

User Manual

for S32K3 PORT Driver

Document Number: UM34PORTASRR21-11 Rev0000R3.0.0 Rev. 1.0

1 Revision History	2
2 Introduction	3
2.1 Supported Derivatives	3
2.2 Overview	4
2.3 About This Manual	5
2.4 Acronyms and Definitions	6
2.5 Reference List	6
3 Driver	7
3.1 Requirements	7
3.2 Driver Design Summary	7
3.3 Hardware Resources	8
3.4 Deviations from Requirements	8
3.5 Driver Limitations	12
3.6 Driver usage and configuration tips	12
3.7 Runtime errors	20
3.8 Symbolic Names Disclaimer	21
4 Tresos Configuration Plug-in	22
4.1 Module Port	24
4.2 Container PortConfigSet	24
4.3 Container NotUsedPortPin	25
4.4 Parameter PortPinPue	25
4.5 Parameter PortPinPus	26
4.6 Parameter PortPinDirection	26
4.7 Parameter PortPinLevelValue	27
4.8 Container PortContainer	27
4.9 Parameter PortNumberOfPortPins	28
4.10 Container PortPin	28
4.11 Parameter PortPinPue	29
4.12 Parameter PortPinPus	29
4.13 Parameter PortPinSafeMode	30
4.14 Parameter PortPinDse	30
4.15 Parameter PortPinWithReadBack	30
4.16 Parameter PortPinPke	31
4.17 Parameter PortPinIfc	31
4.18 Parameter PortPinDirectionChangeable	32
4.19 Parameter PortPinModeChangeable	32
4.20 Parameter PortPinInvertControl	33
4.21 Parameter PortPinSiul2Instance	34

4.22 Parameter PortPinId	34
4.23 Parameter PortPinPcr	35
4.24 Parameter PortPinDirection	35
4.25 Parameter PortPinInitialMode	36
4.26 Parameter PortPinMode	36
4.27 Parameter PortPinLevelValue	39
4.28 Parameter PortPinSlewRate	39
4.29 Parameter OBEGroupSelect	40
4.30 Parameter MscrPdacSlot	40
4.31 Parameter ImcrPdacSlot	41
4.32 Reference PortPinEcucPartitionRef	41
4.33 Container IGFSettings	41
4.34 Parameter IGF_FGEN	42
4.35 Parameter IGFChannel	42
4.36 Parameter IGF_IMM	43
4.37 Parameter IGF_PSSEL	43
4.38 Parameter IGF_FPRE	44
4.39 Parameter IGF_FFM	44
4.40 Parameter IGF_RFM	46
4.41 Parameter IGF_RTH	47
4.42 Parameter IGF_FTH	47
4.43 Container UnTouchedPortPin	47
4.44 Parameter PortPinSiul2Instance	48
4.45 Parameter PortPinPcr	48
4.46 Container UntouchedIMCR	49
4.47 Parameter IMCRSiul2Instance	49
4.48 Parameter UntouchedPortPinImcr	50
4.49 Container PortGeneral	50
4.50 Parameter PortDevErrorDetect	51
4.51 Parameter SIUL2PortIPDevErrorDetect	51
4.52 Parameter PortSetPinDirectionApi	51
4.53 Parameter PortSetPinModeApi	52
4.54 Parameter PortVersionInfoApi	52
4.55 Parameter PortSetPinModeDoesNotTouchGpioLevel	53
4.56 Parameter PortSetAsUnusedPinApi	53
4.57 Parameter PortResetPinModeApi	54
4.58 Parameter PortEnableUserModeSupport	54
4.59 Parameter PortMulticoreSupport	55
4.60 Parameter PortTspcSupport	55
4.61 Parameter SignalInversionConfigEnable	56

4.62 Parameter VirtWrapperSupport	56
4.63 Parameter PortCodeSizeOptimization	57
4.64 Reference PortEcucPartitionRef	57
4.65 Container CommonPublishedInformation	58
4.66 Parameter ArReleaseMajorVersion	58
4.67 Parameter ArReleaseMinorVersion	59
4.68 Parameter ArReleaseRevisionVersion	59
4.69 Parameter ModuleId	60
4.70 Parameter SwMajorVersion	60
4.71 Parameter SwMinorVersion	61
4.72 Parameter SwPatchVersion	61
4.73 Parameter VendorApiInfix	61
4.74 Parameter VendorId	62
5 Module Index	63
5.1 Software Specification	63
6 Module Documentation	64
6.1 IGF IPL	64
6.1.1 Detailed Description	64
6.1.2 Data Structure Documentation	64
6.2 Port HLD	65
6.2.1 Detailed Description	65
6.2.2 Macro Definition Documentation	66
6.2.3 Function Reference	70
6.3 Port IPL	74
6.3.1 Detailed Description	74
6.3.2 Data Structure Documentation	75
6.3.3 Macro Definition Documentation	76
6.3.4 Types Reference	77
6.3.5 Enum Reference	77
6.3.6 Function Reference	84
6.3.7 Variable Documentation	88
6.4 TSPC IPL	89
6.4.1 Detailed Description	89
6.4.2 Function Reference	89



Chapter 1

Revision History

Revision	Date	Author	Description
1.0	31.03.2023	NXP RTD Team	S32K3 Real-Time Drivers AUTOSAR 4.4 & R21-11 Version 3.0.0

Chapter 2

Introduction

- [Supported Derivatives](#)
- [Overview](#)
- [About This Manual](#)
- [Acronyms and Definitions](#)
- [Reference List](#)

This User Manual describes NXP Semiconductor AUTOSAR Port for S32K3XX. 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:

- s32k310_mqfp100
- s32k310_lqfp48
- s32k311_mqfp100 / MWCT2015S_mqfp100
- s32k311_lqfp48
- s32k312_mqfp100 / MWCT2016S_mqfp100
- s32k312_mqfp172 / MWCT2016S_mqfp172
- s32k314_mqfp172
- s32k314_mapbga257
- s32k322_mqfp100 / MWCT2D16S_mqfp100
- s32k322_mqfp172 / MWCT2D16S_mqfp172

- s32k324_mqfp172 / MWCT2D17S_mqfp172
- s32k324_mapbga257
- s32k341_mqfp100
- s32k341_mqfp172
- s32k342_mqfp100
- s32k342_mqfp172
- s32k344_mqfp172
- s32k344_mapbga257
- s32k394_mapbga289
- s32k396_mapbga289
- s32k358_mqfp172
- s32k358_mapbga289
- s32k328_mqfp172
- s32k328_mapbga289
- s32k338_mqfp172
- s32k338_mapbga289
- s32k348_mqfp172
- s32k348_mapbga289
- s32m274_lqfp64
- s32m276_lqfp64

All of the above microcontroller devices are collectively named as S32K3.

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 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
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	S32K3xx Reference Manual, Rev.6, Draft B, 01/2023
		S32K39 and S32K37 Reference Manual, Rev. 2 Draft A, 11/2022
		S32M27x Reference Manual, Rev.2, Draft A, — 02/2023
3	Datasheet	S32K3xx Data Sheet, Rev. 6, 11/2022
		S32K396 Data Sheet, Rev. 1.1 — 08/2022
		S32M2xx Data Sheet, Rev. 2 RC — 12/2022
4	Errata	S32K358_0P14E Mask Set Errata – Rev. 28, 9/2022
		S32K396_0P40E Mask Set Errata, Rev. DEC2022, 12/2022
		S32K311_0P98C Mask Set Errata, Rev. 6/March/2023, 3/2023
		S32K312: Mask Set Errata for Mask 0P09C, Rev. 25/April/2022
		S32K342: Mask Set Errata for Mask 0P97C, Rev. 10, 11/2022
		S32K3x4: Mask Set Errata for Mask 0P55A/1P55A, Rev. 14/Oct/2022

Chapter 3

Driver

- [Requirements](#)
- [Driver Design Summary](#)
- [Hardware Resources](#)
- [Deviations from Requirements](#)
- [Driver Limitations](#)
- [Driver usage and configuration tips](#)
- [Runtime errors](#)
- [Symbolic Names Disclaimer](#)

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 SIUL2.

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 S32K3xx_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. PTB[3]), 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 the column 'CR' there is a number which represents the numeric value of the Multiplexed Signal Configuration Register. Note down this number (eg. 35)
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.	The type Port_PinDirectionType shall be of enumeration type having range as PORT_PIN_IN, PORT_PIN_OUT and PORT_PIN_INOUT.
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	<p>"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 - "</p>	Currently implemented in a different 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	<p>"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_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_MODE_DIO_WDG - Port Pin configured for DIO. It shall be used under control of the watchdog driver. - PORT_PIN_MODE_FLEXRAY - Port Pin used for FlexRay - PORT_PIN_MODE_ICU - Port Pin used by ICU - PORT_PIN_MODE_LIN - Port Pin used for LIN - PORT_PIN_MODE_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-POST-BUILD - Scope / Dependency - scope: local - "</p>	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

1. ADC interleave functionality - internal routing of signal to ADC from external PINS (i.e.(ADC1_S14 routed from PTC5 or PTB0) is not yet supported in PORT (ARTD-6142)).
2. During run-time, changing two or more pins with the same IMCR and the Virtual Wrapper was enable will lead to the pdac slot of IMCR register on each pin may be different and cause HW error.
3. Multiple configurations was not supported in S32K3XX RTM 3.0.0. If uses multiple variants, all variants will have the same configuration parameters.

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 S32K3XX IO muxing table Excel file attached to the Reference Manual pdf. Note when configuring the pins: The user can set the pin sequentially to be able to read the result correctly(for this the user can use a semaphore written by core 0 and read by core 1).

