User Manual

for S32 PORT Driver

Document Number: UM11PORTASR4.4 Rev0000R4.0.0 Rev. 1.0

Chapter 1

Revision History

Revision	Date	Author	Description
1.0	31.10.2022	NXP RTD Team	Prepared for release S32 RTD AUTOSAR 4.4 Version 4.0.0 Release

Chapter 2

Introduction

This User Manual describes NXP Semiconductor AUTOSAR Port for S32. 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:

- s32g274a_bga525
- s32g254a_bga525
- s32g233a_bga525
- s32g234m_bga525
- s32g378a_bga525
- s32g379a_bga525
- s32g398a_bga525
- s32g399a_bga525
- s32g338m_bga525
- s32g339m_bga525
- s32g358a_bga525
- s32g359a_bga525
- s32r45_bga780

All of the above microcontroller devices are collectively named as S32.

2.2 Overview

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

AUTOSAR:

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

2.3 About This Manual

This Technical Reference employs the following typographical conventions:

- **Boldface** style: Used for important terms, notes and warnings.
- *Italic* style: Used for code snippets in the text. Note that C language modifiers such "const" or "volatile" are sometimes omitted to improve readability of the presented code.

Notes and warnings are shown as below:

Note

> This is a note.

Warning

> This is a warning

2.4 Acronyms and Definitions

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

2.5 Reference List

#	Title	Version
1	Specification of Port Driver	AUTOSAR Release 4.4.0
2	Reference Manual	S32G2 Reference Manual, Rev 5, May 2022
		S32G3 Reference Manual, Rev.2 Draft C, June 2022
		S32R45 Reference Manual, Rev. 3, 12/2021
3	Datasheet	S32G2 Data Sheet, Rev 5, May 2022
		S32G3 Data Sheet, Rev 2, Draft B, June 2022
		VR5510 Data Sheet, Rev 5, April 2022
		S32R45 Data Sheet, Rev. 2 — 12/2021
4	Errata	S32G2: Mask Set Errata for Mask 0P77B, Rev. 2.4
		S32G3: Mask Set Errata for Mask 0P72B, Rev. 1.1
		S32R45: Mask Set Errata for Mask P57D, Rev. 2.0

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:
For S32G2

- 1. Open the S32G2_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. PB[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. 19)
- 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 SIUL2 Instance' attribute and choose the instance for the selected pin
- 8. Go to the 'PortPin Mode' attribute and choose the functionality you want to use for the selected pin

For S32G3

- 1. Open the S32G3_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. PB[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. 19)

- 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 SIUL2 Instance' attribute and choose the instance for the selected pin
- 8. Go to the 'PortPin Mode' attribute and choose the functionality you want to use for the selected pin

For S32R45

- 1. Open the S32R45_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. PB[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. 19)
- 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 SIUL2 Instance' attribute and choose the instance for the selected pin
- 8. 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	"Name - PortPinInitialMode - Parent Container - PortPin - Description - Port pin mode from mode list for use with Port_Init() function. - 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 Value - true - Value Configuration Class - Pre-compile time - X - VARIANT-PRE-COMPILE - Link time - -- - - Post-build time - X - VARIANT-POST-BUILD - Scope / Dependency - scope: local - "	Currently implemented in a diffferent mode in MCAL 4.3.0. This requirement was replaced by requirement ECUC_Port_00130.

Requirement	Status	Description	Notes
ECUC_Port_00130	N/S	"Name - PortPinMode - Parent Container - PortPin - Description - Port pin mode from mode list. Note that more than one mode is allowed by default. That way it is e.g. possible to combine DIO with another mode such as ICU. - Multiplicity - 1..∗ - Type - EcucEnumerationParamDef - Range - PORT_PIN_MODE_ADC - Port Pin used by ADC - PORT_↩PIN_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 - "	Replaced by requirement CPR_↩RTD_00372.port
CPR_RTD_00544.port	N/S	Driver shall support Autosar standard configuration format for the IP layer. Note: EPD file for the IP shall be provided.	Current implementation for S32XX release by default is not supporting cross configuration. ARTD-15712 was raised to implement this feature but it was postponed until we have the support from S32CT teams.

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

- Pins Tool should be disabled when Port component is used.
- Siul2_Port should be disabled when Port component is used.
- VSMD reports for PORT have some errors: due to ECUC_PORT_00130 requirement is no longer apply on RTD product (AAI-192).

