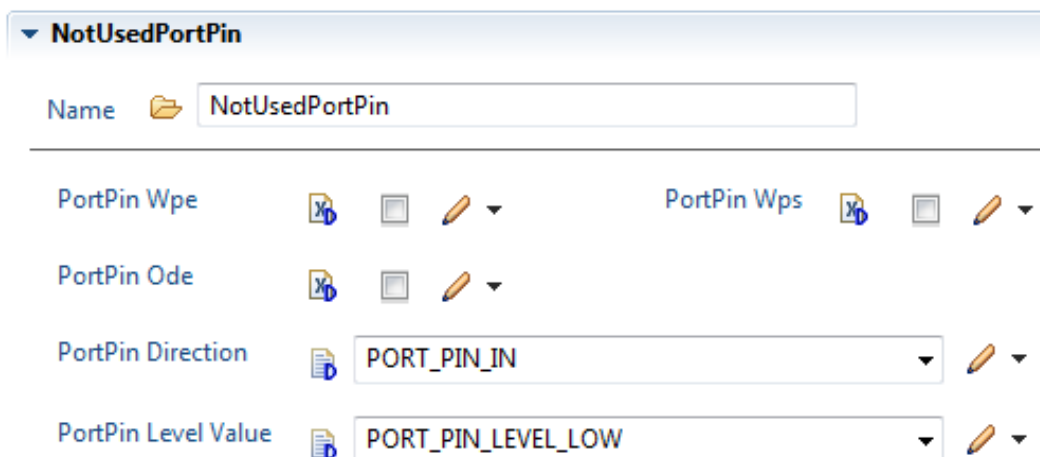
Configure Slew Rate for the configured Pin. This is an implementation specific parameter.

Table 4-33. Attribute PortPinSlewRate (PortPin) detailed description

Property	Value
Label	PortPin Output Slew Rate
Origin	Custom
Symbolic Name	false
Default	HalfDriveWithSlewRate
Range	HalfDriveWithSlewRate FullDriveWithSlewRate HalfDriveWithoutSlewRate FullDriveWithoutSlewRate

4.5.1.3 NotUsedPortPin

Module wide configuration parameters of the PORT driver.

**Figure 4-7. Tresos Plugin snapshot for NotUsedPortPin.**

4.5.1.3.1 PortPinWpe (NotUsedPortPin)

Enable Weak Pull Up/Down for the configured Pin. Checked box means any of the Weak Pull Up or Weak Pull Down configurations is enabled for the pin. This is an implementation specific parameter.

Table 4-34. Attribute PortPinWpe (NotUsedPortPin) detailed description

Property	Value
Label	PortPin Wpe
Origin	Custom
Symbolic Name	false
Default	false

4.5.1.3.2 PortPinWps (NotUsedPortPin)

Select Weak Pull Up/Down for the configured Pin. Checked box means the Weak Pull Up configuration is set. Unchecked box means the Weak Pull Down configuration is set. This is an implementation specific parameter.

Table 4-35. Attribute PortPinWps (NotUsedPortPin) detailed description

Property	Value
Label	PortPin Wps
Origin	Custom
Symbolic Name	false
Default	false

4.5.1.3.3 PortPin Ode (NotUsedPortPin)

Select Open Drain configuration for the unused pins. This is an implementation specific parameter.

Table 4-36. Attribute PortPin Ode (NotUsedPortPin) detailed description

Property	Value
Label	PortPin Ode
Origin	Custom
Symbolic Name	false
Default	false

4.5.1.3.4 PortPinDirection (NotUsedPortPin)

Configure DIRECTION for the configured Pin.

Table 4-37. Attribute PortPinDirection(NotUsedPortPin) detailed description

Property	Value
Label	PortPin Direction
Origin	AUTOSAR_ECUC
Symbolic Name	false
Default	PORT_PIN_DISABLED
Range	PORT_PIN_IN PORT_PIN_OUT PORT_PIN_DISABLED

4.5.1.3.5 PortPinLevelValue(NotUsedPortPin)

Port Pin Level value from Port pin list.

Table 4-38. Attribute PortPinLevelValue(NotUsedPortPin) detailed description

Property	Value
Label	PortPin Level Value
Type	ENUMERATION
Origin	AUTOSAR_ECUC
Symbolic Name	false
Default	PORT_PIN_LEVEL_LOW
Range	PORT_PIN_LEVEL_HIGH PORT_PIN_LEVEL_LOW

How to Reach Us:**Home Page:**nxp.com**Web Support:**nxp.com/support

Information in this document is provided solely to enable system and software implementers to use NXP products. There are no express or implied copyright licenses granted hereunder to design or fabricate any integrated circuits based on the information in this document. NXP reserves the right to make changes without further notice to any products herein.

NXP makes no warranty, representation, or guarantee regarding the suitability of its products for any particular purpose, nor does NXP assume any liability arising out of the application or use of any product or circuit, and specifically disclaims any and all liability, including without limitation consequential or incidental damages. "Typical" parameters that may be provided in NXP data sheets and/or specifications can and do vary in different applications, and actual performance may vary over time. All operating parameters, including "typicals," must be validated for each customer application by customer's technical experts. NXP does not convey any license under its patent rights nor the rights of others. NXP sells products pursuant to standard terms and conditions of sale, which can be found at the following address: nxp.com/SalesTermsandConditions.

NXP, the NXP logo, NXP SECURE CONNECTIONS FOR A SMARTER WORLD, COOLFLUX, EMBRACE, GREENCHIP, HITAG, I2C BUS, ICODE, JCOP, LIFE VIBES, MIFARE, MIFARE CLASSIC, MIFARE DESFire, MIFARE PLUS, MIFARE FLEX, MANTIS, MIFARE ULTRALIGHT, MIFARE4MOBILE, MIGLO, NTAG, ROADLINK, SMARTLX, SMARTMX, STARPLUG, TOPFET, TRENCHMOS, UCODE, Freescale, the Freescale logo, Altivec, C-5, CodeTest, CodeWarrior, ColdFire, ColdFire+, C-Ware, the Energy Efficient Solutions logo, Kinetis, Layerscape, MagniV, mobileGT, PEG, PowerQUICC, Processor Expert, QorIQ, QorIQ Qonverge, Ready Play, SafeAssure, the SafeAssure logo, StarCore, Symphony, VortiQa, Vybrid, Airfast, BeeKit, BeeStack, CoreNet, Flexis, MXC, Platform in a Package, QUICC Engine, SMARTMOS, Tower, TurboLink, and UMEMS are trademarks of NXP B.V. All other product or service names are the property of their respective owners. ARM, AMBA, ARM Powered, Artisan, Cortex, Jazelle, Keil, SecurCore, Thumb, TrustZone, and μ Vision are registered trademarks of ARM Limited (or its subsidiaries) in the EU and/or elsewhere. ARM7, ARM9, ARM11, big.LITTLE, CoreLink, CoreSight, DesignStart, Mali, mbed, NEON, POP, Sensinode, Socrates, ULINK and Versatile are trademarks of ARM Limited (or its subsidiaries) in the EU and/or elsewhere. All rights reserved. Oracle and Java are registered trademarks of Oracle and/or its affiliates. The Power Architecture and Power.org word marks and the Power and Power.org logos and related marks are trademarks and service marks licensed by Power.org.

© 2017 NXP B.V.

Document Number UM35PORTASR4.2 Rev002 R1.0.0
Revision 1.0