User Manual

for S32K1_S32M24X PORT Driver

Document Number: UM2PORTASRR21-11 Rev0000R2.0.0 Rev. 1.0

Chapter 1

Revision History

Revision	Date	Author	Description
1.0	04.08.2023	NXP RTD Team	S32K1_S32M24X Real-Time Drivers AUTOSAR 4.4 & R21-11 Version 2.0.0

Chapter 2

Introduction

This User Manual describes NXP Semiconductor AUTOSAR Port for S32K1_S32M24X. AUTOSAR Port driver configuration parameters and deviations from the specification are described in 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 Semiconductors:

- s32k116_qfn32
- s32k116_lqfp48
- s32k118_lqfp48
- s32k118_lqfp64
- s32k142_lqfp48
- s32k142_lqfp64
- s32k142_lqfp100
- s32k142w_lqfp48
- s32k142w_lqfp64
- s32k144_lqfp48

- s32k144_lqfp64 / MWCT1014S_lqfp64
- s32k144_lqfp100 / MWCT1014S_lqfp100
- s32k144_mapbga100
- s32k144w_lqfp48
- s32k144w_lqfp64
- s32k146_lqfp64
- s32k146_lqfp100 / MWCT1015S_lqfp100
- s32k146_mapbga100 / MWCT1015S_mapbga100
- s32k146_lqfp144
- s32k148_lqfp100
- s32k148_mapbga100 / MWCT1016S_mapbga100
- s32k148_lqfp144
- s32k148_lqfp176
- s32m241_lqfp64
- s32m242_lqfp64
- s32m243_lqfp64
- s32m244_lqfp64

All of the above microcontroller devices are collectively named as S32K1_S32M24X. Note: MWCT part numbers contain NXP confidential IP for Qi Wireless Power

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** style: Used for important terms, notes and warnings.
- *Italic* style: 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.

Warning

This is a warning

2.4 Acronyms and Definitions

Term	Definition
API	Application Programming Interface
ASM	Assembler
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 Signifigant 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
SWS	Software Specification
VLE	Variable Length Encoding
XML	Extensible Markup Language

2.5 Reference List

#	Title	Version
1	Specification of Port Driver	AUTOSAR Release R21-11
2	Reference Manual	S32K1xx Series Reference Manual, Rev. 14, 09/2021
		S32M24x Reference Manual, Rev. 2 Draft A, 05/2023
4	Errata	S32K116_0N96V Rev. 22/OCT/2021
		S32K118_0N97V Rev. 22/OCT/2021
		S32K142_0N33V Rev. 22/OCT/2021
		S32K144_0N57U Rev. 22/OCT/2021
		S32K144W_0P64A Rev. 22/OCT/2021
		S32K146_0N73V Rev. 22/OCT/2021
		S32K148_0N20V Rev. 22/OCT/2021
		S32M244_P64A+P73G, Rev. 0

#	Title	Version
		S32M242_N33V+P73G, Rev. 0, 6/2023
5	Datasheet	S32K1xx Data Sheet, Rev. 14, 08/2021
		S32M2xx Data Sheet, Supports S32M24x and S32M27x, Rev. 3 DraftA, 05/2023

Chapter 3

Driver

3.1 Requirements

Requirements for this driver are detailed in the Autosar 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 PORT (Port Control and Interrupts).

Every PortPin configured in a PortContainer of the Port plugin can be mapped to one and only one microcontroller pin. The following steps must be followed in order to correctly map a Port plugin pin over a specific microcontroller pin:

- 1. Open the IOMUX Excel file attached to the Reference Manual
- 2. Go to 'IO Signal Table' sheet
- 3. Identify the microcontroller pin you want to use (eg. PTC7), searching after the values in columns 'Module' and 'Function'. Scroll to the Excel row where the pin's name appear first in column 'Port'. On S32K1_↩ S32M24X platform, we have 32 pins per port. So 71 is the number which represents the numeric value of the Multiplexed Signal Configuration Register. Note down this number (71).
- 4. Go to port container inside the Port plugin where you want to add the pin
- 5. Add a new PortPin in the port container list then double click the newly added PortPin to open it's properties
- 6. Go to the 'PortPin MSCR' attribute and type the number noted down at step 3
- 7. Go to the 'PortPin Mode' attribute and choose the functionality you want to use for the selected pin

3.4 Deviations from Requirements

The driver deviates from the AUTOSAR Port Driver software specification in some places. The table identifies the AUTOSAR requirements that are not fully implemented, not implemented or out of scope for the Port Driver.

Term	Definition
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, not implemented or out of scope for the driver.

Requirement	Status	Description	Notes
SWS_Port_00220	N/S	The type Port_PinDirectionType shall be of enumeration type having range as PORT_PIN_IN and PORT_PIN↩ _OUT.	Replaced by CPR_RTD_00026.port

Requirement	Status	Description	Notes
SWS_Port_00227	N/S	These requirements are not applicable to this specification. (SRS_BSW_00005, SRS_BSW_00006, SRS_BSW_00007, SRS_BSW_00010, SRS_BSW_00160, SRS_BSW_00161, SRS_BSW_00162, SRS_BSW_00164, SRS_BSW_00167, SRS_BSW_00168, SRS_BSW_00170, SRS_BSW_00172, SRS_BSW_00307, SRS_BSW_00308, SRS_BSW_00309, SRS_BSW_00321, SRS_BSW_00325, SRS_BSW_00328, SRS_BSW_00330, SRS_BSW_00331, SRS_BSW_00333, SRS_BSW_00334, SRS_BSW_00335, SRS_BSW_00336, SRS_BSW_00341, SRS_BSW_00342, SRS_BSW_00343, SRS_BSW_00344, SRS_BSW_00347, SRS_BSW_00357, SRS_BSW_00359, SRS_BSW_00360, SRS_SPAL_12463, SRS_SPAL_12462, SRS_SPAL_12265, SRS_SPAL_12092, SRS_SPAL_12078, SRS_SPAL_12077, SRS_SPAL_12067, SRS_SPAL_12064, SRS_SPAL_12129, SRS_SPAL_12075, SRS_SPAL_12063, SRS_SPAL_12169, SRS_SPAL_00157, SRS_SPAL_12069, SRS_SPAL_12068, SRS_SPAL_12267, SRS_SPAL_12056, SRS_BSW_00440, SRS_BSW_00439, SRS_BSW_00437, SRS_BSW_00433, SRS_BSW_00432, SRS_BSW_00429, SRS_BSW_00428, SRS_BSW_00427, SRS_BSW_00426, SRS_BSW_00425, SRS_BSW_00424, SRS_BSW_00423, SRS_BSW_00419, SRS_BSW_00417, SRS_BSW_00416, SRS_BSW_00413, SRS_BSW_00398, SRS_BSW_00395, SRS_BSW_00378, SRS_BSW_00377, SRS_BSW_00375, SRS_BSW_00373, SRS_BSW_00371)	This is not a requirement

Requirement	Status	Description	Notes
ECUC_Port_00128	N/S	"Name - PortPinInitialMode - Parent Container - PortPin - Description - Port pin mode from mode list for use with Port_Init() function. - Multiplicity - 1 - Type - EcucEnumeration↩ParamDef - Range - PORT_PIN↩_MODE_ADC - Port Pin used by ADC - PORT_PIN_MODE_CAN - Port Pin used for CAN - PORT_↩PIN_MODE_DIO - Port Pin configured for DIO. It shall be used under control of the DIO driver. - PORT↩_PIN_MODE_DIO_GPT - Port Pin configured for DIO. It shall be used under control of the general purpose timer driver. - PORT_PIN_MOD↩E_DIO_WDG - Port Pin configured for DIO. It shall be used under control of the watchdog driver. - PO↩RT_PIN_MODE_FLEXRAY - Port Pin used for FlexRay - PORT_PIN↩_MODE_ICU - Port Pin used by IC↩U - PORT_PIN_MODE_LIN - Port Pin used for LIN - PORT_PIN_M↩ODE_MEM - Port Pin used for external memory under control of a memory driver. - PORT_PIN_MODE_PW↩M - Port Pin used by PWM - POR↩T_PIN_MODE_SPI - Port Pin used by SPI - Post-Build Variant Value - true - Value Configuration Class - Pre-compile time - X - VARIANT-PRE-↩COMPILE - Link time - -- - - Post-build time - X - VARIANT-POST-B↩UILD - Scope / Dependency - scope: local - "	Currently implemented in a diffferent mode in MCAL 4.3.0. This requirement was replaced by requirement E↩CUC_Port_00130.

Requirement	Status	Description	Notes
ECUC_Port_00130	N/S	"Name - PortPinMode - Parent Container - PortPin - Description - Port pin mode from mode list. Note that more than one mode is allowed by default. That way it is e.g. possible to combine DIO with another mode such as ICU. - Multiplicity - 1..∗ - Type - EcucEnumerationParamDef - Range - PORT_PIN_MODE_ADC - Port Pin used by ADC - PORT_PIN_M↩ODE_CAN - Port Pin used for CA↩N - PORT_PIN_MODE_DIO - Port Pin configured for DIO. It shall be used under control of the DIO driver. - PORT_PIN_MODE_DIO_GPT - Port Pin configured for DIO. It shall be used under control of the general purpose timer driver. - PORT_PIN↩_MODE_DIO_WDG - Port Pin configured for DIO. It shall be used under control of the watchdog driver. - PO↩RT_PIN_MODE_FLEXRAY - Port Pin used for FlexRay - PORT_PIN↩_MODE_ICU - Port Pin used by IC↩U - PORT_PIN_MODE_LIN - Port Pin used for LIN - PORT_PIN_M↩ODE_MEM - Port Pin used for external memory under control of a memory driver. - PORT_PIN_MODE_PWM - Port Pin used by PWM - PORT_↩PIN_MODE_SPI - Port Pin used by SPI - Post-Build Variant Multiplicity - true - Post-Build Variant Value - true - Multiplicity Configuration Class - Pre-compile time - X - VARIANT-PRE-↩COMPILE - Link time - -- - - Post-build time - X - VARIANT-POST-↩BUILD - Value Configuration Class - Pre-compile time - X - VARIANT-↩PRE-COMPILE - Link time - -- - - Post-build time - X - VARIANT-P↩OST-BUILD - Scope / Dependency - scope: local - "	Replaced by requirement CPR_RT↩D_00372.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 Driver Limitations

- VSMD report for PORT has some errors: due to ECUC_PORT_00130 requirement is no longer apply on RTD product (AAI-192).
- ECPD import issue: Only one container of PortPin list for importing ECPD is currently supported. When importing an ecpd/epc file that has two or more containers of PortContainer from EB Tresos to Design Studio (S32DS), the S32DS framework will create 2 different Pins Tool functional groups with naming joined by underscores between Port container and BOARD_InitPeripherals. It leads naming of the Pins Tool functional group was changed with S32DS original configuration, so the Port HLD cannot get configuration from the Pins Tool.