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 S32 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 IOSignal description Excel file
  - 3. Go to 'IO Signal Table' sheet
  - 4. Identify the microcontroller pin you want to use (eg. PB[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. 19)
  - 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 IOSignal description Excel file
  - 2. Go to 'IO Signal Table' sheet
  - 3. Identify the microcontroller pin you want the Port driver to not touch (eg. PB[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. 19)
  - 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.

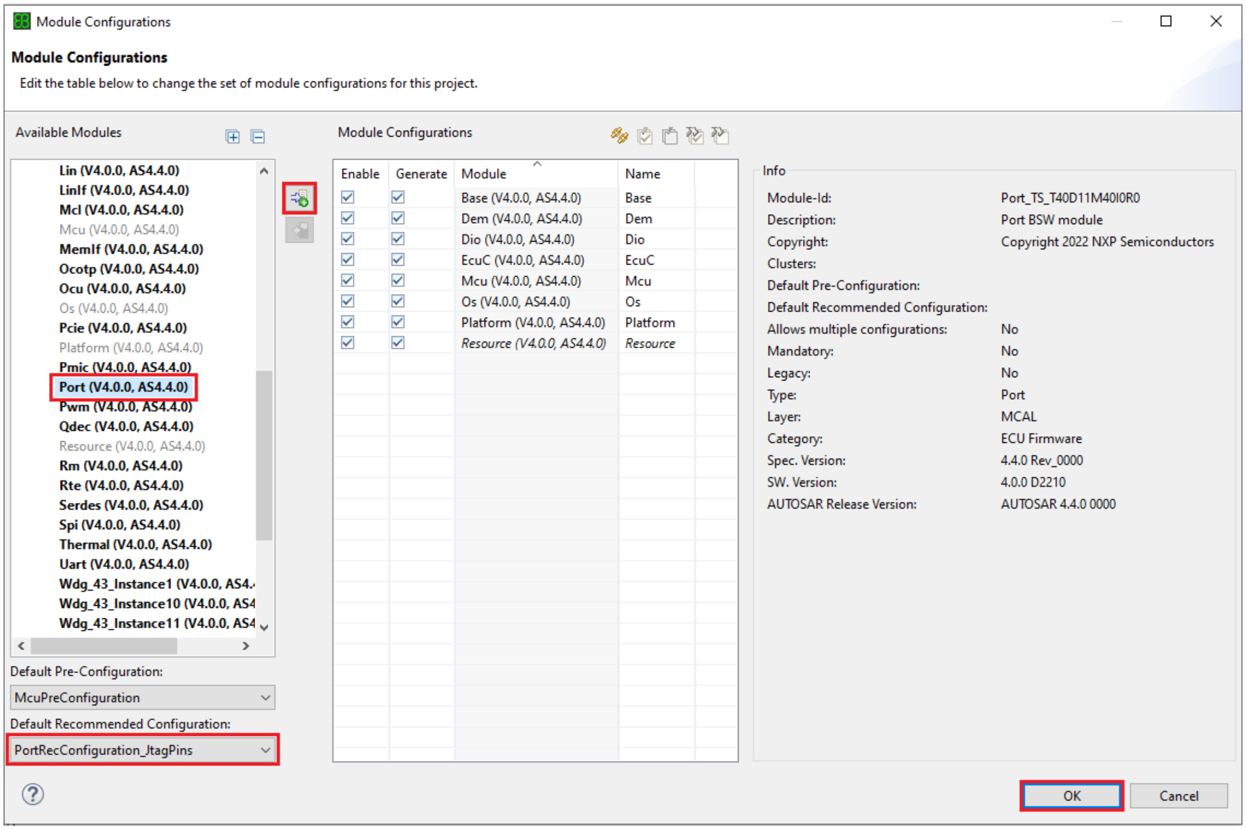
For this reason, if the platform contains hardware pins that need to have certain non GPIO functionalities, these pins must be explicitly added in the Port configuration using mechanism A or B. Otherwise, they will be configured by Port_Init() API as GPIOs.

Important note

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

- JTAG_TDI having PortPin Mscr set to 0 and SIUL2 Instance set to SIUL2_0
- JTAG_TDO having PortPin Mscr set to 1 and SIUL2 Instance set to SIUL2_0
- JTAG_TCK having PortPin Mscr set to 4 and SIUL2 Instance set to SIUL2_0
- JTAG_TMS having PortPin Mscr set to 5 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.