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. Go to PortEcucPartitionRef container inside the Port plugin where you want to add a new partition
 - 2. Open the IOMUX Excel file
 - 3. Go to 'IO Signal Table' sheet
 - 4. Identify the microcontroller pin you want to use (eg. PTB[3]), 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 the column 'CR' there is a number which represents the numeric value of the Multiplexed Signal Configuration Register. Note down this number (eg. 35)
 - 5. Go to port container inside the Port plugin where you want to add the pin
 - 6. Add a new PortPin in the port container list then double click the newly added PortPin to open it's properties
 - 7. Go to the 'PortPin MSCR' attribute and type the number noted down at step A.4
 - 8. Go to the 'PortPin Mode' attribute and choose the functionality you want to use for the selected pin
 - 9. Look at the other attributes of the PortPin and set them to the desired values
 - 10. Go to PortPinEcucPartitionRef container inside the PortPin where you want to add a new partition
- 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 MSCRs, 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 IOMUX Excel file
 - 2. Go to 'IO Signal Table' sheet

- 3. Identify the microcontroller pin you want the Port driver to not touch (eg. PTB[3]), 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 the column 'CR' there is a number which represents the numeric value of the Multiplexed Signal Configuration Register. Note down this number (eg. 35)
- 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
- 7. Go to the 'PortPin Siu2 Instance' attribute and select the SIUL2 instance the pin belongs to
- 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.

Important note

- According to new Design Studio framework, Port HLD with get configuration in the Pins tool, it means the Pin tool should be always enabled. Please following these steps:
 - Use pins tool to configure a signal, for examples: PTE5, mode gpio,133.
 - Refer to the IOMux to get the MSCR register correspond with PTE5: MSCR 133 (S32K396).
 - Open peripheral tool, add Port component, add value PortPin Mscr = 133.
 - With Boolean nodes, will raise errors when un-match configure between Pins tool and Port HLD.
- In order to be able to use the debug capabilities, the JTAG and Reset 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:
 - Reset_b having PortPin Mscr set to 5 and SIUL2 Instance set to SIUL2_0
 - JTAG_TMS having PortPin Mscr set to 4 and SIUL2 Instance set to SIUL2_0
 - JTAG_TDO having PortPin Mscr set to 10 and SIUL2 Instance set to SIUL2_0
 - JTAG_TCK having PortPin Mscr set to 68 and SIUL2 Instance set to SIUL2_0
 - JTAG_TDI having PortPin Mscr set to 69 and SIUL2 Instance set to SIUL2_0

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.

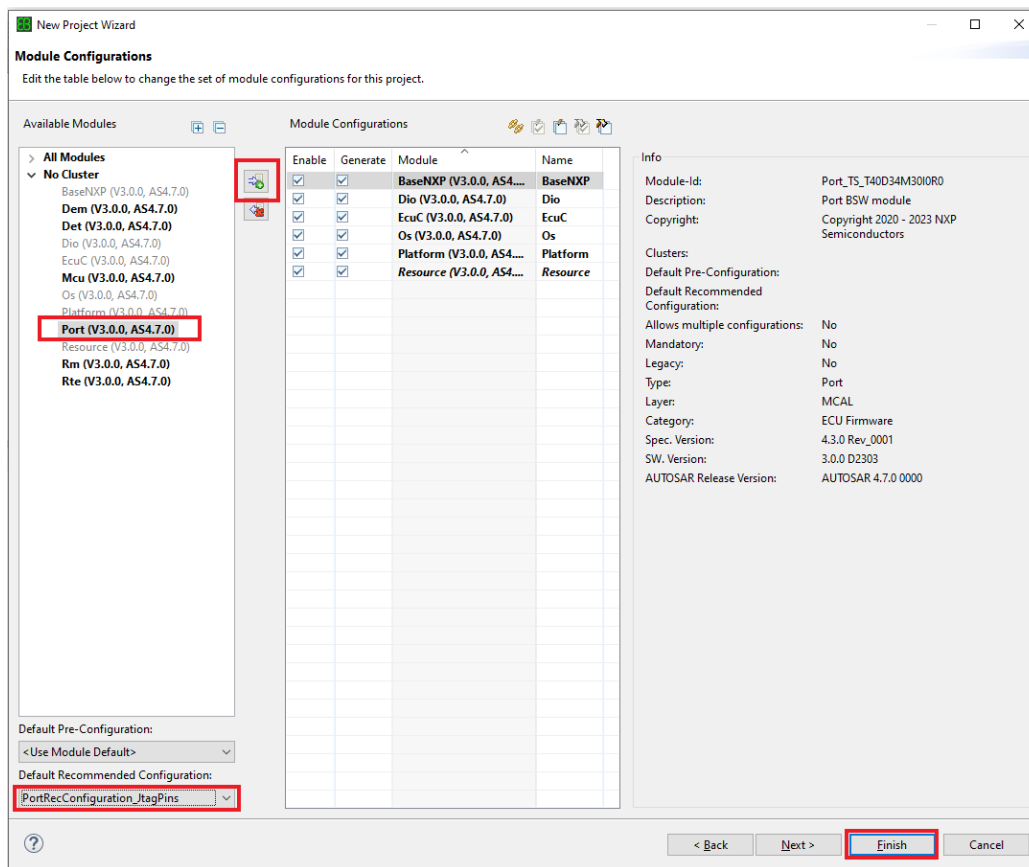


Figure 3.1 How to configure JTAG pins

- When VIRT_WRAPPER is enabled, the PDAC3 slot should not be used because it is default interface for HSE access. Any configuration through PDAC3 slot will not affect to SIUL2 registers.

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'.
- PortSetPinModeDoesNotTouchGpioLevel. 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.
- Port Code Size Optimization option is being used to reduce the size of the code. This option is configurable on/off and it affects the functionality of the [Port_SetPinMode\(\)](#) and [Port_ResetPinMode\(\)](#) API. When checked, the value of two nodes 'Port SetPinMode API' and 'Port Reset Pin Mode API' must be disabled and those two APIs will not be able to use.

Running multicore for multiple image (.elf) files

- 1. Modify linker file and memory map to match the shared memory allocation

```

MEMORY
{
    int_itcm : ORIGIN = 0x00000000, LENGTH = 0x00000000 /* 0KB - Not Supported */
    int_dtcmm : ORIGIN = 0x20000000, LENGTH = 0x00010000 /* 64K */
    int_sram_shareable : ORIGIN = 0x22C00000, LENGTH = 0x00004000 /* 16KB shareable for HSE only */
    int_sram_c0 : ORIGIN = 0x34000000, LENGTH = 0x00178000 /* 1.5MB - 8KB */
    int_sram_stack_c0 : ORIGIN = 0x34178000, LENGTH = 0x00002000 /* 8KB */
    int_sram_no_cacheable_c0 : ORIGIN = 0x34180000, LENGTH = 0x00080000 /* 512KB, needs to include int_results */
    ram_end_c0 : ORIGIN = 0x34200000, LENGTH = 0x00000000 /* End of core 0 ram */
    int_sram_c1 : ORIGIN = 0x34200000, LENGTH = 0x00178000 /* 1.5MB - 8KB */
    int_sram_stack_c1 : ORIGIN = 0x34378000, LENGTH = 0x00002000 /* 8KB */
    int_sram_no_cacheable_c1 : ORIGIN = 0x34380000, LENGTH = 0x00080000 /* 512KB, needs to include int_results */
    ram_end_c1 : ORIGIN = 0x34400000, LENGTH = 0x00000000 /* End of core 1 ram */
    int_sram_c2 : ORIGIN = 0x34400000, LENGTH = 0x00178000 /* 1.5MB - 8KB */
    int_sram_stack_c2 : ORIGIN = 0x34578000, LENGTH = 0x00002000 /* 8KB */
    int_sram_no_cacheable_c2 : ORIGIN = 0x34580000, LENGTH = 0x00080000 /* 512KB, needs to include int_results */
    ram_end_c2 : ORIGIN = 0x34600000, LENGTH = 0x00000000 /* End of core 2 ram */
    int_sram_shareable_dio_port : ORIGIN = 0x34600000, LENGTH = 0x00004000 /* 16KB shareable application cores for only Dio and Port */
    LLCE_CAN_SHAREDMEMORY : ORIGIN = 0x43800000, LENGTH = 0x3C800
    LLCE_LIN_SHAREDMEMORY : ORIGIN = 0x4383C800, LENGTH = 0xa0
    LLCE_BOOT_END : ORIGIN = 0x4383C8A0, LENGTH = 0x50
    LLCE_MEAS_SHAREDMEMORY : ORIGIN = 0x4384FFDF, LENGTH = 0x20
}

```

Figure 3.2 Create new section of memory on linker

```

    .mcval_shared_dio_port_bss : ALIGN(16) :> int_sram_shareable_dio_port
    .mcval_shared_dio_port_data : ALIGN(16) :> .
    .mcval_shared_dio_port_const : ALIGN(16) :> .