3.6 Driver usage and configuration tips

The Port driver is responsible with configuring the functionality that should be active on a platform hardware pin. The information about the functionalities available on each of the hardware pins of the platform can be found in the IO muxing table Excel file attached to the Reference Manual pdf.

The Port plugin allows the user to configure each pin's functionality using 3 distinct mechanisms:

- A. Define the functionality of a specific pin. This can be done by adding a new entry in the PortContainer/↩ PortPin list and setting the attributes of the pin. The following steps should be followed:
  - 1. Open the IOSignal description Excel file
  - 2. Go to 'IO Signal Table' sheet
  - 3. Identify the microcontroller pin you want to use (eg. PTC7), searching after the values in columns 'Module' and 'Function'. Scroll to the Excel row where the pin's name appear first in column 'Port'. On S32K1_S32M24X platform, we have 32 pins per port.
    Value of actual PortPin Pcr is identified by formula:
    PortPinPcr = PinId + PortId∗32
    Where:
    - PortId is the numeric identifier of the port. Symbolic names will be generated for each port pin id for the pins which being used for configuration.

Port Name	A	B	C	D	E
PortId	0	1	2	3	4

    - PinId is in order of selected pin in the port. On S32K1_S32M24X platform, we have 32 pins per port, so the range of PortPinId is: 0-31.
    - So PortPinPcr (eg. PTC7) = 7 + 2∗32 = 71 is the number which represents the numeric value of the Port Configuration Register. Note down this number (71).
  - 4. Go to port container inside the Port plugin where you want to add the pin
  - 5. Add a new PortPin in the port container list then double click the newly added PortPin to open it's properties
  - 6. Go to the 'PortPin MSCR' attribute and type the number noted down at step A.3
  - 7. Go to the 'PortPin Mode' attribute and choose the functionality you want to use for the selected pin
  - 8. Look at the other attributes of the PortPin and set them to the desired values

- B. Define pins that should not be touched by any Port driver functionality, including Port_Init() function. This option allows the user to configure a list of pins for which the driver will not touch their PCRs, leaving them containing the reset values. This list is named UnTouchedPortPin and is available in the PortConfigSet container and adding new entries in this list should follow the next steps:
  - 1. Open the IOSignal description Excel file
  - 2. Go to 'IO Signal Table' sheet
  - 3. Identify the microcontroller pin you want the Port driver to not touch (eg. PTA4), searching after the values in columns 'Module' and 'Function'. Scroll to the Excel row where the pin's name appear first in column 'Port'. On S32K1_S32M24X platform, we have 32 pins per port. So 4 is the number which represents the numeric value of the Port Configuration Register. Note down this number (4).
  - 4. Go to UnTouchedPortPin list inside the PortConfigSet container
  - 5. Add a new entry in the list and double click it to open it's properties
  - 6. Go to the 'PortPin MSCR' attribute and type the number noted down at step A.3
- C. Define the settings for all platform hardware pins that were not configured using mechanism described at point A and point B. This option allows the user to configure all platform pins that are not explicitly configured by the user (point A) or not left untouched (point B) as GPIOs, with some specific settings. These settings are available in the container NotUsedPortPin where the user can define the pin direction (in or out), pin level (high or low), pull up/down.

Every single platform hardware pin is configured by the Port driver, either by mechanism A, mechanism B or mechanism C.

For this reason, if the platform contains hardware pins that need to have certain non GPIO functionalities, these pins must be explicitly added in the Port configuration using mechanism A or B. Otherwise, they will be configured by Port_Init() API as GPIOs.

Design Studio

- According to new Framework of Design Studio, Port HLD with get the configuration from the Pins tool, it means the Pins tool should be always enabled. Please follow these steps:
  - 1. Use Pins tool to configure a signal, for examples: PTE6, mode port, 6.
  - 2. Refer to the IOMux to get the MSCR register corresponding to PTE6: PCR 134 (S32K148).
  - 3. Open Peripheral tool, add Port component, add value PortPin Pcr = 134.
  - 4. With Boolean nodes, an error shall be throwed when the configuration does not match between Port HLD and Pins tool.

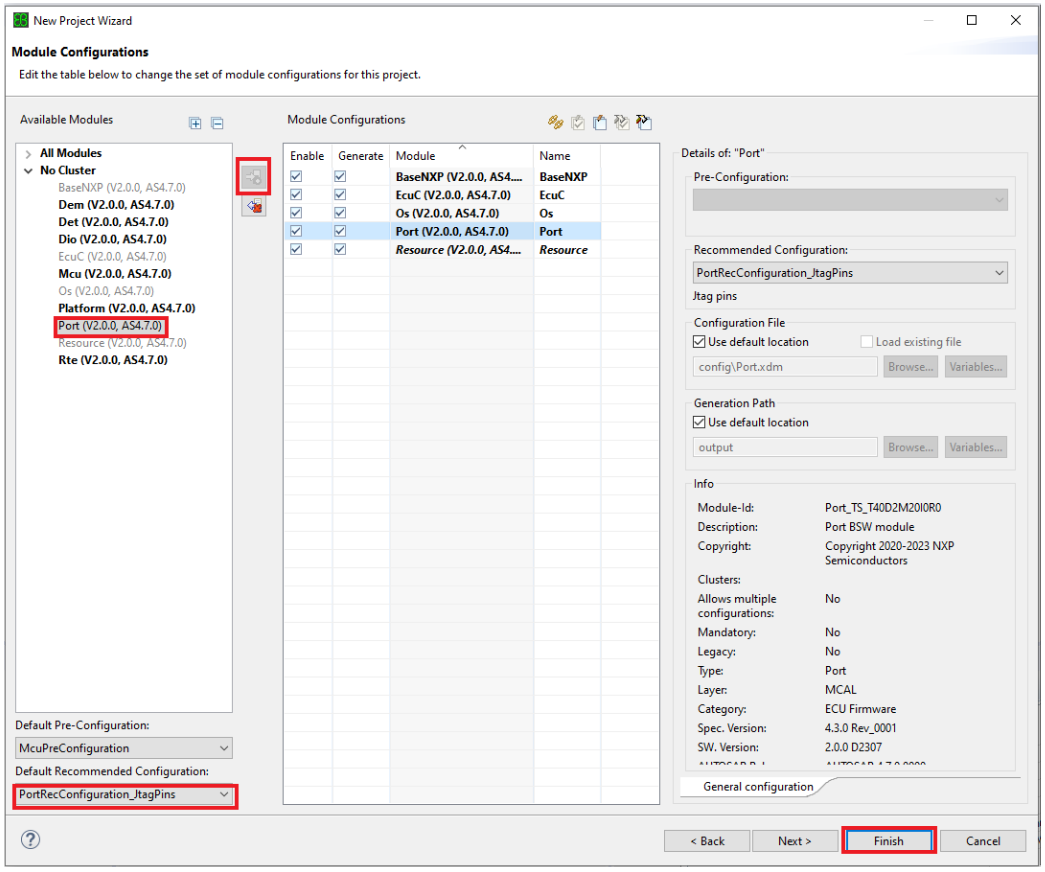
Important note

In order to be able to use the debug capabilities, the JTAG pins need to be configured in the Port driver using mechanism B. This means that the following pins/functionalities need to be added in the UnTouchedPortPin list:

- JTAG_TDI having PortPinPcr set to 69
- JTAG_TDO having PortPinPcr set to 10
- JTAG_TCK having PortPinPcr set to 68
- JTAG_TMS having PortPinPcr set to 4

- Reset_b having PortPinPcr set to 5

The Jtag pins can be automatically added in the Port driver configuration if when adding Port plugin in the Tresos project, the user selects the Default recommended configuration as: PortRecConfiguration_JtagPins.



New Project Wizard

Module Configurations

Edit the table below to change the set of module configurations for this project.

Available Modules

- All Modules
- No Cluster
  - BaseNXP (V2.0.0, AS4.7.0)
  - Dem (V2.0.0, AS4.7.0)
  - Det (V2.0.0, AS4.7.0)
  - Dio (V2.0.0, AS4.7.0)
  - EcuC (V2.0.0, AS4.7.0)
  - Mcu (V2.0.0, AS4.7.0)
  - Os (V2.0.0, AS4.7.0)
  - Platform (V2.0.0, AS4.7.0)
  - Port (V2.0.0, AS4.7.0)
  - Resource (V2.0.0, AS4.7.0)
  - Rte (V2.0.0, AS4.7.0)

Default Pre-Configuration: McuPreConfiguration

Default Recommended Configuration: PortRecConfiguration_JtagPins

Module Configurations

Enable	Generate	Module	Name
✓	✓	BaseNXP (V2.0.0, AS4....	BaseNXP
✓	✓	EcuC (V2.0.0, AS4.7.0)	EcuC
✓	✓	Os (V2.0.0, AS4.7.0)	Os
✓	✓	Port (V2.0.0, AS4.7.0)	Port
✓	✓	Resource (V2.0.0, AS4....	Resource

Details of: "Port"

Pre-Configuration:

Recommended Configuration: PortRecConfiguration_JtagPins

Jtag pins

Configuration File

☑ Use default location ☐ Load existing file

config\Port.xdm

Generation Path

☑ Use default location

output

Info

Module-Id:	Port_TS_T40D2M20I0R0
Description:	Port BSW module
Copyright:	Copyright 2020-2023 NXP Semiconductors
Clusters:	
Allows multiple configurations:	No
Mandatory:	No
Legacy:	No
Type:	Port
Layer:	MCAL
Category:	ECU Firmware
Spec. Version:	4.3.0 Rev_0001
SW. Version:	2.0.0 D2307

Figure 3.1 How to configure JTAG pins

Autosar extension functionality

- Support to run driver's code from User Mode. This option is configurable on/off per entire driver, using the checkbox 'Enable Port User Mode Support' in PortGeneral container. When this parameter is enabled, the Port module will adapt to run from user mode so that the registers under protection can be accessed from user mode. For more information, please see the IM chapter 'User Mode Support'.
- Port SetPinMode Does Not Touch GPIO Levels. This option is configurable on/off and it affects the functionality of the Port_SetPinMode() API. 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.