Module Configurations

Module Configurations

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

Available Modules

Lin (V4.0.0, AS4.4.0)
LinIf (V4.0.0, AS4.4.0)
Mcl (V4.0.0, AS4.4.0)
Mcu (V4.0.0, AS4.4.0)
MemIf (V4.0.0, AS4.4.0)
Ocotp (V4.0.0, AS4.4.0)
Ocu (V4.0.0, AS4.4.0)
Os (V4.0.0, AS4.4.0)
Pcie (V4.0.0, AS4.4.0)
Platform (V4.0.0, AS4.4.0)
Pmic (V4.0.0, AS4.4.0)
Port (V4.0.0, AS4.4.0)
Pwm (V4.0.0, AS4.4.0)
Qdec (V4.0.0, AS4.4.0)
Resource (V4.0.0, AS4.4.0)
Rm (V4.0.0, AS4.4.0)
Rte (V4.0.0, AS4.4.0)
Serdes (V4.0.0, AS4.4.0)
Spi (V4.0.0, AS4.4.0)
Thermal (V4.0.0, AS4.4.0)
Uart (V4.0.0, AS4.4.0)
Wdg_43_Instance1 (V4.0.0, AS4.
Wdg_43_Instance10 (V4.0.0, AS4
Wdg_43_Instance11 (V4.0.0, AS4

Default Pre-Configuration: McuPreConfiguration

Default Recommended Configuration: PortRecConfiguration_JtagPins

Module Configurations

Enable	Generate	Module	Name
☑	☑	Base (V4.0.0, AS4.4.0)	Base
☑	☑	Dem (V4.0.0, AS4.4.0)	Dem
☑	☑	Dio (V4.0.0, AS4.4.0)	Dio
☑	☑	EcuC (V4.0.0, AS4.4.0)	EcuC
☑	☑	Mcu (V4.0.0, AS4.4.0)	Mcu
☑	☑	Os (V4.0.0, AS4.4.0)	Os
☑	☑	Platform (V4.0.0, AS4.4.0)	Platform
☑	☑	*Resource (V4.0.0, AS4.4.0)*	*Resource*

Info

Module-Id:	Port_TS_T40D11M40I0R0
Description:	Port BSW module
Copyright:	Copyright 2022 NXP Semiconductors
Clusters:	
Default Pre-Configuration:	
Default Recommended Configuration:	
Allows multiple configurations:	No
Mandatory:	No
Legacy:	No
Type:	Port
Layer:	MCAL
Category:	ECU Firmware
Spec. Version:	4.4.0 Rev_0000
SW. Version:	4.0.0 D2210
AUTOSAR Release Version:	AUTOSAR 4.4.0 0000

Figure 3.1 How to configure JTAG pins

Autosar extension functionality

- Support to run driver's code from User Mode. This option is configurable on/off per entire driver, using the checkbox 'Enable Port User Mode Support' in PortGeneral container. When this parameter is enabled, the Port module will adapt to run from user mode so that the registers under protection can be accessed from user mode. For more information, please see the IM chapter 'User Mode Support'.

- Port SetPinMode Does Not Touch GPIO Levels. This option is configurable on/off and it affects the functionality of the Port_SetPinMode() API. When not checked, the function Port_SetPinMode() will set the output level of the pin to the value configured in the PortPinLevelValue combo when called at run time to change mode of a pin from alternate function to GPIO. When checked, the function Port_SetPinMode() will not touch the output level of the pin when called at run time to change mode of a pin from alternate function to GPIO.

3.7 Runtime errors

This driver doesn't generate any runtime error.

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 PortPinOde
            - Parameter PortPinWithReadBack
            - Parameter PortPinRcvr
            - Parameter PortPinDirectionChangeable
            - Parameter PortPinModeChangeable
            - Parameter PortPinSiul2Instance
            - Parameter PortPinId
            - Parameter PortPinPcr
            - Parameter PortPinDirection
            - Parameter PortPinInitialMode
            - Parameter PortPinMode
            - Parameter PortPinLevelValue
            - Parameter PortPinSlewRate
            - Reference PortPinEcucPartitionRef
        - 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
        - Reference PortEcucPartitionRef
    - Container CommonPublishedInformation
        - Parameter ArReleaseMajorVersion
        - Parameter ArReleaseMinorVersion
        - Parameter ArReleaseRevisionVersion
        - Parameter ModuleId
        - Parameter SwMajorVersion
        - Parameter SwMinorVersion
        - Parameter SwPatchVersion
        - Parameter VendorApiInfix
        - Parameter VendorId

4.1 Module Port

Configuration of the Port module.