```

Figure 3.3 Define and align sections for bss data and const


```

#ifdef PORT_START_SEC_CONFIG_DATA_8
.../**
...*.@file.Port_MemMap.h
...*/
...#undef PORT_START_SEC_CONFIG_DATA_8
...#define ENTERED_PORT_START_SEC_CONFIG_DATA_8
...#ifndef MEMMAP_MATCH_ERROR
...#define MEMMAP_MATCH_ERROR
...#else
...#ifndef PORT_STOP_SEC_CONFIG_DATA_8
...#error "MemMap.h, no valid matching start-stop section defined."
...#endif
...#endif
.../**
...*.@file.Port_MemMap.h
...*/
...#undef MEMMAP_ERROR
...#pragma ghs section rodata=".mcal_shared_dio_port_const"
#endif

```

Figure 3.4 Use the new sections which defined in the linker files

- 2. Prepare at least 2 image (.elf) files

ELF #1				
Name	Date modified	Type	✓	Size
 Port_TS_001_cfgPB_CORE4.elf	9 Feb 2023 10:56 am	ELF File		505 KB


ELF #2				
Name	Date modified	Type	✓	Size
 Port_TS_200_cfgCORE4_5.elf	9 Feb 2023 11:07 am	ELF File		478 KB

Figure 3.5 Prepare 2 image (.elf) files

- 3. Normally, in debug section, the .elf will be loaded automatically but following the changes from build_env branch we have to load them manually by choosing the directory of elf1 then elf2 in turn