ADC Interleave functionality

- Only standalone ip-layer is supported for interleave feature.
- When using interleave, please enable the user mode support in the configuration.
- In order to use the interleave, choose only the ADCx which is connected to that pin via SIM module (please refer to information tooltip on the respective interleave-enabled pins in S32CT Pins Tool).

3.7 Runtime errors

The driver generates the following DET errors at runtime.

Function	Error Code	Condition triggering the error
Port_SetPinDirection	PORT_E_PARAM_PIN	Invalid Port Pin ID requested
Port_SetPinMode	PORT_E_PARAM_PIN	Invalid Port Pin ID requested
Port_Set2PinsDirection	PORT_E_PARAM_PIN	Invalid Port Pin ID requested
Port_SetPinDirection	PORT_E_DIRECTION_UNCHAN↩ GEABLE	Port Pin not configured as changeable
Port_Set2PinsDirection	PORT_E_DIRECTION_UNCHAN↩ GEABLE	Port Pin not configured as changeable
Port_SetPinMode	PORT_E_MODE_UNCHANGEAB↩ LE	API Port_SetPinMode service called when mode is unchangeable.
Port_Init	PORT_E_INIT_FAILED	API Port_Init service called with wrong parameter.
Port_SetPinMode	PORT_E_PARAM_INVALID_MO↩ DE	API Port_SetPinMode service called when mode is unchangeable.
Port_SetPinDirection	PORT_E_UNINIT	API service called without module initialization
Port_SetPinMode	PORT_E_UNINIT	API service called without module initialization
Port_RefreshPortDirection	PORT_E_UNINIT	API service called without module initialization
Port_Set2PinsDirection	PORT_E_UNINIT	API service called without module initialization
Port_GetVersionInfo	PORT_E_PARAM_POINTER	APIs called with a Null Pointer

3.8 Symbolic Names Disclaimer

All containers having symbolicNameValue set to TRUE in the AUTOSAR schema will generate defines like:

#define <Mip>Conf_<Container_ShortName>_<Container_ID>

For this reason it is forbidden to duplicate the names of such containers across the RTD configurations 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 driver. All the parameters are described below.

- Module Port
  - Container PortConfigSet
    - Container NotUsedPortPin
      - Parameter PortPinMode
      - Parameter PortPinDirection
      - Parameter PortPinLevelValue
      - Parameter PortPinDSE
      - Parameter PortPinPE
      - Parameter PortPinPS
    - Container PortContainer
      - Parameter PortNumberOfPortPins
      - Container PortPin
      - Parameter PortPinPFE
      - Parameter PortPinODE
      - Parameter PortPinLK
      - Parameter PortPinDirectionChangeable
      - Parameter PortPinModeChangeable
      - Parameter PortPinId
      - Parameter PortPinPcr
      - Parameter PortPinMode
      - Parameter PortPinDSE
      - Parameter PortPinPE
      - Parameter PortPinPS
      - Parameter PortPinSlewRate
      - Parameter PortPinDirection
      - Parameter PortPinInitialMode
      - Parameter PortPinLevelValue
      - Reference PortPinEcucPartitionRef
    - Container UnTouchedPortPin
      - Parameter PortPinPcr

        * Container DigitalFilter
            · Parameter DigitalFilterPort
            · Parameter DigitalFilterClock
            · Parameter DigitalFilterWidth
            · Container DigitalFilterChannel
            · Parameter DigitalFilterChannelIndex
    - Container PortGeneral
        * Parameter PortDevErrorDetect
        * Parameter PortCiPortIPDevErrorDetect
        * Parameter PortSetPinDirectionApi
        * Parameter PortSet2PinsDirectionApi
        * Parameter PortSetPinModeApi
        * Parameter PortVersionInfoApi
        * Parameter PortSetPinModeDoesNotTouchGpioLevel
        * Parameter PortSetAsUnusedPinApi
        * Parameter PortResetPinModeApi
        * Parameter PortEnableUserModeSupport
        * Parameter PortMulticoreSupport
        * Reference PortEcucPartitionRef
    - Container CommonPublishedInformation
        * Parameter ArReleaseMajorVersion
        * Parameter ArReleaseMinorVersion
        * Parameter ArReleaseRevisionVersion
        * Parameter ModuleId
        * Parameter SwMajorVersion
        * Parameter SwMinorVersion
        * Parameter SwPatchVersion
        * Parameter VendorApiInfix
        * Parameter VendorId

4.1 Module Port

Configuration of the Port module.

Included containers:

- PortConfigSet
- PortGeneral
- CommonPublishedInformation

Property	Value
type	ECUC-MODULE-DEF
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantSupport	true
supportedConfigVariants	VARIANT-POST-BUILD, VARIANT-PRE-COMPILE

4.2 Container PortConfigSet

This container contains a configuration of the PORT driver.

Included subcontainers:

- NotUsedPortPin
- PortContainer
- UnTouchedPortPin
- DigitalFilter

Property	Value
type	ECUC-PARAM-CONF-CONTAINER-DEF
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A

4.3 Container NotUsedPortPin

The init parameters values for the not used pins in the PORT configuration.

Included subcontainers:

- None

Property	Value
type	ECUC-PARAM-CONF-CONTAINER-DEF
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A

4.4 Parameter PortPinMode

Selects if the unused port pins are configured as GPIOs or are disabled.

Property	Value
type	ECUC-ENUMERATION-PARAM-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	true
valueConfigClasses	VARIANT-POST-BUILD: POST-BUILD
	VARIANT-PRE-COMPILE: PRE-COMPILE
defaultValue	GPIO
literals	['GPIO', 'Disabled']

4.5 Parameter PortPinDirection

Selects the initial direction of the pin (IN or OUT or HIGH_Z). 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 HIGH_Z and 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.

Set input pin to high-Z not available in S32K11X

Property	Value
type	ECUC-ENUMERATION-PARAM-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	true
valueConfigClasses	VARIANT-POST-BUILD: POST-BUILD
	VARIANT-PRE-COMPILE: PRE-COMPILE

4.6 Parameter PortPinLevelValue

Port Pin Level value from Port pin list.

Property	Value
type	ECUC-ENUMERATION-PARAM-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	true
valueConfigClasses	VARIANT-POST-BUILD: POST-BUILD
	VARIANT-PRE-COMPILE: PRE-COMPILE
defaultValue	PORT_PIN_LEVEL_LOW
literals	['PORT_PIN_LEVEL_HIGH', 'PORT_PIN_LEVEL_LOW']

4.7 Parameter PortPinDSE

Selects the drive strenght length value for this pin.

Property	Value
type	ECUC-ENUMERATION-PARAM-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	true
valueConfigClasses	VARIANT-POST-BUILD: POST-BUILD
	VARIANT-PRE-COMPILE: PRE-COMPILE
defaultValue	Low_Drive_Strength
literals	['Low_Drive_Strength', 'High_Drive_Strength']

4.8 Parameter PortPinPE

Selects if the pull-up or pull-down resistors are enabled.

Property	Value
type	ECUC-ENUMERATION-PARAM-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	true
valueConfigClasses	VARIANT-POST-BUILD: POST-BUILD
	VARIANT-PRE-COMPILE: PRE-COMPILE
defaultValue	PullDisabled
literals	['PullDisabled', 'PullEnabled']

4.9 Parameter PortPinPS

Selects between the pull-up and pull-down resistors. Only valid when PortPin PE is set to ’PullEnabled’.

Property	Value
type	ECUC-ENUMERATION-PARAM-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	true
valueConfigClasses	VARIANT-POST-BUILD: POST-BUILD
	VARIANT-PRE-COMPILE: PRE-COMPILE
defaultValue	PullDown
literals	['PullDown', 'PullUp']

4.10 Container PortContainer

Container collecting the PortPins.

Included subcontainers:

- PortPin

Property	Value
type	ECUC-PARAM-CONF-CONTAINER-DEF
lowerMultiplicity	1
upperMultiplicity	Infinite
postBuildVariantMultiplicity	false
multiplicityConfigClasses	VARIANT-POST-BUILD: PRE-COMPILE
	VARIANT-PRE-COMPILE: PRE-COMPILE

4.11 Parameter PortNumberOfPortPins

The number of specified PortPins in this PortContainer.

Property	Value
type	ECUC-INTEGER-PARAM-DEF
origin	AUTOSAR_ECUC
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	false
valueConfigClasses	VARIANT-POST-BUILD: PRE-COMPILE
	VARIANT-PRE-COMPILE: PRE-COMPILE
defaultValue	3
max	145
min	1

4.12 Container PortPin

Configuration of the individual port pins.

Included subcontainers:

- None

Property	Value
type	ECUC-PARAM-CONF-CONTAINER-DEF
lowerMultiplicity	1
upperMultiplicity	Infinite
postBuildVariantMultiplicity	false
multiplicityConfigClasses	VARIANT-POST-BUILD: PRE-COMPILE
	VARIANT-PRE-COMPILE: PRE-COMPILE

4.13 Parameter PortPinPFE

Passive Filter Enable

Passive filter configuration is valid in all digital pin muxing modes.

NotePORT_PCRn[PFE] is configurable for only PTA5 and PTD3. PFE for these should be configured in ALT7 mode only. For other modes, PFE should be kept 0.