Included containers:

- PortConfigSet
- PortGeneral
- CommonPublishedInformation

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

4.2 Container PortConfigSet

This container contains a configuration of the PORT driver / 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
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	191
min	1

4.10 Container PortPin

Configuration of the individual port pins.

Included subcontainers:

- None

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

4.11 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

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	false

4.13 Parameter PortPinSafeMode

Safe Mode Control

Used only when the associated destination is a chip pin. Specifies whether the chip disables the pin's output buffer when the chip enters Safe Mode.

Unchecked box means output is disabled in Safe Mode. The output buffer returns to its previous state when the chip leaves Safe Mode.

Checked box means that output is not disabled in Safe Mode.

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

4.14 Parameter PortPinOde

Enable Open Drain Output for the configured Pin.

Checked box means the Open Drain configuration is set.

This is an implementation specific parameter.

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

4.15 Parameter 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 PortPinRcvr

Receiver Select

Checked box means that the single ended receiver is enabled.

Unchecked box means that the differential vref based receiver is enabled.

Note: In S32R or SIUL2_0 of S32G platform, 1833 and GPIO33 pad types do not support this feature

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 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
defaultValue	true

4.18 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

4.19 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', 'SIUL2_1']

4.20 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

Property	Value
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	191
min	1

4.21 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	190
min	0

4.22 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
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']

4.23 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']