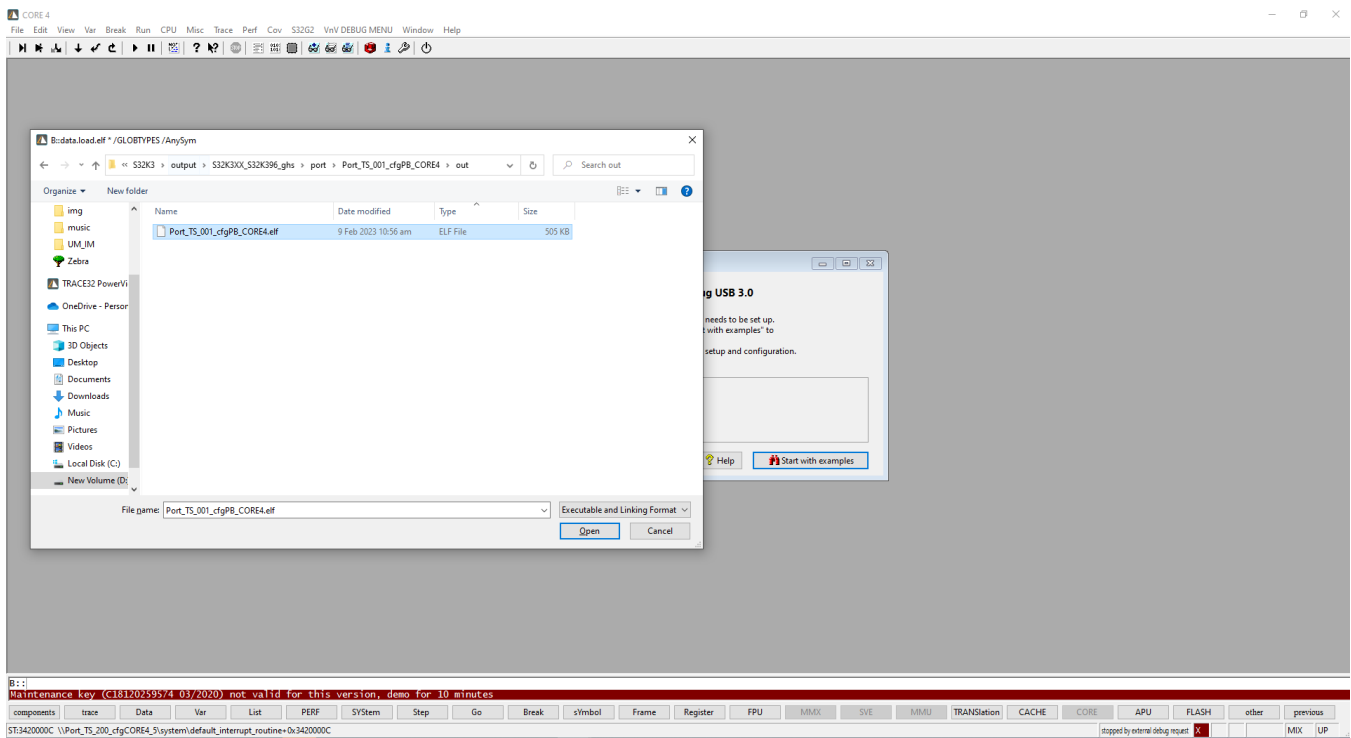


Figure 3.6 Choose the image (.elf) #1

Type: "go main" in T32 debug section

Driver

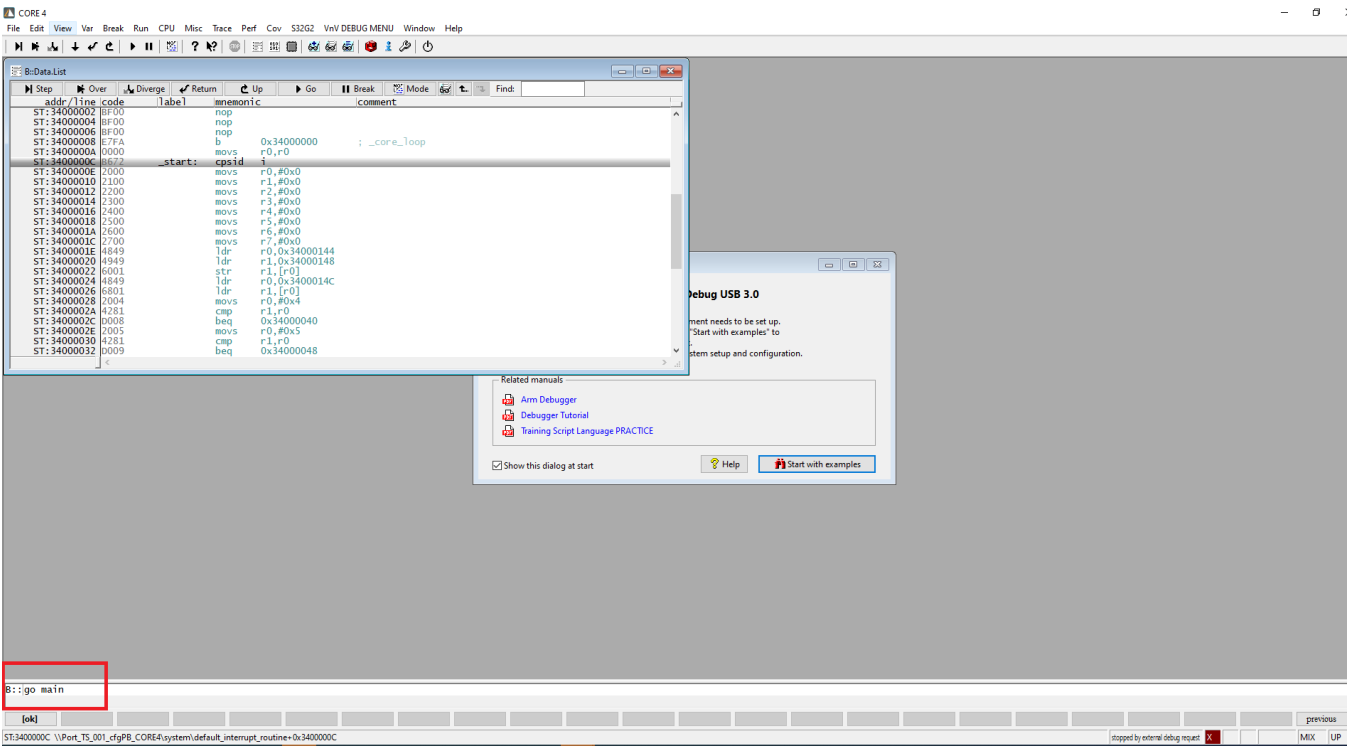


Figure 3.7 Go to main

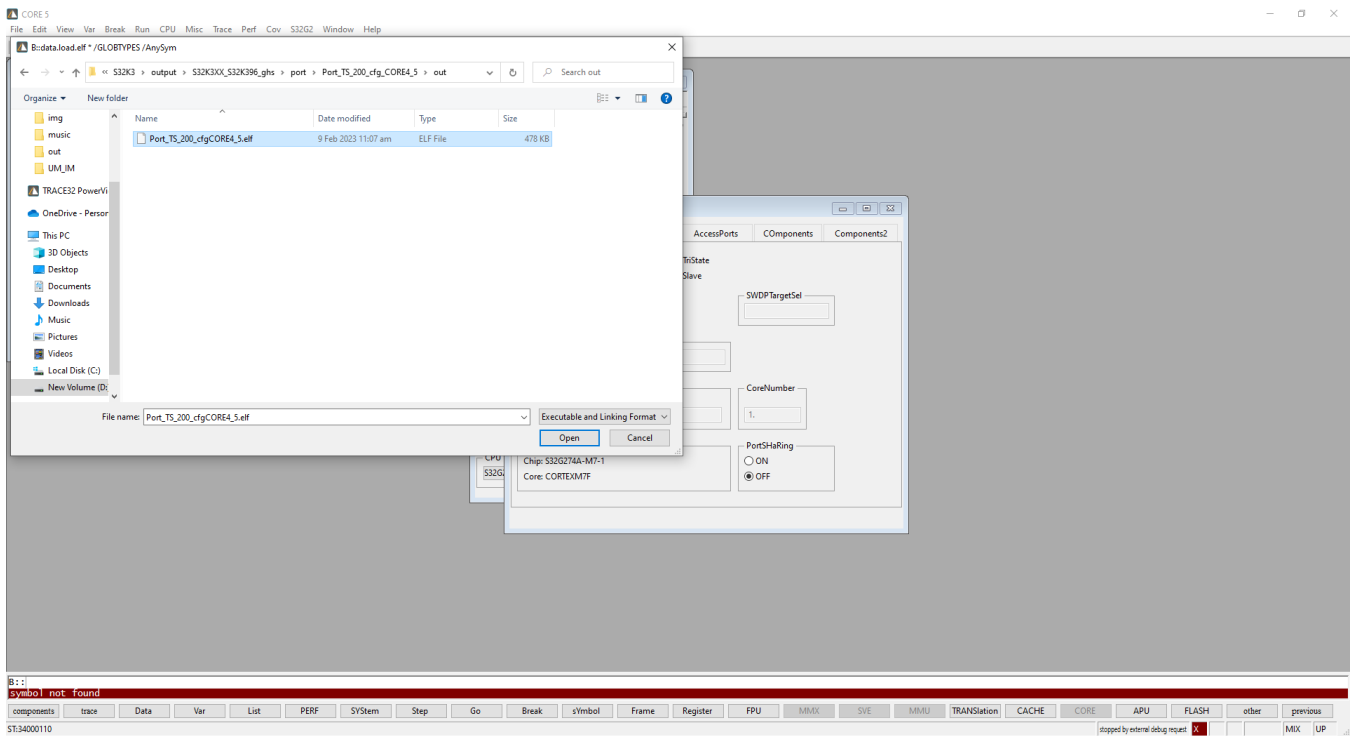


Figure 3.8 Choose the image (.elf) #2

Type: "go main" in T32 debug section

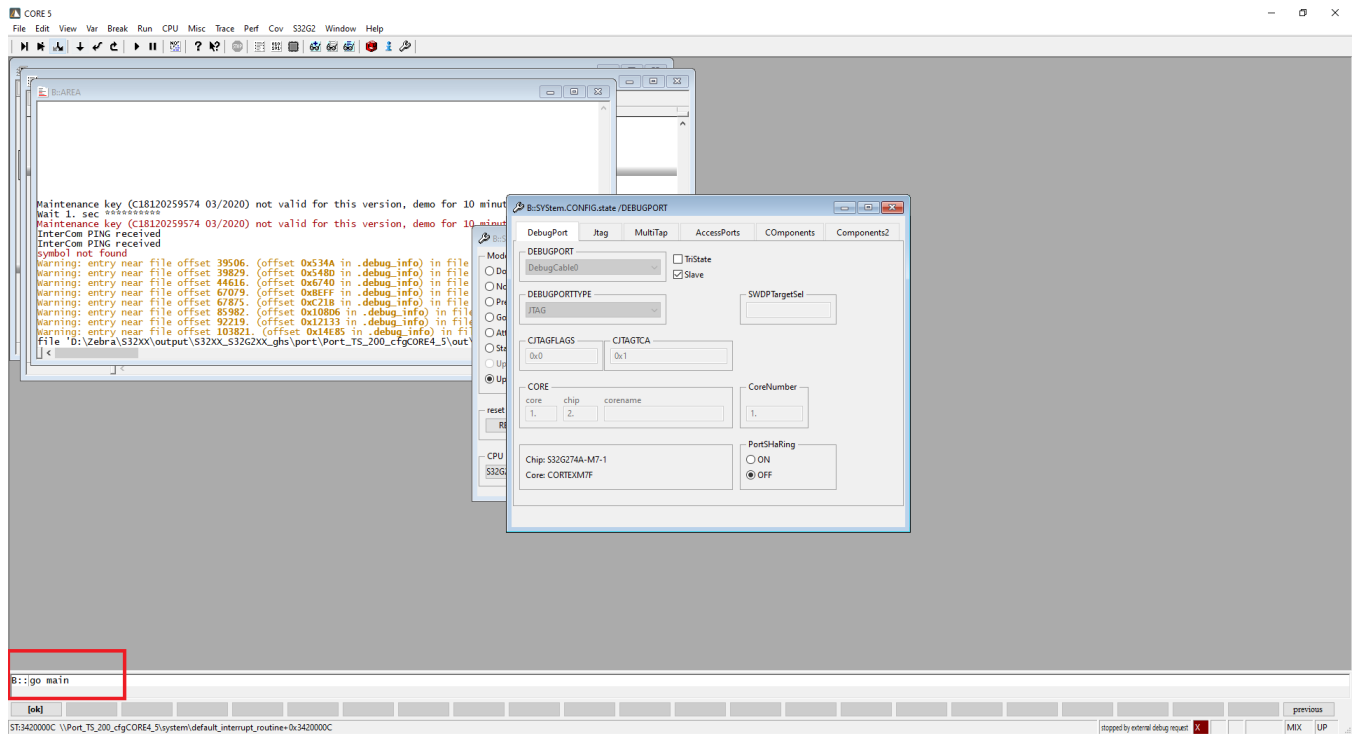


Figure 3.9 Go to main

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_SetPinDirection	PORT_E_DIRECTION_UNCHANGEABLE	Port Pin not configured as changeable
Port_SetPinMode	PORT_E_MODE_UNCHANGEABLE	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_MODE	API Port_SetPinMode service called when mode is unchangeable.
PORT_SETPINDIRECTION_ID	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_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 [PortPinPue](#)
 - Parameter [PortPinPus](#)
 - Parameter [PortPinDirection](#)
 - Parameter [PortPinLevelValue](#)
 - * Container [PortContainer](#)
 - Parameter [PortNumberOfPortPins](#)
 - Container [PortPin](#)
 - Parameter [PortPinPue](#)
 - Parameter [PortPinPus](#)
 - Parameter [PortPinSafeMode](#)
 - Parameter [PortPinDse](#)
 - Parameter [PortPinWithReadBack](#)
 - Parameter [PortPinPke](#)
 - Parameter [PortPinIfe](#)
 - Parameter [PortPinDirectionChangeable](#)
 - Parameter [PortPinModeChangeable](#)
 - Parameter [PortPinInvertControl](#)
 - Parameter [PortPinSiul2Instance](#)
 - Parameter [PortPinId](#)
 - Parameter [PortPinPcr](#)
 - Parameter [PortPinDirection](#)
 - Parameter [PortPinInitialMode](#)
 - Parameter [PortPinMode](#)
 - Parameter [PortPinLevelValue](#)
 - Parameter [PortPinSlewRate](#)
 - Parameter [OBEGroupSelect](#)
 - Parameter [MscrPdacSlot](#)

- Parameter [ImcrPdacSlot](#)
- Reference [PortPinEcucPartitionRef](#)
- Container [IGFSettings](#)
- Parameter [IGF_FGEN](#)
- Parameter [IGFChannel](#)
- Parameter [IGF_IMM](#)
- Parameter [IGF_PSSEL](#)
- Parameter [IGF_FPRE](#)
- Parameter [IGF_FFM](#)
- Parameter [IGF_RFM](#)
- Parameter [IGF_RTH](#)
- Parameter [IGF_FTH](#)
- * Container [UnTouchedPortPin](#)
 - Parameter [PortPinSiul2Instance](#)
 - Parameter [PortPinPcr](#)
- * Container [UntouchedIMCR](#)
 - Parameter [IMCRSiul2Instance](#)
 - Parameter [UntouchedPortPinImcr](#)
- Container [PortGeneral](#)
 - * Parameter [PortDevErrorDetect](#)
 - * Parameter [SIUL2PortIPDevErrorDetect](#)
 - * Parameter [PortSetPinDirectionApi](#)
 - * Parameter [PortSetPinModeApi](#)
 - * Parameter [PortVersionInfoApi](#)
 - * Parameter [PortSetPinModeDoesNotTouchGpioLevel](#)
 - * Parameter [PortSetAsUnusedPinApi](#)
 - * Parameter [PortResetPinModeApi](#)
 - * Parameter [PortEnableUserModeSupport](#)
 - * Parameter [PortMulticoreSupport](#)
 - * Parameter [PortTspcSupport](#)
 - * Parameter [SignalInversionConfigEnable](#)
 - * Parameter [VirtWrapperSupport](#)
 - * Parameter [PortCodeSizeOptimization](#)
 - * 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	0
upperMultiplicity	1
postBuildVariantSupport	true
supportedConfigVariants	VARIANT-POST-BUILD, VARIANT-PRE-COMPILE

4.2 Container PortConfigSet

This container contains a configuration of the PORT driver / SIUL2 module.

Included subcontainers:

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

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 PortPinPue

Enables the pull function. Used only when the associated destination is a chip pin.

Checked box means the Pull Up or Pull Down configuration selected by 'PortPin PUS' is enabled for the pin.

Unchecked box means the Pull Up or Pull Down configuration selected by 'PortPin PUS' is disabled for the pin.

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

Property	Value
postBuildVariantValue	true
valueConfigClasses	VARIANT-POST-BUILD: POST-BUILD
	VARIANT-PRE-COMPILE: PRE-COMPILE
defaultValue	false

4.5 Parameter PortPinPus

Determines whether the pull function is a pullup or pulldown when the pull function is enabled by the 'PortPin Pull Enable' field. Used only when the associated destination is a chip pin.

Checked box means the Pull Up configuration is set. Unchecked box means the Pull Down 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

4.6 Parameter PortPinDirection

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.

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_IN
literals	['PORT_PIN_IN', 'PORT_PIN_OUT', 'PORT_PIN_DISABLED']

4.7 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']

4.8 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

4.9 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

4.10 Container PortPin

Configuration of the individual port pins.

Included subcontainers:

- [IGFSettings](#)

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 PortPinPue

Enables the pull function. Used only when the associated destination is a chip pin.

Checked box means the Pull Up or Pull Down configuration selected by 'PortPin PUS' is enabled for the pin.

Unchecked box means the Pull Up or Pull Down configuration selected by 'PortPin PUS' is disabled for the pin.

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

4.12 Parameter PortPinPus

Determines whether the pull function is a pullup or pulldown when the pull function is enabled by the 'PortPin Pull Enable' field. Used only when the associated destination is a chip pin.

Checked box means the Pull Up configuration is set. Unchecked box means the Pull Down 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

4.13 Parameter PortPinSafeMode

Enable/Disable Safe Mode for the pin. Checked box means the Safe Mode is enabled.

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 PortPinDse

Enable Drive Strength for the configured Pin.

Checked box means the Drive Strength is enabled.

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 PortPinWithReadBack

Enables/Disables the read back possibility for this pin. Checked box means the Read Back is enabled.

When ReadBack is enabled, the Input Bufer of the pin gets enabled by setting the IBE bit in the MSCR (PCR) of the pin. Some alternate functions working as inputs might require having the IBE set to 1, so check this box in order to achieve this.