Property	Value
type	ECUC-BOOLEAN-PARAM-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	true
valueConfigClasses	VARIANT-POST-BUILD: POST-BUILD
	VARIANT-PRE-COMPILE: PRE-COMPILE
defaultValue	false

4.14 Parameter PortPinODE

Enable Open Drain Output for the configured Pin.

Checked box means the Open Drain configuration is set.

This is an implementation specific parameter.

Property	Value
type	ECUC-BOOLEAN-PARAM-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	true
valueConfigClasses	VARIANT-POST-BUILD: POST-BUILD
	VARIANT-PRE-COMPILE: PRE-COMPILE
defaultValue	false

4.15 Parameter PortPinLK

Lock Register Enable

Pin Control Register fields [15:0] are locked and cannot be updated until the next system reset.

Property	Value
type	ECUC-BOOLEAN-PARAM-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	true
valueConfigClasses	VARIANT-POST-BUILD: POST-BUILD
	VARIANT-PRE-COMPILE: PRE-COMPILE
defaultValue	false

4.16 Parameter PortPinDirectionChangeable

Enable/Disable the changeability for the configured Pin. Checked box means the Direction Changeability is enabled.

This is an implementation specific parameter. The changeable pin direction can be set only for the GPIO pins.

For a mode different than GPIO, pin direction changeablity shall be disabled.

Property	Value
type	ECUC-BOOLEAN-PARAM-DEF
origin	AUTOSAR_ECUC
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	true
valueConfigClasses	VARIANT-POST-BUILD: POST-BUILD
	VARIANT-PRE-COMPILE: PRE-COMPILE
defaultValue	true

4.17 Parameter PortPinModeChangeable

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

Property	Value
type	ECUC-BOOLEAN-PARAM-DEF
origin	AUTOSAR_ECUC
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	true
valueConfigClasses	VARIANT-POST-BUILD: POST-BUILD
	VARIANT-PRE-COMPILE: PRE-COMPILE
defaultValue	true

4.18 Parameter PortPinId

Pin Id of the port pin.

This value will be assigned to the symbolic name

derived from the port pin container short name.

Property	Value
type	ECUC-INTEGER-PARAM-DEF
origin	AUTOSAR_ECUC
symbolicNameValue	true
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	false
valueConfigClasses	VARIANT-POST-BUILD: PRE-COMPILE
	VARIANT-PRE-COMPILE: PRE-COMPILE
defaultValue	2
max	145
min	1

4.19 Parameter PortPinPcr

Used to specify the PCR (Port Configuration Register) for the configured pin.

Property	Value
type	ECUC-INTEGER-PARAM-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	true
valueConfigClasses	VARIANT-POST-BUILD: POST-BUILD
	VARIANT-PRE-COMPILE: PRE-COMPILE
defaultValue	0
max	144
min	0

4.20 Parameter PortPinMode

Selects the PORT pin mode from the modes list. One or more modes may be valid for a pin. This way it is possible to select between multiple modes.(e.g. DIO (GPIO option) or ICU (eTimer option)).

Property	Value
type	ECUC-ENUMERATION-PARAM-DEF
origin	AUTOSAR_ECUC
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	true
valueConfigClasses	VARIANT-POST-BUILD: POST-BUILD
	VARIANT-PRE-COMPILE: PRE-COMPILE
defaultValue	GPIO

Property	Value
literals	['GPIO', 'ADC0_SE10_CMP0_IN5', 'ADC0_SE11_CMP0_IN4', 'ADC0_SE13', 'ADC0_SE14', 'ADC0_SE2', 'ADC0_SE3', 'ADC0_SE4_ADC1_SE14', 'ADC0_SE5_ADC1_SE15', 'ADC0_SE8', 'ADC0_SE9', 'ADC1_SE0', 'ADC1_SE10', 'ADC1_SE11', 'ADC1_SE12', 'ADC1_SE14', 'ADC1_SE15', 'ADC1_SE1', 'ADC1_SE2', 'ADC1_SE3', 'ADC1_SE6', 'ADC1_SE7', 'ADC1_SE8_ADC0_SE8', 'ADC1_SE9_ADC0_SE9', 'CAN0_RX', 'CAN0_TX', 'CLKOUT', 'CMP0_IN2', 'CMP0_OUT', 'CMP0_RRT', 'DISABLED', 'EWM_IN', 'EWM_OUT_B', 'EXTAL', 'FTM0_CH0', 'FTM0_CH1', 'FTM0_CH2', 'FTM0_CH3', 'FTM0_CH4', 'FTM0_CH5', 'FTM0_CH6', 'FTM0_CH7', 'FTM0_FLT1', 'FTM0_FLT2', 'FTM0_FLT3', 'FTM1_CH0', 'FTM1_CH2', 'FTM1_CH3', 'FTM1_CH4', 'FTM1_CH6', 'FTM1_CH7', 'FTM1_FLT0', 'FTM1_FLT1', 'FTM1_FLT2', 'FTM1_QD_PHB', 'FTM2_CH0', 'FTM2_CH1', 'FTM2_CH2', 'FTM2_CH3', 'FTM2_CH5', 'FTM2_CH6', 'FTM2_CH7', 'FTM2_QD_PHA', 'FTM2_QD_PHB', 'FTM3_CH0', 'FTM3_CH1', 'FTM3_CH2', 'FTM3_CH3', 'FTM3_CH4', 'FTM3_CH5', 'FTM3_CH6', 'FTM3_CH7', 'FTM3_FLT0', 'FTM3_FLT1', 'FTM3_FLT2', 'FTM3_FLT3', 'FXIO_D0', 'FXIO_D1', 'FXIO_D2', 'FXIO_D3', 'FXIO_D4', 'FXIO_D5', 'FXIO_D6', 'FXIO_D7', 'JTAG_TCLK_SWD_CLK', 'JTAG_TDI', 'JTAG_TDO_NOETM_TRACE_SWO', 'JTAG_TMS_SWD_DIO', 'LPI2C0_HREQ', 'LPI2C0_SCL', 'LPI2C0_SDAS', 'LPI2C0_SDA', 'LPSPI0_PCS0', 'LPSPI0_PCS1', 'LPSPI0_PCS2', 'LPSPI0_PCS3', 'LPSPI0_SCK', 'LPSPI0_SIN', 'LPSPI0_SOUT', 'LPSPI1_PCS0', 'LPSPI1_PCS1', 'LPSPI1_PCS3', 'LPSPI1_SCK', 'LPSPI1_SIN', 'LPSPI1_SOUT', 'LPSPI2_PCS0', 'LPSPI2_PCS3', 'LPSPI2_SCK', 'LPSPI2_SIN', 'LPSPI2_SOUT', 'LPTMR0_ALT1', 'LPTMR0_ALT3', 'LPUART0_CTS', 'LPUART0_RTS', 'LPUART0_RX', 'LPUART0_TX', 'LPUART1_CTS', 'LPUART1_RTS', 'LPUART1_RX', 'LPUART1_TX', 'NMI_B', 'RESET_B', 'RTC_CLKIN', 'RTC_CLKOUT', 'TCLK0', 'TCLK1', 'TCLK2', 'TRGMUX_IN0', 'TRGMUX_IN10', 'TRGMUX_IN11', 'TRGMUX_IN1', 'TRGMUX_IN4', 'TRGMUX_IN5', 'TRGMUX_IN8', 'TRGMUX_OUT1', 'TRGMUX_OUT2', 'TRGMUX_OUT5', 'TRGMUX_OUT6', 'TRGMUX_OUT7', 'XTAL']

4.21 Parameter PortPinDSE

Selects the drive strenght length value for this pin.

Property	Value
type	ECUC-ENUMERATION-PARAM-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	true
valueConfigClasses	VARIANT-POST-BUILD: POST-BUILD
	VARIANT-PRE-COMPILE: PRE-COMPILE

Property	Value
defaultValue	Low__Drive__Strength
literals	['Low__Drive__Strength', 'High__Drive__Strength']

4.22 Parameter PortPinPE

Selects if the pull-up or pull-down resistors are enabled.

Property	Value
type	ECUC-ENUMERATION-PARAM-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	true
valueConfigClasses	VARIANT-POST-BUILD: POST-BUILD
	VARIANT-PRE-COMPILE: PRE-COMPILE
defaultValue	PullDisabled
literals	['PullDisabled', 'PullEnabled']

4.23 Parameter PortPinPS

Selects between the pull-up and pull-down resistors. Only valid when PortPin PE is set to ’PullEnabled’.

Property	Value
type	ECUC-ENUMERATION-PARAM-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	true
valueConfigClasses	VARIANT-POST-BUILD: POST-BUILD
	VARIANT-PRE-COMPILE: PRE-COMPILE
defaultValue	PullDown
literals	['PullDown', 'PullUp']

4.24 Parameter PortPinSlewRate

Configures Slew Rate for the configured Pin.

This is an implementation specific parameter.

Property	Value
type	ECUC-ENUMERATION-PARAM-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	true
valueConfigClasses	VARIANT-POST-BUILD: POST-BUILD
	VARIANT-PRE-COMPILE: PRE-COMPILE
defaultValue	FAST_SLEW_RATE
literals	['FAST_SLEW_RATE', 'SLOW_SLEW_RATE']

4.25 Parameter PortPinDirection

Selects the direction of the pin (IN, OUT, HIGH_Z) that will be configured by Port_Init() function if the pin is configured as GPIO.

If the direction is not changeable, the value configured here is fixed. For the Alternative Function modes (PortPinMode is different than GPIO),

the setting in this enumeration control is kept in the port configuration structure and it is used when Port_SetPinMode() is called at runtime to change the mode of the pin to GPIO.

Set input pin to high-Z not available in S32K11X

Property	Value
type	ECUC-ENUMERATION-PARAM-DEF
origin	AUTOSAR_ECUC
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	true
valueConfigClasses	VARIANT-POST-BUILD: POST-BUILD
	VARIANT-PRE-COMPILE: PRE-COMPILE
defaultValue	PORT_PIN_IN
literals	['PORT_PIN_IN', 'PORT_PIN_OUT', 'PORT_PIN_HIGH_Z']

4.26 Parameter PortPinInitialMode

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.