4.24 Parameter PortPinMode

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

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

Property	Value
literals	['GPIO', 'ADCSAR_0_ADCSAR0_INJ_TRIG', 'ADCSAR_0_ADCSAR0_TRIG', 'ADCSAR_1_ADCSAR1_INJ_TRIG', 'ADCSAR_1_ADCSAR1_TRIG', 'BOOT_BOOTMOD_0', 'BOOT_BOOTMOD_1', 'BOOT_RCON0', 'BOOT_RCON10', 'BOOT_RCON11', 'BOOT_RCON12', 'BOOT_RCON13', 'BOOT_RCON14', 'BOOT_RCON15', 'BOOT_RCON16', 'BOOT_RCON17', 'BOOT_RCON18', 'BOOT_RCON19', 'BOOT_RCON1', 'BOOT_RCON20', 'BOOT_RCON21', 'BOOT_RCON22', 'BOOT_RCON23', 'BOOT_RCON24', 'BOOT_RCON25', 'BOOT_RCON26', 'BOOT_RCON27', 'BOOT_RCON28', 'BOOT_RCON29', 'BOOT_RCON2', 'BOOT_RCON30', 'BOOT_RCON31', 'BOOT_RCON3', 'BOOT_RCON4', 'BOOT_RCON5', 'BOOT_RCON6', 'BOOT_RCON7', 'BOOT_RCON8', 'BOOT_RCON9', 'CTU_CTU_EXT_TRIG', 'DSPI_0_DSPI0_PCS0_INOUT', 'DSPI_0_DSPI0_PCS0_IN', 'DSPI_0_DSPI0_PCS0_OUT', 'DSPI_0_DSPI0_PCS1', 'DSPI_0_DSPI0_PCS2', 'DSPI_0_DSPI0_PCS3', 'DSPI_0_DSPI0_PCS4', 'DSPI_0_DSPI0_PCS5', 'DSPI_0_DSPI0_PCS6', 'DSPI_0_DSPI0_PCS7', 'DSPI_0_DSPI0_SCK_INOUT', 'DSPI_0_DSPI0_SCK_IN', 'DSPI_0_DSPI0_SCK_OUT', 'DSPI_0_DSPI0_SIN', 'DSPI_0_DSPI0_SOUT', 'DSPI_1_DSPI1_PCS0_INOUT', 'DSPI_1_DSPI1_PCS0_IN', 'DSPI_1_DSPI1_PCS0_OUT', 'DSPI_1_DSPI1_PCS1', 'DSPI_1_DSPI1_PCS2', 'DSPI_1_DSPI1_PCS3', 'DSPI_1_DSPI1_PCS4', 'DSPI_1_DSPI1_SCK_INOUT', 'DSPI_1_DSPI1_SCK_IN', 'DSPI_1_DSPI1_SCK_OUT', 'DSPI_1_DSPI1_SIN', 'DSPI_1_DSPI1_SOUT', 'DSPI_2_DSPI2_PCS0_INOUT', 'DSPI_2_DSPI2_PCS0_IN', 'DSPI_2_DSPI2_PCS0_OUT', 'DSPI_2_DSPI2_PCS1', 'DSPI_2_DSPI2_PCS2', 'DSPI_2_DSPI2_PCS3', 'DSPI_2_DSPI2_PCS4', 'DSPI_2_DSPI2_SCK', 'DSPI_2_DSPI2_SCK_INOUT', 'DSPI_2_DSPI2_SCK_IN', 'DSPI_2_DSPI2_SCK_OUT', 'DSPI_2_DSPI2_SIN', 'DSPI_2_DSPI2_SOUT', 'DSPI_3_DSPI3_PCS0_INOUT', 'DSPI_3_DSPI3_PCS0_IN', 'DSPI_3_DSPI3_PCS0_OUT', 'DSPI_3_DSPI3_PCS1', 'DSPI_3_DSPI3_PCS2', 'DSPI_3_DSPI3_PCS3', 'DSPI_3_DSPI3_PCS4', 'DSPI_3_DSPI3_SCK', 'DSPI_3_DSPI3_SCK_INOUT', 'DSPI_3_DSPI3_SCK_IN', 'DSPI_3_DSPI3_SCK_OUT', 'DSPI_3_DSPI3_SIN', 'DSPI_3_DSPI3_SOUT', 'DSPI_4_DSPI4_PCS0_INOUT', 'DSPI_4_DSPI4_PCS0_IN', 'DSPI_4_DSPI4_PCS0_OUT', 'DSPI_4_DSPI4_PCS1', 'DSPI_4_DSPI4_PCS2', 'DSPI_4_DSPI4_PCS3', 'DSPI_4_DSPI4_PCS4', 'DSPI_4_DSPI4_SCK', 'DSPI_4_DSPI4_SCK_INOUT', 'DSPI_4_DSPI4_SCK_IN', 'DSPI_4_DSPI4_SCK_OUT', 'DSPI_4_DSPI4_SIN', 'DSPI_4_DSPI4_SOUT', 'DSPI_5_DSPI5_PCS0_INOUT', 'DSPI_5_DSPI5_PCS0_IN', 'DSPI_5_DSPI5_PCS0_OUT', 'DSPI_5_DSPI5_PCS1', 'DSPI_5_DSPI5_PCS2', 'DSPI_5_DSPI5_PCS3', 'DSPI_5_DSPI5_PCS4', 'DSPI_5_DSPI5_SCK', 'DSPI_5_DSPI5_SCK_INOUT', 'DSPI_5_DSPI5_SCK_IN', 'DSPI_5_DSPI5_SCK_OUT', 'DSPI_5_DSPI5_SIN', 'DSPI_5_DSPI5_SOUT', 'FLEXCAN_0_CAN0_RX', 'FLEXCAN_0_CAN0_TX', 'FLEXCAN_1_CAN1_RX', 'FLEXCAN_1_CAN1_TX', 'FLEXCAN_2_CAN2_RX', 'FLEXCAN_2_CAN2_TX', 'FLEXCAN_3_CAN3_RX', 'FLEXCAN_3_CAN3_TX', 'FLEXRAY_0_FR_DBG_0', 'FLEXRAY_0_FR_DBG_1', 'FLEXRAY_0_FR_DBG_2', 'FLEXRAY_0_FR_DBG_3', 'FLEXRAY_0_FR_RXD_A', 'FLEXRAY_0_FR_RXD_B', 'FLEXRAY_0_FR_TXD_A', 'FLEXRAY_0_FR_TXD_B', 'FLEXRAY_0_FR_TXE_A_b', 'FLEXRAY_0_FR_TXE_B_b', 'FLEXTIMER_0_FTM0_CH0_INOUT', 'FLEXTIMER_0_FTM0_CH0_IN', 'FLEXTIMER_0_FTM0_CH0_OUT', 'FLEXTIMER_0_FTM0_CH1_INOUT', 'FLEXTIMER_0_FTM0_CH1_IN', 'FLEXTIMER_0_FTM0_CH1_OUT', 'FLEXTIMER_0_FTM0_CH2_INOUT', 'FLEXTIMER_0_FTM0_CH2_IN', 'FLEXTIMER_0_FTM0_CH2_OUT', 'FLEXTIMER_0_FTM0_CH3_INOUT', 'FLEXTIMER_0_FTM0_CH3_IN', 'FLEXTIMER_0_FTM0_CH3_OUT', 'FLEXTIMER_0_FTM0_CH4_INOUT', 'FLEXTIMER_0_FTM0_CH4_IN', 'FLEXTIMER_0_FTM0_CH4_OUT', 'FLEXTIMER_0_FTM0_CH5', 'FLEXTIMER_0_FTM0_CH5_INOUT', 'FLEXTIMER_0_FTM0_CH5_IN',