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 PortPinPke

Enables/disables pad keeping in standby mode. When padkeeping is enabled in standby mode, the pad output path latches the value till standby exit.

Checked box means pad keeping in stanby mode is enabled for the pin.

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

4.17 Parameter PortPinIfc

Enables/disables Input Filter.

Checked box means Input Filter is enabled for the pin.

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
default Value	false

4.18 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 changeability 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
default Value	true

4.19 Parameter PortPinModeChangeable

Parameter to indicate if the mode of a port pin is changeable during runtime.

Checked box: Port Pin mode changeable allowed.

Unchecked box: Port Pin mode changeable not permitted

The function for changing the pin modes is not supported by the safety implementation.

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

4.20 Parameter PortPinInvertControl

Invert the signal selected by SSS before transmitting it to the associated destination.

Checked box means the signal will be inverted.

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

4.21 Parameter PortPinSiul2Instance

Selects one of the SIULs instances available on the platform to configure the current pin from.

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	SIUL2_0
literals	['SIUL2_0']

4.22 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

4.23 Parameter PortPinPcr

Used to specify port configuration register: SIUL I/O Pin Multiplexed Signal Configuration Registers (MSCR number).

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	240
min	0

4.24 Parameter PortPinDirection

Selects the direction of the pin (IN, OUT , INOUT or 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.

If your Alternative Function is an input functionality that requires the IBE bit to be set in the MSCR, please select the checkbox 'PortPinWithReadback'.

If direction is PORT_PIN_HIGH_Z, there will be no initial direction setting.

Property	Value
type	ECUC-ENUMERATION-PARAM-DEF
origin	AUTOSAR_ECUC
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	true

Property	Value
valueConfigClasses	VARIANT-POST-BUILD: POST-BUILD
	VARIANT-PRE-COMPILE: PRE-COMPILE
defaultValue	PORT_PIN_HIGH_Z
literals	['PORT_PIN_IN', 'PORT_PIN_OUT', 'PORT_PIN_INOUT', 'PORT_PIN_HIGH_Z']

4.25 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_FUNC_MODE', 'PORT_ALT3_FUNC_MODE', 'PORT_ANALOG_INPUT_MODE', 'PORT_ONLY_INPUT_MODE', 'PORT_EXTRA_INPUT_MODE']

4.26 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

Property	Value
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

38 CH_16 - I_N-IN; EMIO5_1 EMIOS_1 CH_16 - I_{OUT}-OUT; EMIO5_1 EMIO5_1
S32K3 PORT Driver
 IOS_1 CH_17 - I_N-INOUT; EMIO5_1 EMIO5_1 CH_17 - I_N-IN; EMIO5_1

Property	Value
----------	-------

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 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	SLEW_RATE_NOT_AVAILABLE
literals	['FASTEST_SETTING', 'SLOWEST_SETTING', 'SLEW_RATE_NOT_A↵VAILABLE']

4.29 Parameter OBEGroupSelect

Configures OBE Group Select 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	NO_OBE_GROUP
literals	['NO_OBE_GROUP', 'OBE_GROUP1', 'OBE_GROUP2']

4.30 Parameter MscrPdacSlot

Select the PDAC slot for PortPin controlling MSCR, GPDO, PGPDO, and MPGPDO

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	VIRTUAL_WRAPPER_PDAC0
literals	['VIRTUAL_WRAPPER_PDAC0', 'VIRTUAL_WRAPPER_PDAC1', 'VIRTUAL_WRAPPER_PDAC2', 'VIRTUAL_WRAPPER_PDAC3']

4.31 Parameter ImcrPdacSlot

Select the PDAC slot for PortPin controlling IMCR

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	VIRTUAL_WRAPPER_PDAC0
literals	['VIRTUAL_WRAPPER_PDAC0', 'VIRTUAL_WRAPPER_PDAC1', 'VIRTUAL_WRAPPER_PDAC2', 'VIRTUAL_WRAPPER_PDAC3']

4.32 Reference PortPinEcucPartitionRef

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.

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

4.33 Container IGFSettings

Configure the parameters for IGF (Input Glitch Filter) IP.

Included subcontainers:

- None

Property	Value
type	ECUC-PARAM-CONF-CONTAINER-DEF
lowerMultiplicity	0
upperMultiplicity	1
postBuildVariantMultiplicity	false
multiplicityConfigClasses	VARIANT-POST-BUILD: POST-BUILD
	VARIANT-PRE-COMPILE: PRE-COMPILE

4.34 Parameter IGF_FGEN

This control bit enables the filter operation, meaning that if the filter is enabled the filtering selected types are applied to the rising and falling edges of the input signal. If the filter is not enabled, then the filter output remains unchanged independent of the filter input signal.

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

4.35 Parameter IGFChannel

Selects the IGF channel that needs to be configured.

Property	Value
type	ECUC-ENUMERATION-PARAM-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1

Property	Value
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	true
valueConfigClasses	VARIANT-POST-BUILD: POST-BUILD
	VARIANT-PRE-COMPILE: PRE-COMPILE
defaultValue	PleaseSelect
literals	['PleaseSelect', 'IGF0_3', 'IGF0_4', 'IGF0_11', 'IGF0_13', 'IGF0_16', 'IGF0_17', 'IGF0_6', 'IGF0_18', 'IGF0_2', 'IGF0_7', 'IGF0_5', 'IGF0_21', 'IGF0_19', 'IGF0_24', 'IGF0_0', 'IGF0_25', 'IGF0_8', 'IGF0_29', 'IGF0_30', 'IGF0_15', 'IGF0_22', 'IGF0_28', 'IGF0_23', 'IGF0_20', 'IGF0_27', 'IGF0_14', 'IGF0_12', 'IGF0_10', 'IGF0_26', 'IGF0_9', 'IGF0_31']

4.36 Parameter IGF_IMM

The IMM bit controls the propagation of an edge through the filter. If asserted, this bit defines that an edge at the filter input propagates through the filter output independent of the prescaler settings. In bypass mode the propagation delay is three system clock cycles. If negated, the signal propagation depends upon the prescaler settings.

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

4.37 Parameter IGF_PSSEL

The PSSEL selects if the prescaler used by the Input Glitch Filter is from internal source (internal prescaler counter) or from external source.

Property	Value
type	ECUC-ENUMERATION-PARAM-DEF

Property	Value
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	PleaseSelect
literals	['PleaseSelect', 'Internal', 'External']

4.38 Parameter IGF_FPRE

The prescaler defines the rate of the filter counter FCOUNT. The prescaler defines a division over the system clock, thus it is required to know the system clock frequency in order to define the prescaler output rate.

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	511
min	0

4.39 Parameter IGF_FFM

This bit field selects the filter type for the filter input signal falling edge.

- Bypass: edge is propagated to filter output without any filtering in three system clock cycles
- Windowing: windowing filter is selected for the falling edge
- Integrating: integrating filter is selected for the falling edge

- IntegratingHold: integrating-hold filter is selected for the falling edge
- WindowingWithPostSample : windowing filter with post sample is selected for the falling edge

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	Bypass
literals	['Bypass', 'Windowing', 'Integrating', 'IntegratingHold', 'WindowingWithPostSample']

4.40 Parameter IGF_RFM

This bit field selects the filter type for the filter input signal rising edge.

- Bypass: edge is propagated to filter output without any filtering in three system clock cycles
- Windowing: windowing filter is selected for the rising edge
- Integrating: integrating filter is selected for the rising edge
- IntegratingHold: integrating-hold filter is selected for the rising edge
- WindowingWithPostSample : windowing filter with post sample is selected for the falling edge

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	Bypass
literals	['Bypass', 'Windowing', 'Integrating', 'IntegratingHold', 'WindowingWithPostSample']

4.41 Parameter IGF_RTH

This bit field defines the filter counter threshold when a rising edge is being filtered.

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	16777215
min	0

4.42 Parameter IGF_FTH

This bit field defines the filter counter threshold when a falling edge is being filtered.

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	16777215
min	0

4.43 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

4.44 Parameter PortPinSiul2Instance

Selects one of the SIULs instances available on the platform to configure the current pin from.

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	SIUL2_0
literals	['SIUL2_0']

4.45 Parameter PortPinPcr

Used to specify port configuration register: SIUL I/O Pin Multiplexed Signal Configuration Registers (MSCR number).

Property	Value
type	ECUC-INTEGER-PARAM-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1

Property	Value
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	true
valueConfigClasses	VARIANT-POST-BUILD: POST-BUILD
	VARIANT-PRE-COMPILE: PRE-COMPILE
defaultValue	4
max	240
min	0

4.46 Container UntouchedIMCR

List containing IMCR of 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

4.47 Parameter IMCRSiul2Instance

Selects one of the SIULs instances available on the platform to configure the current IMCR.

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

Property	Value
valueConfigClasses	VARIANT-POST-BUILD: POST-BUILD
	VARIANT-PRE-COMPILE: PRE-COMPILE
defaultValue	SIUL2_0
literals	['SIUL2_0']

4.48 Parameter UntouchedPortPinImcr

Selects one of the IMCR will be Untouched

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	376
min	0

4.49 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

4.50 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

4.51 Parameter SIUL2PortIPDevErrorDetect

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

4.52 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
default Value	true

4.53 Parameter PortSetPinModeApi

The function for changing the pin modes is not supported by the safety implementation.

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
default Value	true

4.54 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

4.55 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

4.56 Parameter PortSetAsUnusedPinApi

The function void Port_SetAsUnusedPin shall configure the referenced pin with all the properties specified in the NotUsedPortPin container.

The function void Port_SetAsUsedPin shall configure the referenced pin with all the properties that were set during the Port_Init operation.

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

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

4.57 Parameter PortResetPinModeApi

The function Port_ResetPinMode shall revert the port pin mode of the referenced pin to the value that was set by Port_Init operation.