Property	Value
type	ECUC-ENUMERATION-PARAM-DEF
origin	AUTOSAR_ECUC
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	true
valueConfigClasses	VARIANT-POST-BUILD: POST-BUILD
	VARIANT-PRE-COMPILE: PRE-COMPILE
defaultValue	PORT_GPIO_MODE
literals	['PORT_GPIO_MODE', 'PORT_ALT1_FUNC_MODE', 'PORT_ALT2_F↩ UNC_MODE', 'PORT_ALT3_FUNC_MODE', 'PORT_ALT4_FUNC_MO↩ DE', 'PORT_ALT5_FUNC_MODE', 'PORT_ALT6_FUNC_MODE', 'POR↩ T_ALT7_FUNC_MODE']

4.27 Parameter PortPinLevelValue

Port Pin Level value from Port pin list.

Property	Value
type	ECUC-ENUMERATION-PARAM-DEF
origin	AUTOSAR_ECUC
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	true
valueConfigClasses	VARIANT-POST-BUILD: POST-BUILD
	VARIANT-PRE-COMPILE: PRE-COMPILE
defaultValue	PORT_PIN_LEVEL_LOW
literals	['PORT_PIN_LEVEL_HIGH', 'PORT_PIN_LEVEL_LOW', 'PORT_PIN↩ _LEVEL_NOTCHANGED']

4.28 Reference PortPinEcucPartitionRef

EN: Maps the Port pin to zero a multiple ECUC partitions. The ECUC partitions referenced are a subset of the ECUC partitions where the Port driver is mapped to.

EN: Tags: atp.Status=draft

NoteThe S32K1 platform will not support the Multicore feature.

Property	Value
type	ECUC-REFERENCE-DEF
origin	AUTOSAR_ECUC
lowerMultiplicity	0
upperMultiplicity	Infinite
postBuildVariantMultiplicity	true
multiplicityConfigClasses	VARIANT-POST-BUILD: PRE-COMPILE
	VARIANT-PRE-COMPILE: PRE-COMPILE
postBuildVariantValue	true
valueConfigClasses	VARIANT-POST-BUILD: PRE-COMPILE
	VARIANT-PRE-COMPILE: PRE-COMPILE
requiresSymbolicNameValue	False
destination	/AUTOSAR/EcucDefs/EcuC/EcucPartitionCollection/EcucPartition

4.29 Container UnTouchedPortPin

List containing Pins that will not be touched by Port_Init() function.

Included subcontainers:

- None

Property	Value
type	ECUC-PARAM-CONF-CONTAINER-DEF
lowerMultiplicity	0
upperMultiplicity	Infinite
postBuildVariantMultiplicity	false
multiplicityConfigClasses	VARIANT-POST-BUILD: PRE-COMPILE
	VARIANT-PRE-COMPILE: PRE-COMPILE

4.30 Parameter PortPinPcr

Used to specify the PCR (Port Configuration Register) for the configured pin.

Property	Value
type	ECUC-INTEGER-PARAM-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	true
valueConfigClasses	VARIANT-POST-BUILD: POST-BUILD
	VARIANT-PRE-COMPILE: PRE-COMPILE
defaultValue	0
max	144
min	0

4.31 Container DigitalFilter

Configuration of a Digital Filter module available on the platfom.

Included subcontainers:

- DigitalFilterChannel

Property	Value
type	ECUC-PARAM-CONF-CONTAINER-DEF
lowerMultiplicity	0
upperMultiplicity	Infinite
postBuildVariantMultiplicity	true
multiplicityConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: POST-BUILD

4.32 Parameter DigitalFilterPort

Selects one of the PORTs available on the platform.

Property	Value
type	ECUC-STRING-PARAM-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	true
valueConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: POST-BUILD
defaultValue	PORT_A

4.33 Parameter DigitalFilterClock

Configures the clock source for the digital input filters.

Property	Value
type	ECUC-ENUMERATION-PARAM-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	true
valueConfigClasses	VARIANT-POST-BUILD: POST-BUILD
	VARIANT-PRE-COMPILE: PRE-COMPILE
defaultValue	BUS_CLOCK
literals	['BUS_CLOCK', 'LPO_CLOCK']

4.34 Parameter DigitalFilterWidth

Configures the maximum size of the glitches, in clock cycles, that the digital filter absorbs for the enabled digital filters. Glitches that are longer than this register setting will pass through the digital filter, and glitches that are equal to or less than this register setting are filtered.

Property	Value
type	ECUC-INTEGER-PARAM-DEF
origin	NXP

Property	Value
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	true
valueConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: POST-BUILD
defaultValue	0
max	31
min	0

4.35 Container DigitalFilterChannel

Configuration of a Digital Filter channel available on the platfom.

Included subcontainers:

- None

Property	Value
type	ECUC-PARAM-CONF-CONTAINER-DEF
lowerMultiplicity	0
upperMultiplicity	Infinite
postBuildVariantMultiplicity	false
multiplicityConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: POST-BUILD

4.36 Parameter DigitalFilterChannelIndex

Selects the channel in the port for which Digital Filter will be enabled.

Property	Value
type	ECUC-INTEGER-PARAM-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A

Property	Value
multiplicityConfigClasses	N/A
postBuildVariantValue	true
valueConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: POST-BUILD
defaultValue	0
max	31
min	0

4.37 Container PortGeneral

Module wide configuration parameters of the PORT driver.

Included subcontainers:

- None

Property	Value
type	ECUC-PARAM-CONF-CONTAINER-DEF
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A

4.38 Parameter PortDevErrorDetect

Switches the Development Error Detection and Notification ON or OFF.

Property	Value
type	ECUC-BOOLEAN-PARAM-DEF
origin	AUTOSAR_ECUC
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	false
valueConfigClasses	VARIANT-POST-BUILD: PRE-COMPILE
	VARIANT-PRE-COMPILE: PRE-COMPILE
defaultValue	true

4.39 Parameter PortCiPortIPDevErrorDetect

Enables and Disables DevAssert checks in IP code.

True: Enabled.

False: Disabled.

Property	Value
type	ECUC-BOOLEAN-PARAM-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	false
valueConfigClasses	VARIANT-POST-BUILD: PRE-COMPILE
	VARIANT-PRE-COMPILE: PRE-COMPILE
defaultValue	false

4.40 Parameter PortSetPinDirectionApi

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

Property	Value
type	ECUC-BOOLEAN-PARAM-DEF
origin	AUTOSAR_ECUC
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	false
valueConfigClasses	VARIANT-POST-BUILD: PRE-COMPILE
	VARIANT-PRE-COMPILE: PRE-COMPILE
defaultValue	true

4.41 Parameter PortSet2PinsDirectionApi

Pre-processor switch to enable/disable the use of the function Port_Set2PinsDirection().

Property	Value
type	ECUC-BOOLEAN-PARAM-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	false
valueConfigClasses	VARIANT-POST-BUILD: PRE-COMPILE
	VARIANT-PRE-COMPILE: PRE-COMPILE
defaultValue	false

4.42 Parameter PortSetPinModeApi

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

Property	Value
type	ECUC-BOOLEAN-PARAM-DEF
origin	AUTOSAR_ECUC
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	false
valueConfigClasses	VARIANT-POST-BUILD: PRE-COMPILE
	VARIANT-PRE-COMPILE: PRE-COMPILE
defaultValue	true

4.43 Parameter PortVersionInfoApi

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

Property	Value
type	ECUC-BOOLEAN-PARAM-DEF
origin	AUTOSAR_ECUC
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1

Property	Value
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	false
valueConfigClasses	VARIANT-POST-BUILD: PRE-COMPILE
	VARIANT-PRE-COMPILE: PRE-COMPILE
defaultValue	true

4.44 Parameter PortSetPinModeDoesNotTouchGpioLevel

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.

Property	Value
type	ECUC-BOOLEAN-PARAM-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	false
valueConfigClasses	VARIANT-POST-BUILD: PRE-COMPILE
	VARIANT-PRE-COMPILE: PRE-COMPILE
defaultValue	false

4.45 Parameter PortSetAsUnusedPinApi

Pre-processor switch to enable/disable the use of the function Port_SetAsUnusedPin() and Port_SetAsUsedPin(). The function Port_SetAsUnusedPin shall configure the referenced pin with all the properties specified in the NotUsedPortPin container. The function Port_SetAsUsedPin shall configure the referenced pin with all the properties that where set during the Port_Init operation.

TRUE: Enabled - Functions Port_SetAsUnusedPin() and Port_SetAsUsedPin() are available.

FALSE: Disabled - Functions Port_SetAsUnusedPin() and Port_SetAsUsedPin() are not available.

Property	Value
type	ECUC-BOOLEAN-PARAM-DEF
origin	NXP

Property	Value
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	false
valueConfigClasses	VARIANT-POST-BUILD: PRE-COMPILE
	VARIANT-PRE-COMPILE: PRE-COMPILE
defaultValue	false

4.46 Parameter PortResetPinModeApi

Pre-processor switch to enable/disable the use of the function Port_ResetPinMode(). The function Port_ResetPinMode shall revert the port pin mode of the referenced pin to the value that was set by Port_Init operation.

TRUE: Enabled - Function Port_ResetPinMode() is available.

FALSE: Disabled - Function Port_ResetPinMode() is not available.

Property	Value
type	ECUC-BOOLEAN-PARAM-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	false
valueConfigClasses	VARIANT-POST-BUILD: PRE-COMPILE
	VARIANT-PRE-COMPILE: PRE-COMPILE
defaultValue	false

4.47 Parameter PortEnableUserModeSupport

This parameter is added in Port configuration in order to keep a consistent design over the entire set of RTD drivers.

It cannot be configured by the user and is always set to ’false’.

There are no registers used by the driver which require special measures in order to be accessed from user mode, so Port driver can be run from either user or supervisor mode.