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

4.58 Parameter PortEnableUserModeSupport

When this parameter is enabled, the Port module will adapt to run from user mode, with the following measures:

- a) 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
- b) using 'call trusted function' stubs for all internal function calls that access registers requiring supervisor mode.

For more information, please see chapter 5.7 user mode Support in IM

Note: Implementation Specific Parameter.

Property	Value
type	ECUC-BOOLEAN-PARAM-DEF

Property	Value
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

4.59 Parameter PortMulticoreSupport

This parameter globally enables the possibility to support multicore. If this parameter is enabled, at least one EcucPartition needs to be defined (in all variants). The S32K311, S32K310, S32K312, S32K314, S32K341, S32K342, S32K344, S32K348, S32M276 and S32M274 derivatives will be treated as a single-core device.

Note 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	false
valueConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE VARIANT-POST-BUILD: PRE-COMPILE
defaultValue	false

4.60 Parameter PortTspcSupport

This parameter shall allow to support configure The Touch Sensing Pin Coupling

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

Property	Value
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

4.61 Parameter SignalInversionConfigEnable

This parameter globally enables the possibility to invert the signal of the port pin.

If this parameter is disabled, the signal of the port pin will not be allowed to invert.

If this parameter is enabled, the signal of the port pin will be possible to be inverted in case the invert configuration of each pin is also enabled for this pin.

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	false
valueConfigClasses	VARIANT-POST-BUILD: PRE-COMPILE
	VARIANT-PRE-COMPILE: PRE-COMPILE
defaultValue	false

4.62 Parameter VirtWrapperSupport

This parameter enables the virtualization wrapper functionality.

This is an implementation specific parameter.

RM Enable Virtual Wrapper feature is required to be enabled when Port Virtual Wrapper Support is enabled.

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
default Value	false

4.63 Parameter PortCodeSizeOptimization

This parameter reduces generation code size for only S32K311, S32K310 and S32K312 derivatives.

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
default Value	false

4.64 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

Property	Value
type	ECUC-REFERENCE-DEF
origin	AUTOSAR_ECUC
lowerMultiplicity	0

Property	Value
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/EcuDefs/EcuC/EcuPartitionCollection/EcuPartition

4.65 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

4.66 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
multiplicityConfigClasses	N/A
postBuildVariantValue	false

Property	Value
valueConfigClasses	VARIANT-POST-BUILD: PUBLISHED-INFORMATION
	VARIANT-PRE-COMPILE: PUBLISHED-INFORMATION
defaultValue	4
max	4
min	4

4.67 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

4.68 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
postBuildVariantValue	false
valueConfigClasses	VARIANT-POST-BUILD: PUBLISHED-INFORMATION
	VARIANT-PRE-COMPILE: PUBLISHED-INFORMATION

Property	Value
defaultValue	0
max	0
min	0

4.69 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

4.70 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
	VARIANT-PRE-COMPILE: PUBLISHED-INFORMATION
defaultValue	3
max	3
min	3

4.71 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

4.72 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
max	0
min	0

4.73 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.

Tresos Configuration Plug-in

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	

4.74 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
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:

IGF IPL	64
Port HLD	65
Port IPL	74
TSPC IPL	89

Chapter 6

Module Documentation

6.1 IGF IPL

6.1.1 Detailed Description

Data Structures

- struct [Igf_Port_Ip_ChannelConfigType](#)
Single Igf channel configuration. [More...](#)

6.1.2 Data Structure Documentation

6.1.2.1 struct Igf_Port_Ip_ChannelConfigType

Single Igf channel configuration.

This structure contains all configuration parameters of a single IGF channel identified by u8IgfInstance and u8IgfChannel.

Definition at line 100 of file Igf_Port_Ip_Types.h.

6.2 Port HLD

6.2.1 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_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_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.

6.2.2 Macro Definition Documentation

6.2.2.1 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.

6.2.2.2 PORT_E_PARAM_CONFIG

```
#define PORT_E_PARAM_CONFIG
```

The PORT module is not properly configured.

Definition at line 132 of file Port.h.

6.2.2.3 PORT_INSTANCE_ID

```
#define PORT_INSTANCE_ID
```

Instance ID of port driver.

Definition at line 139 of file Port.h.

6.2.2.4 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.

6.2.2.5 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.

6.2.2.6 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.

6.2.2.7 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.

6.2.2.8 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.

6.2.2.9 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.

6.2.2.10 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.

6.2.2.11 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.

6.2.2.12 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.

6.2.2.13 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.

6.2.2.14 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.

6.2.2.15 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.

6.2.2.16 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.

6.2.2.17 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.

6.2.2.18 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.

6.2.3 Function Reference

6.2.3.1 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	<i>Port_ConfigType</i>	* ConfigPtr Pointer to configuration (NULL_PTR if only one variant is used)
----	------------------------	---

Returns

void

6.2.3.2 Port_SetPinDirection()

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

Port_SetPinDirection.

Function used for changing the pin direction at runtime

Parameters

in		
----	--	--

6.2.3.3 Port_SetPinMode()

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

Port_SetPinMode.

Function used to change the pin mode at runtime.

Parameters

in		
----	--	--

6.2.3.4 Port_GetVersionInfo()

```
void Port_GetVersionInfo (
```

```
Std_VersionInfoType * versioninfo )
```

Port_GetVersionInfo.

Function used to read the driver version information

Parameters

in	<i>versioninfo</i>	pointer to structure that will contain the version information
----	--------------------	--

Returns

void

6.2.3.5 Port_RefreshPortDirection()

```
void Port_RefreshPortDirection (
    void )
```

Port_RefreshPortDirection.

function used to reset the direction of the pin

Returns

void

6.2.3.6 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

6.2.3.7 Port_SetAsUsedPin()

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

Port_SetAsUsedPin.

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

Returns

void

6.2.3.8 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

6.3 Port IPL

6.3.1 Detailed Description

Data Structures

- struct [Siul2_Port_Ip_PortType](#)
- struct [Siul2_Port_Ip_PinSettingsConfig](#)
Defines the converter configuration. [More...](#)

Macros

- `#define PORT_PIN_LEVEL_NOTCHANGED_U8`
Not changed port pin logic.
- `#define FEATURE_SIUL2_MAX_NUMBER_OF_INPUT`
SIUL2 module maximum number of input signal on a pin.
- `#define FEATURE_ADC_INTERLEAVE_MAX_MUX_MODE`
SIUL2 module maximum number of input signal on a pin.

Types Reference

- typedef uint8 [Siul2_Port_Ip_PortPinsLevelType](#)
Type of a port levels representation. Implements : [Siul2_Port_Ip_PortPinsLevelType](#).

Enum Reference

- enum [Siul2_Port_Ip_PortPullConfig](#)
Internal resistor pull feature selection Implements : [Siul2_Port_Ip_PortPullConfig](#).
- enum [Siul2_Port_Ip_PortMux](#)
Configures the Pin output muxing selection Implements : [Siul2_Port_Ip_PortMux](#).
- enum [Siul2_Port_Ip_PortInputFilter](#)
Configures the Pin filter enable Implements : [Siul2_Port_Ip_PortInputFilter](#).
- enum [Siul2_Port_Ip_PortPullKeep](#)
Configures the Pad keep enable Implements : [Siul2_Port_Ip_PortPullKeep](#).
- enum [Siul2_Port_Ip_PortInvert](#)
Configures signal invert for the pin Implements : [Siul2_Port_Ip_PortInvert](#).
- enum [Siul2_Port_Ip_PortOutputBuffer](#)
Configures the output buffer enable Implements : [Siul2_Port_Ip_PortOutputBuffer](#).
- enum [Siul2_Port_Ip_PortInputBuffer](#)
Configures the Input Buffer Enable field. Implements : [Siul2_Port_Ip_PortInputBuffer](#).
- enum [Siul2_Port_Ip_PortInputMux](#)
Configures the Pin input muxing selection Implements : [Siul2_Port_Ip_PortInputMux](#).
- enum [Siul2_Port_Ip_PortSafeMode](#)

- Configures the Safe Mode Control. Implements : Siul2_Port_Ip_PortSafeMode.*
- enum [Siul2_Port_Ip_PortSlewRateControl](#)

Configures the slew rate control. Implements : Siul2_Port_Ip_PortSlewRateControl.
- enum [Siul2_Port_Ip_PortDriveStrength](#)

Configures the drive strength. Implements : Siul2_Port_Ip_PortDriveStrength.
- enum [Siul2_Port_Ip_PortDirectionType](#)

Configures port direction.
- enum [Siul2_Port_Ip_AdcInterleaves](#)

Configures adc interleave mux mode.

Function Reference

- void [Siul2_Port_Ip_SetPullSel](#) ([Siul2_Port_Ip_PortType](#) *const base, uint16 pin, [Siul2_Port_Ip_PortPullConfig](#) pullConfig)

Configures the internal resistor.
- void [Siul2_Port_Ip_SetOutputBuffer](#) ([Siul2_Port_Ip_PortType](#) *const base, uint16 pin, boolean enable, [Siul2_Port_Ip_PortMux](#) mux)

Configures the output buffer and output signal.
- void [Siul2_Port_Ip_SetInputBuffer](#) ([Siul2_Port_Ip_PortType](#) *const base, uint16 pin, boolean enable, uint32 inputMuxReg, [Siul2_Port_Ip_PortInputMux](#) inputMux)

Configures the input buffer and input signal.
- [Siul2_Port_Ip_PortStatusType](#) [Siul2_Port_Ip_Init](#) (uint32 pinCount, const [Siul2_Port_Ip_PinSettingsConfig](#) config[])

Initializes the pins with the given configuration structure.
- void [Siul2_Port_Ip_SetPinDirection](#) ([Siul2_Port_Ip_PortType](#) *const base, uint16 pin, [Siul2_Port_Ip_PortDirectionType](#) direction)

Configures the pin with the values form the configuration structure.
- uint32 [Siul2_Port_Ip_RevertPinConfiguration](#) (const [Siul2_Port_Ip_PortType](#) *const base, uint16 pin)

This function configures the pin configuration with the values from the configuration structure.
- void [Siul2_Port_Ip_GetPinConfiguration](#) (const [Siul2_Port_Ip_PortType](#) *const base, [Siul2_Port_Ip_PinSettingsConfig](#) *config, uint16 pin)

This function shall return the value of the pin configuration register.