Property	Value
type	ECUC-BOOLEAN-PARAM-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	false
valueConfigClasses	VARIANT-POST-BUILD: PRE-COMPILE
	VARIANT-PRE-COMPILE: PRE-COMPILE
defaultValue	false

4.48 Parameter PortMulticoreSupport

This parameter globally enables the possibility to support multicore. If this parmeter is enabled, at least one EcucPartition needs to be defined (in all variants).

NoteThe S32K1 platform will not support the Multicore feature.

Property	Value
type	ECUC-BOOLEAN-PARAM-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	false
valueConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: PRE-COMPILE
defaultValue	false

4.49 Reference PortEcucPartitionRef

Maps the Port driver to zero a multiple ECUC partitions to make the modules API available in this partition.

Tags: atp.Status=draft

NoteThe S32K1 platform will not support the Multicore feature.

Property	Value
type	ECUC-REFERENCE-DEF
origin	AUTOSAR_ECUC
lowerMultiplicity	0
upperMultiplicity	Infinite
postBuildVariantMultiplicity	true
multiplicityConfigClasses	VARIANT-POST-BUILD: PRE-COMPILE
	VARIANT-PRE-COMPILE: PRE-COMPILE
postBuildVariantValue	true
valueConfigClasses	VARIANT-POST-BUILD: PRE-COMPILE
	VARIANT-PRE-COMPILE: PRE-COMPILE
requiresSymbolicNameValue	False
destination	/AUTOSAR/EcucDefs/EcuC/EcucPartitionCollection/EcucPartition

4.50 Container CommonPublishedInformation

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

Included subcontainers:

- None

Property	Value
type	ECUC-PARAM-CONF-CONTAINER-DEF
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A

4.51 Parameter ArReleaseMajorVersion

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

Property	Value
type	ECUC-INTEGER-PARAM-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A

Property	Value
multiplicityConfigClasses	N/A
postBuildVariantValue	false
valueConfigClasses	VARIANT-POST-BUILD: PUBLISHED-INFORMATION
	VARIANT-PRE-COMPILE: PUBLISHED-INFORMATION
defaultValue	4
max	4
min	4

4.52 Parameter ArReleaseMinorVersion

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

Property	Value
type	ECUC-INTEGER-PARAM-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	false
valueConfigClasses	VARIANT-POST-BUILD: PUBLISHED-INFORMATION
	VARIANT-PRE-COMPILE: PUBLISHED-INFORMATION
defaultValue	7
max	7
min	7

4.53 Parameter ArReleaseRevisionVersion

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

Property	Value
type	ECUC-INTEGER-PARAM-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A

Property	Value
postBuildVariantValue	false
valueConfigClasses	VARIANT-POST-BUILD: PUBLISHED-INFORMATION
	VARIANT-PRE-COMPILE: PUBLISHED-INFORMATION
defaultValue	0
max	0
min	0

4.54 Parameter ModuleId

Module ID of this module from Module List.

Property	Value
type	ECUC-INTEGER-PARAM-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	false
valueConfigClasses	VARIANT-POST-BUILD: PUBLISHED-INFORMATION
	VARIANT-PRE-COMPILE: PUBLISHED-INFORMATION
defaultValue	124
max	124
min	124

4.55 Parameter SwMajorVersion

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

Property	Value
type	ECUC-INTEGER-PARAM-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	false
valueConfigClasses	VARIANT-POST-BUILD: PUBLISHED-INFORMATION

Property	Value
	VARIANT-PRE-COMPILE: PUBLISHED-INFORMATION
defaultValue	2
max	2
min	2

4.56 Parameter SwMinorVersion

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

Property	Value
type	ECUC-INTEGER-PARAM-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	false
valueConfigClasses	VARIANT-POST-BUILD: PUBLISHED-INFORMATION
	VARIANT-PRE-COMPILE: PUBLISHED-INFORMATION
defaultValue	0
max	0
min	0

4.57 Parameter SwPatchVersion

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

Property	Value
type	ECUC-INTEGER-PARAM-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	false
valueConfigClasses	VARIANT-POST-BUILD: PUBLISHED-INFORMATION
	VARIANT-PRE-COMPILE: PUBLISHED-INFORMATION
defaultValue	0

Property	Value
max	0
min	0

4.58 Parameter VendorApiInfix

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>.

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.

Property	Value
type	ECUC-STRING-PARAM-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	0
upperMultiplicity	1
postBuildVariantMultiplicity	false
multiplicityConfigClasses	VARIANT-POST-BUILD: PUBLISHED-INFORMATION
	VARIANT-PRE-COMPILE: PUBLISHED-INFORMATION
postBuildVariantValue	false
valueConfigClasses	VARIANT-POST-BUILD: PUBLISHED-INFORMATION
	VARIANT-PRE-COMPILE: PUBLISHED-INFORMATION
defaultValue	

4.59 Parameter VendorId

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

Property	Value
type	ECUC-INTEGER-PARAM-DEF
origin	NXP
symbolicNameValue	false

Property	Value
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	false
valueConfigClasses	VARIANT-POST-BUILD: PUBLISHED-INFORMATION
	VARIANT-PRE-COMPILE: PUBLISHED-INFORMATION
defaultValue	43
max	43
min	43

Chapter 5

Module Index

5.1 Software Specification

Here is a list of all modules:

Chapter 6

Module Documentation

6.1 Port HLD

6.1.1 Detailed Description

Macros

- #define PORT_VENDOR_ID

    Parameters that shall be published within the Port driver header file and also in the module's description file.
- #define PORT_E_PARAM_CONFIG

    The PORT module is not properly configured.
- #define PORT_INSTANCE_ID

    Instance ID of port driver.
- #define PORT_INIT_ID

    API service ID for PORT Init function.
- #define PORT_SETPINDIRECTION_ID

    API service ID for PORT set pin direction function.
- #define PORT_REFRESHPINDIRECTION_ID

    API service ID for PORT refresh pin direction function.
- #define PORT_GETVERSIONINFO_ID

    API service ID for PORT get version info function.
- #define PORT_SETPINMODE_ID

    API service ID for PORT set pin mode.
- #define PORT_SETASUNUSEDPIN_ID

    API service ID for PORT set as unused pin.
- #define PORT_SETASUSEDPIN_ID

    API service ID for PORT set as used pin.
- #define PORT_RESETPINMODE_ID

    API service ID for PORT reset pin mode.
- #define PORT_SET2PINSDIRECTION_ID

    API service ID for PORT set 2 pins direction function.
- #define PORT_E_PARAM_PIN

Error ID of port driver.
- #define PORT_E_DIRECTION_UNCHANGEABLE

  Port Pin Direction not configured as changeable.
- #define PORT_E_INIT_FAILED

  API Port_Init() service called with wrong parameter.
- #define PORT_E_PARAM_INVALID_MODE

  API Port_SetPinMode() service called when mode is invalid.
- #define PORT_E_MODE_UNCHANGEABLE

  API Port_SetPinMode() service called when mode is unchangeable.
- #define PORT_E_UNINIT

  API service called without module initialization.
- #define PORT_E_PARAM_POINTER

  API service called with NULL Pointer Parameter.

Function Reference

- void Port_Init (const Port_ConfigType ∗ConfigPtr)

  Port driver initialization function.
- void Port_SetPinDirection (Port_PinType Pin, Port_PinDirectionType Direction)

  Port_SetPinDirection.
- void Port_Set2PinsDirection (Port_PinType Pin1, Port_PinType Pin2, Port_PinDirectionType Direction)

  Sets the direction of 2 pins.
- void Port_SetPinMode (Port_PinType Pin, Port_PinModeType Mode)

  Port_SetPinMode.
- void Port_GetVersionInfo (Std_VersionInfoType ∗versioninfo)

  Port_GetVersionInfo.
- void Port_RefreshPortDirection (void)

  Port_RefreshPortDirection.
- void Port_SetAsUnusedPin (Port_PinType Pin)

  Port_SetAsUnusedPin.
- void Port_SetAsUsedPin (Port_PinType Pin)

  Port_SetAsUsedPin.
- void Port_ResetPinMode (Port_PinType Pin)

  Port_ResetPinMode.

6.1.2 Macro Definition Documentation

6.1.2.1 PORT_VENDOR_ID

```
#define PORT_VENDOR_ID
```

Parameters that shall be published within the Port driver header file and also in the module's description file.

Definition at line 60 of file Port.h.

6.1.2.2 PORT_E_PARAM_CONFIG

```
#define PORT_E_PARAM_CONFIG
```

The PORT module is not properly configured.

Definition at line 132 of file Port.h.

6.1.2.3 PORT_INSTANCE_ID

```
#define PORT_INSTANCE_ID
```

Instance ID of port driver.

Definition at line 139 of file Port.h.

6.1.2.4 PORT_INIT_ID

```
#define PORT_INIT_ID
```

API service ID for PORT Init function.

Parameters used when raising an error/exception.

Definition at line 153 of file Port.h.

6.1.2.5 PORT_SETPINDIRECTION_ID

```
#define PORT_SETPINDIRECTION_ID
```

API service ID for PORT set pin direction function.

Parameters used when raising an error/exception.

Definition at line 162 of file Port.h.

6.1.2.6 PORT_REFRESHPINDIRECTION_ID

```
#define PORT_REFRESHPINDIRECTION_ID
```

API service ID for PORT refresh pin direction function.

Parameters used when raising an error/exception.

Definition at line 170 of file Port.h.

6.1.2.7 PORT_GETVERSIONINFO_ID

```
#define PORT_GETVERSIONINFO_ID
```

API service ID for PORT get version info function.

Parameters used when raising an error/exception.

Definition at line 179 of file Port.h.

6.1.2.8 PORT_SETPINMODE_ID

```
#define PORT_SETPINMODE_ID
```

API service ID for PORT set pin mode.

Parameters used when raising an error/exception.

Definition at line 188 of file Port.h.

6.1.2.9 PORT_SETASUNUSEDPIN_ID

```
#define PORT_SETASUNUSEDPIN_ID
```

API service ID for PORT set as unused pin.

Parameters used when raising an error/exception.

Definition at line 194 of file Port.h.

6.1.2.10 PORT_SETASUSEDPIN_ID

```
#define PORT_SETASUSEDPIN_ID
```

API service ID for PORT set as used pin.