Variables

- const uint32 [Port_au32Siul2BaseAddr](#) []

Base address array for Siul2 instances.

6.3.2 Data Structure Documentation

6.3.2.1 struct [Siul2_Port_Ip_PortType](#)

PORT - Register Layout Typedef

Definition at line 338 of file [Siul2_Port_Ip_Types.h](#).

6.3.2.2 struct Siul2_Port_Ip_PinSettingsConfig

Defines the converter configuration.

This structure is used to configure the pins Implements : [Siul2_Port_Ip_PinSettingsConfig](#)

Definition at line 385 of file Siul2_Port_Ip_Types.h.

Data Fields

Type	Name	Description
SIUL2_Type *	base	The main SIUL2 base pointer.
uint32	pinPortIdx	Port pin number.
Siul2_Port_Ip_PortPullConfig	pullConfig	Internal resistor pull feature selection.
Siul2_Port_Ip_PortMux	mux	Pin output muxing selection.
Siul2_Port_Ip_PortSafeMode	safeMode	Configures the Safe Mode Control, apply for SIUL2_0/1
Siul2_Port_Ip_PortSlewRateControl	slewRateCtrlSel	Configures the Slew Rate Control field.
Siul2_Port_Ip_PortDriveStrength	driveStrength	Configures DSE
Siul2_Port_Ip_PortInputFilter	inputFilter	Configures IFE
Siul2_Port_Ip_PortPullKeep	pullKeep	Configures PKE
Siul2_Port_Ip_PortInvert	invert	Configures IFE
Siul2_Port_Ip_PortOutputBuffer	outputBuffer	Configures the Output Buffer Enable.
Siul2_Port_Ip_PortInputBuffer	inputBuffer	Configures the Input Buffer Enable.
Siul2_Port_Ip_AdcInterleaves	adcInterleaves[(2U)]	Configures the adc interleave mux modes.
Siul2_Port_Ip_PortInputMux	inputMux[(16U)]	Configures the input muxing
uint32	inputMuxReg[(16U)]	Configures the input muxing register. For the pins controlled by both SIUL2_0 and SIUL2_1 instances, refer the note for PINS_DRV_SetInputBuffer function
Siul2_Port_Ip_PortPinsLevelType	initValue	Initial value

6.3.3 Macro Definition Documentation

6.3.3.1 PORT_PIN_LEVEL_NOTCHANGED_U8

```
#define PORT_PIN_LEVEL_NOTCHANGED_U8
```

Not changed port pin logic.

Definition at line 183 of file Siul2_Port_Ip.h.

6.3.3.2 FEATURE_SIUL2_MAX_NUMBER_OF_INPUT

```
#define FEATURE_SIUL2_MAX_NUMBER_OF_INPUT
```

SIUL2 module maximum number of input signal on a pin.

Definition at line 102 of file Siul2_Port_Ip_Types.h.

6.3.3.3 FEATURE_ADC_INTERLEAVE_MAX_MUX_MODE

```
#define FEATURE_ADC_INTERLEAVE_MAX_MUX_MODE
```

SIUL2 module maximum number of input signal on a pin.

Definition at line 106 of file Siul2_Port_Ip_Types.h.

6.3.4 Types Reference

6.3.4.1 Siul2_Port_Ip_PortPinsLevelType

```
typedef uint8 Siul2_Port_Ip_PortPinsLevelType
```

Type of a port levels representation. Implements : Siul2_Port_Ip_PortPinsLevelType.

Definition at line 119 of file Siul2_Port_Ip_Types.h.

6.3.5 Enum Reference

6.3.5.1 Siul2_Port_Ip_PortPullConfig

```
enum Siul2_Port_Ip_PortPullConfig
```

Internal resistor pull feature selection Implements : Siul2_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 125 of file Siul2_Port_Ip_Types.h.

6.3.5.2 Siul2_Port_Ip_PortMux

enum `Siul2_Port_Ip_PortMux`

Configures the Pin output muxing selection Implements : `Siul2_Port_Ip_PortMux`.

Enumerator

PORT_MUX_AS_GPIO	corresponding pin is configured as GPIO
PORT_MUX_ALT1	chip-specific
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_ALT8	chip-specific
PORT_MUX_ALT9	chip-specific
PORT_MUX_ALT10	chip-specific
PORT_MUX_ALT11	chip-specific
PORT_MUX_ALT12	chip-specific
PORT_MUX_ALT13	chip-specific
PORT_MUX_ALT14	chip-specific
PORT_MUX_ALT15	chip-specific
PORT_MUX_NOT_AVAILABLE	chip-specific

Definition at line 136 of file Siul2_Port_Ip_Types.h.

6.3.5.3 Siul2_Port_Ip_PortInputFilter

enum [Siul2_Port_Ip_PortInputFilter](#)

Configures the Pin filter enable Implements : Siul2_Port_Ip_PortInputFilter.

Enumerator

PORT_INPUT_FILTER_DISABLED	IFE OFF
PORT_INPUT_FILTER_ENABLED	IFE ON
PORT_INPUT_FILTER_NOT_AVAILABLE	IFE NOT AVAILABLE

Definition at line 163 of file Siul2_Port_Ip_Types.h.

6.3.5.4 Siul2_Port_Ip_PortPullKeep

enum [Siul2_Port_Ip_PortPullKeep](#)

Configures the Pad keep enable Implements : Siul2_Port_Ip_PortPullKeep.

Enumerator

PORT_PULL_KEEP_DISABLED	PKE OFF
PORT_PULL_KEEP_ENABLED	PKE ON
PORT_PULL_KEEP_NOT_AVAILABLE	PKE NOT AVAILABLE

Definition at line 176 of file Siul2_Port_Ip_Types.h.

6.3.5.5 Siul2_Port_Ip_PortInvert

enum [Siul2_Port_Ip_PortInvert](#)

Configures signal invert for the pin Implements : Siul2_Port_Ip_PortInvert.

Enumerator

PORT_INVERT_DISABLED	INV OFF
PORT_INVERT_ENABLED	INV ON
PORT_INVERT_NOT_AVAILABLE	INV NOT AVAILABLE

Definition at line 189 of file Siul2_Port_Ip_Types.h.

6.3.5.6 Siul2_Port_Ip_PortOutputBuffer

enum `Siul2_Port_Ip_PortOutputBuffer`

Configures the output buffer enable Implements : Siul2_Port_Ip_PortOutputBuffer.

Enumerator

PORT_OUTPUT_BUFFER_DISABLED	Output buffer disabled
PORT_OUTPUT_BUFFER_ENABLED	Output buffer enabled
PORT_OUTPUT_BUFFER_NOT_AVAILABLE	Output buffer not available

Definition at line 201 of file Siul2_Port_Ip_Types.h.

6.3.5.7 Siul2_Port_Ip_PortInputBuffer

enum `Siul2_Port_Ip_PortInputBuffer`

Configures the Input Buffer Enable field. Implements : Siul2_Port_Ip_PortInputBuffer.

Enumerator

PORT_INPUT_BUFFER_DISABLED	Input buffer disabled
PORT_INPUT_BUFFER_ENABLED	Input buffer enabled
PORT_INPUT_BUFFER_NOT_AVAILABLE	Input buffer not available

Definition at line 212 of file Siul2_Port_Ip_Types.h.

6.3.5.8 Siul2_Port_Ip_PortInputMux

enum `Siul2_Port_Ip_PortInputMux`

Configures the Pin input muxing selection Implements : Siul2_Port_Ip_PortInputMux.

Enumerator

PORT_INPUT_MUX_ALT0	Chip-specific
PORT_INPUT_MUX_ALT1	Chip-specific
PORT_INPUT_MUX_ALT2	Chip-specific
PORT_INPUT_MUX_ALT3	Chip-specific
PORT_INPUT_MUX_ALT4	Chip-specific
PORT_INPUT_MUX_ALT5	Chip-specific
PORT_INPUT_MUX_ALT6	Chip-specific
PORT_INPUT_MUX_ALT7	Chip-specific
PORT_INPUT_MUX_ALT8	Chip-specific
PORT_INPUT_MUX_ALT9	Chip-specific
PORT_INPUT_MUX_ALT10	Chip-specific
PORT_INPUT_MUX_ALT11	Chip-specific
PORT_INPUT_MUX_ALT12	Chip-specific
PORT_INPUT_MUX_ALT13	Chip-specific
PORT_INPUT_MUX_ALT14	Chip-specific
PORT_INPUT_MUX_ALT15	Chip-specific
PORT_INPUT_MUX_NO_INIT	No initialization

Definition at line 236 of file Siul2_Port_Ip_Types.h.

6.3.5.9 Siul2_Port_Ip_PortSafeMode

enum `Siul2_Port_Ip_PortSafeMode`

Configures the Safe Mode Control. Implements : `Siul2_Port_Ip_PortSafeMode`.

Enumerator

PORT_SAFE_MODE_DISABLED	To drive pad in hi-z state using OBE = 0, when FCCU in fault state. The OBE will be driven by IP/SIUL when FCCU leaves the fault state.
PORT_SAFE_MODE_ENABLED	No effect on IP/SIUL driven OBE value
PORT_SAFE_MODE_NOT_AVAILABLE	Not available

Definition at line 261 of file Siul2_Port_Ip_Types.h.