Parameters used when raising an error/exception.

Definition at line 200 of file Port.h.

6.1.2.11 PORT_RESETPINMODE_ID

```
#define PORT_RESETPINMODE_ID
```

API service ID for PORT reset pin mode.

Parameters used when raising an error/exception.

Definition at line 209 of file Port.h.

6.1.2.12 PORT_SET2PINSDIRECTION_ID

```
#define PORT_SET2PINSDIRECTION_ID
```

API service ID for PORT set 2 pins direction function.

Parameters used when raising an error/exception.

Definition at line 220 of file Port.h.

6.1.2.13 PORT_E_PARAM_PIN

```
#define PORT_E_PARAM_PIN
```

Error ID of port driver.

The following errors and exception are detectable by the PORT driver if development error detection is enabled.

Invalid Port Pin ID requested.

Det Error value, returned by Port_SetPinDirection and Port_PinMode if an wrong PortPin ID is passed.

Definition at line 239 of file Port.h.

6.1.2.14 PORT_E_DIRECTION_UNCHANGEABLE

```
#define PORT_E_DIRECTION_UNCHANGEABLE
```

Port Pin Direction not configured as changeable.

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

Definition at line 248 of file Port.h.

6.1.2.15 PORT_E_INIT_FAILED

```
#define PORT_E_INIT_FAILED
```

API Port_Init() service called with wrong parameter.

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

Definition at line 258 of file Port.h.

6.1.2.16 PORT_E_PARAM_INVALID_MODE

```
#define PORT_E_PARAM_INVALID_MODE
```

API Port_SetPinMode() service called when mode is invalid.

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

Definition at line 267 of file Port.h.

6.1.2.17 PORT_E_MODE_UNCHANGEABLE

```
#define PORT_E_MODE_UNCHANGEABLE
```

API Port_SetPinMode() service called when mode is unchangeable.

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

Definition at line 276 of file Port.h.

6.1.2.18 PORT_E_UNINIT

```
#define PORT_E_UNINIT
```

API service called without module initialization.

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

Definition at line 285 of file Port.h.

6.1.2.19 PORT_E_PARAM_POINTER

```
#define PORT_E_PARAM_POINTER
```

API service called with NULL Pointer Parameter.

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

Definition at line 294 of file Port.h.

6.1.3 Function Reference

6.1.3.1 Port_Init()

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

Port driver initialization function.

Function used for initializing the port driver and for initializing the configured pins.

Parameters

in	*Port_ConfigType*	∗ ConfigPtr Pointer to configuration (NULL_PTR if only one variant is used)

Returns

void

6.1.3.2 Port_SetPinDirection()

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

Port_SetPinDirection.

Function used for changing the pin direction at runtime

Parameters

in		

6.1.3.3 Port_Set2PinsDirection()

```
void Port_Set2PinsDirection (
            Port_PinType Pin1,
            Port_PinType Pin2,
            Port_PinDirectionType Direction )
```

Sets the direction of 2 pins.

The function `Port_Set2PinsDirection()` will set the port pins direction during runtime.

Precondition

`Port_Init()` must have been called first. In order to change the pin direction the PortPinDirection↩ Changeable flag must have been set to `TRUE` for both pins.

Parameters

in	*Pin1*	Pin 1 ID number.
in	*Pin2*	Pin 2 ID number.
in	*Direction*	Port Pin direction.

Port_Set2PinsDirection_Activity

6.1.3.4 Port_SetPinMode()

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

Port_SetPinMode.

Function used to change the pin mode at runtime.

Parameters

in		

6.1.3.5 Port_GetVersionInfo()

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

Port_GetVersionInfo.

Function used to read the driver version information

Parameters

in	*versioninfo*	pointer to structure that will contain the version information

Returns

void

6.1.3.6 Port_RefreshPortDirection()

```
void Port_RefreshPortDirection (
            void  )
```

Port_RefreshPortDirection.

function used to reset the direction of the pin

Returns

void

6.1.3.7 Port_SetAsUnusedPin()

```
void Port_SetAsUnusedPin (
            Port_PinType Pin )
```

Port_SetAsUnusedPin.

configures the referenced pin with all the properties specified in the NotUsedPortPin container.

Returns

void

6.1.3.8 Port_SetAsUsedPin()

```
void Port_SetAsUsedPin (
            Port_PinType Pin )
```

Port_SetAsUsedPin.

configures the referenced pin with all the properties that where set during the Port_Init operation.

Returns

void

6.1.3.9 Port_ResetPinMode()

```
void Port_ResetPinMode (
            Port_PinType Pin )
```

Port_ResetPinMode.

reverts the port pin mode of the referenced pin to the value that was set by Port_Init operation.

Returns

void

6.2 Port IPL

6.2.1 Detailed Description

Data Structures

- struct Port_Ci_Port_Ip_PinSettingsConfig

  Defines the converter configuration. More...

Macros

- #define PORT_CI_PORT_IP_VENDOR_ID_H

  Parameters that shall be published within the Port driver header file and also in the module description file.
- #define PORT_CI_PORT_IP_VENDOR_ID_TYPES_H

  Parameters that shall be published within the Port driver header file and also in the module description file.

Types Reference

- typedef uint8 Port_Ci_Port_Ip_PortPinsLevelType

  Type of a port levels representation. Implements : Port_Ci_Port_Ip_PortPinsLevelType.

Enum Reference

- enum Port_Ci_Port_Ip_PortPullConfig

  Internal resistor pull feature selection Implements : Port_Ci_Port_Ip_PortPullConfig.
- enum Port_Ci_Port_Ip_PortSlewRateControl

  Configures the slew rate control. Implements : Port_Ci_Port_Ip_PortSlewRateControl.
- enum Port_Ci_Port_Ip_PortMux

  Configures the Pin output muxing selection Implements : Port_Ci_Port_Ip_PortMux.
- enum Port_Ci_Port_Ip_PortGlobalControlPins
- enum Port_Ci_Port_Ip_PortLockRegister

  Configures the Lock Register enable Implements : Port_Ci_Port_Ip_PortLockRegister.
- enum Port_Ci_Port_Ip_PortOpenDrain

  Configures the Open Drain Enable field. Implements : Port_Ci_Port_Ip_PortOpenDrain.
- enum Port_Ci_Port_Ip_PortDriveStrength

  Configures the drive strength. Implements : Port_Ci_Port_Ip_PortDriveStrength.
- enum Port_Ci_Port_Ip_PortDirectionType

  Configures port direction.
- enum Port_Ci_Port_Ip_InterleaveMux

Function Reference

- Port_Ci_Port_Ip_PortStatusType Port_Ci_Port_Ip_Init (uint32 pinCount, const Port_Ci_Port_Ip_PinSettingsConfig config[ ])

  Initializes the pins with the given configuration structure.
- void Port_Ci_Port_Ip_SetMuxModeSel (PORT_Type ∗const base, uint32 pin, Port_Ci_Port_Ip_PortMux mux)

  Configures the pin muxing.
- void Port_Ci_Port_Ip_EnableDigitalFilter (PORT_Type ∗const base, uint32 pin)

  Enables digital filter for digital pin muxing.
- void Port_Ci_Port_Ip_DisableDigitalFilter (PORT_Type ∗const base, uint32 pin)

  Disables digital filter for digital pin muxing.
- void Port_Ci_Port_Ip_ConfigDigitalFilter (PORT_Type ∗const base, const Port_Ci_Port_Ip_Digital↩FilterConfigType ∗config)

  Configures digital filter for port with given configuration.
- void Port_Ci_Port_Ip_SetGlobalPinControl (PORT_Type ∗const base, uint16 pins, uint16 value, Port_Ci_Port_Ip_PortGlobalControlPins halfPort)

  Quickly configures multiple pins with the same pin configuration.
- Port_Ci_Port_Ip_PortMux Port_Ci_Port_Ip_ConfigureInterleave (const PORT_Type ∗const base, uint32 pin, Port_Ci_Port_Ip_PortMux muxing)

  Configure the Interleave feature.

6.2.2 Data Structure Documentation

6.2.2.1 struct Port_Ci_Port_Ip_PinSettingsConfig

Defines the converter configuration.

This structure is used to configure the pins Implements : Port_Ci_Port_Ip_PinSettingsConfig

Definition at line 253 of file Port_Ci_Port_Ip_Types.h.

Data Fields

Type	Name	Description
PORT_Type ∗	portBase	The main PORT base pointer.
GPIO_Type ∗	gpioBase	The main GPIO base pointer.
uint32	pinPortIdx	Port pin number in a port.
Port_Ci_Port_Ip_PortPullConfig	pullConfig	Internal resistor pull feature selection.
Port_Ci_Port_Ip_PortMux	mux	Pin output muxing selection.
Port_Ci_Port_Ip_PortDirectionType	direction	Pin output muxing selection.

Data Fields

Type	Name	Description
Port_Ci_Port_Ip_PortDriveStrength	driveStrength	Configures DSE.
boolean	passiveFilter	Configures DFE.
Port_Ci_Port_Ip_PortSlewRateControl	slewRateCtrlSel	Configures SRE.
Port_Ci_Port_Ip_PortLockRegister	lockRegister	Configures LK.
Port_Ci_Port_Ip_PortOpenDrain	openDrain	Configures ODE.
boolean	digitalFilter	Configures digitalFilter.
Port_Ci_Port_Ip_PortPinsLevelType	initValue	Initial value.

6.2.3 Macro Definition Documentation

6.2.3.1 PORT_CI_PORT_IP_VENDOR_ID_H

```
#define PORT_CI_PORT_IP_VENDOR_ID_H
```

Parameters that shall be published within the Port driver header file and also in the module description file.

The integration of incompatible files shall be avoided.

Definition at line 60 of file Port_Ci_Port_Ip.h.

6.2.3.2 PORT_CI_PORT_IP_VENDOR_ID_TYPES_H

```
#define PORT_CI_PORT_IP_VENDOR_ID_TYPES_H
```

Parameters that shall be published within the Port driver header file and also in the module description file.

The integration of incompatible files shall be avoided.

Definition at line 56 of file Port_Ci_Port_Ip_Types.h.