6.3.5.10 Siul2_Port_Ip_PortSlewRateControl

enum `Siul2_Port_Ip_PortSlewRateControl`

Configures the slew rate control. Implements : `Siul2_Port_Ip_PortSlewRateControl`.

Enumerator

PORT_SLEW_RATE_FASTEST	Fmax=133 MHz(at 1.8V), 100 MHz (at 3.3V), apply for SIUL2_0/1
PORT_SLEW_RATE_SLOWEST	Fmax=83 MHz (at 1.8V), 63 MHz (at 3.3V), apply for SIUL2_0/1
PORT_SLEW_RATE_NOT_AVAILABLE	Not available

Definition at line 274 of file Siul2_Port_Ip_Types.h.

6.3.5.11 Siul2_Port_Ip_PortDriveStrength

enum `Siul2_Port_Ip_PortDriveStrength`

Configures the drive strength. Implements : `Siul2_Port_Ip_PortDriveStrength`.

Enumerator

PORT_DRIVE_STRENGTH_DISABLED	Disables DSE.
PORT_DRIVE_STRENGTH_ENABLED	Enables DSE.
PORT_DRIVE_STRENGTH_NOT_AVAILABLE	Not available.

Definition at line 297 of file Siul2_Port_Ip_Types.h.

6.3.5.12 Siul2_Port_Ip_PortDirectionType

enum `Siul2_Port_Ip_PortDirectionType`

Configures port direction.

Enumerator

<code>SIUL2_PORT_IN</code>	Sets port pin as input.
<code>SIUL2_PORT_OUT</code>	Sets port pin as output.
<code>SIUL2_PORT_IN_OUT</code>	Sets port pin as bidirectional.
<code>SIUL2_PORT_HI_Z</code>	Sets port pin as high_z.

Definition at line 345 of file Siul2_Port_Ip_Types.h.

6.3.5.13 Siul2_Port_Ip_AdcInterleaves

enum `Siul2_Port_Ip_AdcInterleaves`

Configures adc interleave mux mode.

Enumerator

<code>MUX_MODE_NOT_AVAILABLE</code>	Adc Interleave not available.
<code>MUX_MODE_EN_ADC1_S14_1</code>	Set bit ADC1_S14 to 1
<code>MUX_MODE_EN_ADC1_S15_1</code>	Set bit ADC1_S15 to 1
<code>MUX_MODE_EN_ADC0_S8_1</code>	Set bit ADC0_S8 to 1
<code>MUX_MODE_EN_ADC2_S8_1</code>	Set bit ADC2_S8 to 1
<code>MUX_MODE_EN_ADC0_S9_1</code>	Set bit ADC0_S9 to 1
<code>MUX_MODE_EN_ADC2_S9_1</code>	Set bit ADC2_S9 to 1
<code>MUX_MODE_EN_ADC1_S22_1</code>	Set bit ADC1_S22 to 1
<code>MUX_MODE_EN_ADC1_S23_1</code>	Set bit ADC1_S23 to 1
<code>MUX_MODE_EN_ADC1_S14_0</code>	With bits 15-0, only clear ADC1_S14 bit, the other bits set to 1

Enumerator

MUX_MODE_EN_ADC1_S15_0	With bits 15-0, only clear ADC1_S15 bit, the other bits set to 1
MUX_MODE_EN_ADC0_S8_0	With bits 15-0, only clear ADC0_S8 bit, the other bits set to 1
MUX_MODE_EN_ADC2_S8_0	With bits 15-0, only clear ADC2_S8 bit, the other bits set to 1
MUX_MODE_EN_ADC0_S9_0	With bits 15-0, only clear ADC0_S9 bit, the other bits set to 1
MUX_MODE_EN_ADC2_S9_0	With bits 15-0, only clear ADC2_S9 bit, the other bits set to 1
MUX_MODE_EN_ADC1_S22_0	With bits 15-0, only clear ADC1_S22 bit, the other bits set to 1
MUX_MODE_EN_ADC1_S23_0	With bits 15-0, only clear ADC1_S23 bit, the other bits set to 1

Definition at line 357 of file Siul2_Port_Ip_Types.h.

6.3.6 Function Reference

6.3.6.1 Siul2_Port_Ip_SetPullSel()

```
void Siul2_Port_Ip_SetPullSel (
    Siul2_Port_Ip_PortType *const base,
    uint16 pin,
    Siul2_Port_Ip_PortPullConfig pullConfig )
```

Configures the internal resistor.

This function configures the internal resistor.

Parameters

in	<i>base</i>	Port base pointer (PORTA, PORTB, PORTA_AE, etc.)
in	<i>pin</i>	Port pin number
in	<i>pullConfig</i>	The pull configuration

6.3.6.2 Siul2_Port_Ip_SetOutputBuffer()

```
void Siul2_Port_Ip_SetOutputBuffer (
    Siul2_Port_Ip_PortType *const base,
    uint16 pin,
    boolean enable,
    Siul2_Port_Ip_PortMux mux )
```

Configures the output buffer and output signal.

This function configures the output buffer for the pin and the path for output signal from module to pin

Parameters

in	<i>base</i>	Port base pointer (PORTA, PORTB, PORTA_AE, etc.)
in	<i>pin</i>	Port pin number
in	<i>enable</i>	Enable output buffer
in	<i>mux</i>	Pin muxing slot selection

6.3.6.3 Siul2_Port_Ip_SetInputBuffer()

```
void Siul2_Port_Ip_SetInputBuffer (
    Siul2_Port_Ip_PortType *const base,
    uint16 pin,
    boolean enable,
    uint32 inputMuxReg,
    Siul2_Port_Ip_PortInputMux inputMux )
```

Configures the input buffer and input signal.

This function configures the input buffer for the pin and the path for input signal from pin to module

Parameters

in	<i>base</i>	Port base pointer (PORTA, PORTB, PORTA_AE, etc.), NULL if disabling inputMux only
in	<i>pin</i>	Port pin number
in	<i>enable</i>	Enable input buffer
in	<i>inputMuxReg</i>	Pin muxing register slot selection
in	<i>inputMux</i>	Pin muxing slot selection

Note

: There are some pins controlled by both SIUL2_0, SIUL2_1, SIUL2_3, SIUL2_4 and SIUL2_5 instances In order to configure correctly and be consistent with other platforms, the inputMuxReg parameter of SIUL2_3 instance must be added 512 units. For example: The actual inputMuxReg is 10 then the value there must be (10 + 512)

6.3.6.4 Siul2_Port_Ip_Init()

```
Siul2_Port_Ip_PortStatusType Siul2_Port_Ip_Init (
    uint32 pinCount,
    const Siul2_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	<i>pinCount</i>	The number of configured pins in structure
in	<i>config</i>	The configuration structure

Returns

The status of the operation

6.3.6.5 Siul2_Port_Ip_SetPinDirection()

```
void Siul2_Port_Ip_SetPinDirection (
    Siul2_Port_Ip_PortType *const base,
    uint16 pin,
    Siul2_Port_Ip_PortDirectionType direction )
```

Configures the pin with the values form the configuration structure.

This function configures the pin configuration with the values form the configuration structure

Parameters

in	<i>base</i>	Port base pointer
in	<i>pin</i>	Port pin number
in	<i>direction</i>	The direction of pin

Returns

void

6.3.6.6 Siul2_Port_Ip_RevertPinConfiguration()

```
uint32 Siul2_Port_Ip_RevertPinConfiguration (
    const Siul2_Port_Ip_PortType *const base,
    uint16 pin )
```

This function configures the pin configuration with the values from the configuration structure.

This function configures the pin configuration with the values from the configuration structure

Parameters

in	<i>base</i>	Port base pointer
in	<i>pin</i>	Port pin number

Returns

MSCR register value

6.3.6.7 Siul2_Port_Ip_GetPinConfiguration()

```
void Siul2_Port_Ip_GetPinConfiguration (
    const Siul2_Port_Ip_PortType *const base,
    Siul2_Port_Ip_PinSettingsConfig * config,
    uint16 pin )
```

This function shall return the value of the pin configuration register.

This function shall return the value of the pin configuration register.

Parameters

in	<i>base</i>	Port base pointer
in	<i>pin</i>	Port pin number
out	<i>config->pointer</i>	to output configuration structure information

Returns

MSCR register value

6.3.7 Variable Documentation

6.3.7.1 Port_au32Siul2BaseAddr

```
const uint32 Port_au32Siul2BaseAddr[] [extern]
```

Base address array for Siul2 instances.

6.4 TSPC IPL

6.4.1 Detailed Description

Function Reference

- void [Tspc_Port_Ip_EnableObeGroup](#) (uint8 group)
Initializes the pins with the given configuration structure.
- void [Tspc_Port_Ip_ConfigureObeGroup](#) (uint32 cfgCount, const Tspc_Port_Ip_ObeGroupConfig config[])
Initializes the pins with the given configuration structure.

6.4.2 Function Reference

6.4.2.1 Tspc_Port_Ip_EnableObeGroup()

```
void Tspc_Port_Ip_EnableObeGroup (
    uint8 group )
```

Initializes the pins with the given configuration structure.

This function shall enable the specified group whose pads are participating in simultaneous transition.

Parameters

in		
----	--	--

6.4.2.2 Tspc_Port_Ip_ConfigureObeGroup()

```
void Tspc_Port_Ip_ConfigureObeGroup (
    uint32 cfgCount,
    const Tspc_Port_Ip_ObeGroupConfig config[] )
```

Initializes the pins with the given configuration structure.

This function shall configure which channels participate in the OBE group. This function enables the specified group whose pads are participating in simultaneous transition.

Parameters

in		
----	--	--

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 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.

© 2023 NXP B.V.