6.2.4 Types Reference

6.2.4.1 Port_Ci_Port_Ip_PortPinsLevelType

```
typedef uint8 Port_Ci_Port_Ip_PortPinsLevelType
```

Type of a port levels representation. Implements : Port_Ci_Port_Ip_PortPinsLevelType.

Definition at line 114 of file Port_Ci_Port_Ip_Types.h.

6.2.5 Enum Reference

6.2.5.1 Port_Ci_Port_Ip_PortPullConfig

```
enum Port_Ci_Port_Ip_PortPullConfig
```

Internal resistor pull feature selection Implements : Port_Ci_Port_Ip_PortPullConfig.

Enumerator

PORT_INTERNAL_PULL_DOWN_ENABLED	internal pull-down resistor is enabled.
PORT_INTERNAL_PULL_UP_ENABLED	internal pull-up resistor is enabled.
PORT_INTERNAL_PULL_NOT_ENABLED	internal pull-down/up resistor is disabled.

Definition at line 120 of file Port_Ci_Port_Ip_Types.h.

6.2.5.2 Port_Ci_Port_Ip_PortSlewRateControl

```
enum Port_Ci_Port_Ip_PortSlewRateControl
```

Configures the slew rate control. Implements : Port_Ci_Port_Ip_PortSlewRateControl.

Enumerator

PORT_FAST_SLEW_RATE	Fast slew rate is configured on the corresponding pin, if the pin is configured as a digital output.
PORT_SLOW_SLEW_RATE	Slow slew rate is configured on the corresponding pin, if the pin is configured as a digital output.

Definition at line 132 of file Port_Ci_Port_Ip_Types.h.

6.2.5.3 Port_Ci_Port_Ip_PortMux

```
enum Port_Ci_Port_Ip_PortMux
```

Configures the Pin output muxing selection Implements : Port_Ci_Port_Ip_PortMux.

Enumerator

PORT_MUX_ALT0	DISABLED or ALT0 mode
PORT_MUX_AS_GPIO	corresponding pin is configured as GPIO
PORT_MUX_ALT2	chip-specific
PORT_MUX_ALT3	chip-specific
PORT_MUX_ALT4	chip-specific
PORT_MUX_ALT5	chip-specific
PORT_MUX_ALT6	chip-specific
PORT_MUX_ALT7	chip-specific
PORT_MUX_ADC_INTERLEAVE	chip-specific

Definition at line 143 of file Port_Ci_Port_Ip_Types.h.

6.2.5.4 Port_Ci_Port_Ip_PortGlobalControlPins

```
enum Port_Ci_Port_Ip_PortGlobalControlPins
```

Enumerator

PORT_GLOBAL_CONTROL_LOWER_HALF_PINS	the lower of pins is configured
PORT_GLOBAL_CONTROL_UPPER_HALF_PINS	the upper of pins is configured

Definition at line 156 of file Port_Ci_Port_Ip_Types.h.

6.2.5.5 Port_Ci_Port_Ip_PortLockRegister

```
enum Port_Ci_Port_Ip_PortLockRegister
```

Configures the Lock Register enable Implements : Port_Ci_Port_Ip_PortLockRegister.

Enumerator

PORT_LOCK_REGISTER_DISABLED	IFE OFF
PORT_LOCK_REGISTER_ENABLED	IFE ON

Definition at line 167 of file Port_Ci_Port_Ip_Types.h.

6.2.5.6 Port_Ci_Port_Ip_PortOpenDrain

```
enum Port_Ci_Port_Ip_PortOpenDrain
```

Configures the Open Drain Enable field. Implements : Port_Ci_Port_Ip_PortOpenDrain.

Enumerator

PORT_OPEN_DRAIN_DISABLED	ODE OFF
PORT_OPEN_DRAIN_ENABLED	ODE ON

Definition at line 179 of file Port_Ci_Port_Ip_Types.h.

6.2.5.7 Port_Ci_Port_Ip_PortDriveStrength

```
enum Port_Ci_Port_Ip_PortDriveStrength
```

Configures the drive strength. Implements : Port_Ci_Port_Ip_PortDriveStrength.

Enumerator

PORT_DRIVE_STRENGTH_LOW	Enables DSE.
PORT_DRIVE_STRENGTH_HIGH	Enables DSE.

Definition at line 215 of file Port_Ci_Port_Ip_Types.h.

6.2.5.8 Port_Ci_Port_Ip_PortDirectionType

```
enum Port_Ci_Port_Ip_PortDirectionType
```

Configures port direction.

Enumerator

PORT_CI_PORT_PIN_DISABLED	Sets port pin as disable or ALT0 mode.
PORT_CI_PORT_PIN_IN	Sets port pin as output.
PORT_CI_PORT_PIN_OUT	Sets port pin as bidirectional.
PORT_CI_PORT_PIN_HIGH_Z	Sets port pin as high_z.

Definition at line 239 of file Port_Ci_Port_Ip_Types.h.

6.2.5.9 Port_Ci_Port_Ip_InterleaveMux

```
enum Port_Ci_Port_Ip_InterleaveMux
```

Enumerator

PIN_ADC_INTERLEAVE_DISABLE0	xxx0b ADC1_SE14 channel is connected to PTB15
PIN_ADC_INTERLEAVE_DISABLE1	xx0xb ADC1_SE15 channel is connected to PTB16
PIN_ADC_INTERLEAVE_DISABLE2	x0xxb ADC0_SE8 channel is connected to PTC0
PIN_ADC_INTERLEAVE_DISABLE3	0xxxb ADC0_SE9 channel is connected to PTC1
PIN_ADC_INTERLEAVE_ENABLE0	xxx1b ADC1_SE14 channel is connected to PTB0
PIN_ADC_INTERLEAVE_ENABLE1	xx1xb ADC1_SE15 channel is connected to PTB1
PIN_ADC_INTERLEAVE_ENABLE2	x1xxb ADC0_SE8 channel is connected to PTB13
PIN_ADC_INTERLEAVE_ENABLE3	1xxxb ADC0_SE9 channel is connected to PTB14
PIN_ADC_INTERLEAVE_INVALID	ADC interleave is invalid

Definition at line 282 of file Port_Ci_Port_Ip_Types.h.

6.2.6 Function Reference

6.2.6.1 Port_Ci_Port_Ip_Init()

```
Port_Ci_Port_Ip_PortStatusType Port_Ci_Port_Ip_Init (
            uint32 pinCount,
            const Port_Ci_Port_Ip_PinSettingsConfig config[] )
```

Initializes the pins with the given configuration structure.

This function configures the pins with the options provided in the provided structure.

Parameters

in	*pinCount*	The number of configured pins in structure
in	*config*	The configuration structure

Returns

The status of the operation

6.2.6.2 Port_Ci_Port_Ip_SetMuxModeSel()

```
void Port_Ci_Port_Ip_SetMuxModeSel (
            PORT_Type *const base,
            uint32 pin,
            Port_Ci_Port_Ip_PortMux mux )
```

Configures the pin muxing.

This function configures the pin muxing.

Parameters

in	*base*	Port base pointer (PORTA, PORTB, PORTC, etc.)
in	*pin*	Port pin number
in	*mux*	Pin muxing slot selection

6.2.6.3 Port_Ci_Port_Ip_EnableDigitalFilter()

```
void Port_Ci_Port_Ip_EnableDigitalFilter (
            PORT_Type *const base,
            uint32 pin )
```

Enables digital filter for digital pin muxing.

This function enables digital filter feature for digital pin muxing

Parameters

in	*base*	Port base pointer (PORTA, PORTB, PORTC, etc.)
in	*pin*	Port pin number

6.2.6.4 Port_Ci_Port_Ip_DisableDigitalFilter()

```
void Port_Ci_Port_Ip_DisableDigitalFilter (
            PORT_Type *const base,
            uint32 pin )
```

Disables digital filter for digital pin muxing.

This function disables digital filter feature for digital pin muxing

Parameters

in	*base*	Port base pointer (PORTA, PORTB, PORTC, etc.)
in	*pin*	Port pin number

6.2.6.5 Port_Ci_Port_Ip_ConfigDigitalFilter()

```
void Port_Ci_Port_Ip_ConfigDigitalFilter (
            PORT_Type *const base,
            const Port_Ci_Port_Ip_DigitalFilterConfigType * config )
```

Configures digital filter for port with given configuration.

This function configures digital filter for port with given configuration

Note: Updating the filter configuration must be done only after all filters are disabled.

Parameters

in	*base*	Port base pointer (PORTA, PORTB, PORTC, etc.)
in	*config*	the digital filter configuration struct

6.2.6.6 Port_Ci_Port_Ip_SetGlobalPinControl()

```
void Port_Ci_Port_Ip_SetGlobalPinControl (
            PORT_Type *const base,
            uint16 pins,
            uint16 value,
            Port_Ci_Port_Ip_PortGlobalControlPins halfPort )
```

Quickly configures multiple pins with the same pin configuration.

This function quickly configures multiple pins within the one port for the same peripheral function with the same pin configuration. Supports up to 16 pins with the lower or upper half of pin registers at the same port.

Parameters

in	*base*	Port base pointer (PORTA, PORTB, PORTC, etc.)
in	*pins*	Pin mask where each bit represents one pin. For each bit: • 0: pins corresponding to bits with value of '1' is updated with the value input • 1: pins corresponding to bits with value of '0' is not updated with the value input
in	*value*	the config value will be updated for the pins are set to '1'
in	*halfPort*	the lower or upper half of pin registers at the same port

6.2.6.7 Port_Ci_Port_Ip_ConfigureInterleave()

```
Port_Ci_Port_Ip_PortMux Port_Ci_Port_Ip_ConfigureInterleave (
            const PORT_Type *const base,
            uint32 pin,
            Port_Ci_Port_Ip_PortMux muxing )
```

Configure the Interleave feature.

Configure the Interleave feature

Parameters

in	*base*	Port base pointer (PORTA, PORTB, PORTC, etc.)
in	*pin*	Port pin number
in	*muxing*	Pin muxing slot selection

Return values

none	

Precondition

The user mode is enabled