Property	Value

4.25 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']

4.26 Parameter PortPinSlewRate

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

For SIUL2_0, SIUL2_1 signals of S32G and SIUL2_0, SIUL2_1 signals of S32R, the Slew Rate can be configured to one of the following values:

For "3.3 V/1.8 V" FAST pads:

Fmax=208 MHz (at 1.8V), 166 MHz (at 3.3V)

Fmax=166 MHz (at 1.8V), 150 MHz (at 3.3V)

Fmax=150 MHz (at 1.8V), 133 MHz (at 3.3V)

Fmax=133 MHz(at 1.8V), 100 MHz (at 3.3V)

Fmax=100 MHz (at 1.8V), 83 MHz (at 3.3V)

For "1.8 V" GPIO pads:

Fmax=208 MHz

Fmax=150 MHz

Fmax=133 MHz

Fmax=100 MHz

Fmax=50 MHz

For "3.3 V" GPIO pads:

Fmax=50 MHz

Fmax=1 MHz

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	N/A
literals	['SRE_3_3V_50MHZ_b100', 'SRE_3_3V_50MHZ_b101', 'SRE_3_3V_50MHZ_b110', 'SRE_3_3V_1MHZ', 'SRE_208MHZ_1_8V_166MHZ_3_3V', 'SRE_166MHZ_1_8V_150MHZ_3_3V', 'SRE_150MHZ_1_8V_133MHZ_3_3V', 'SRE_133MHZ_1_8V_100MHZ_3_3V', 'SRE_100MHZ_1_8V_83MHZ_3_3V', 'SRE_1_8V_208MHZ', 'SRE_1_8V_150MHZ', 'SRE_1_8V_133MHZ', 'SRE_1_8V_100MHZ', 'SRE_1_8V_50MHZ', 'SRE_FASTEST_208MHZ', 'SRE_150MHZ_LOWER', 'SRE_100MHZ_LOWER', 'SRE_50MHZ_LOWER', 'SRE_SLOWEST_25MHZ']

4.27 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

Property	Value
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

4.28 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

4.29 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

Property	Value
defaultValue	SIUL2_0
literals	['SIUL2_0', 'SIUL2_1']

4.30 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	190
min	0

4.31 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

4.32 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
valueConfigClasses	VARIANT-POST-BUILD: POST-BUILD
	VARIANT-PRE-COMPILE: PRE-COMPILE
defaultValue	SIUL2_0
literals	['SIUL2_0', 'SIUL2_1']

4.33 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	496
min	0

4.34 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

4.35 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

4.36 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

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

4.37 Parameter PortSetPinDirectionApi

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

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

4.38 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
defaultValue	true

4.39 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
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	false
valueConfigClasses	VARIANT-POST-BUILD: PRE-COMPILE
	VARIANT-PRE-COMPILE: PRE-COMPILE
defaultValue	true

4.40 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

4.41 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 where set during the 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

4.42 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

4.43 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
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

4.44 Parameter PortMulticoreSupport

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

NoteThis 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

4.45 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
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

4.46 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

4.47 Parameter ArReleaseMajorVersion

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

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

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

4.48 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	4
max	4
min	4

4.49 Parameter ArReleaseRevisionVersion

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

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

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

4.50 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

4.51 Parameter SwMajorVersion

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

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

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

4.52 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

4.53 Parameter SwPatchVersion

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

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

Property	Value
max	0
min	0

4.54 Parameter VendorApiInfix

In driver modules which can be instantiated several times on a single ECU, BSW00347 requires that the name of APIs is extended by the VendorId and a vendor specific name.

This parameter is used to specify the vendor specific name. In total, the implementation specific name is generated as follows:

<ModuleName>_>VendorId>_<VendorApiInfix>.

E.g. assuming that the VendorId of the implementor is 123 and the implementer chose a VendorApiInfix of "v11r456" a api name Can_Write defined in the SWS will translate to Can_123_v11r456Write.

This parameter is mandatory for all modules with upper multiplicity > 1. It shall not be used for modules with upper multiplicity =1.

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

4.55 Parameter VendorId

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

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

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

Chapter 5

Module Index

5.1 Software Specification

Here is a list of all modules:

Chapter 6

Module Documentation

6.1 Port HLD

6.1.1 Detailed Description

Macros

- #define PORT_VENDOR_ID

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

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

  Instance ID of port driver.
- #define PORT_INIT_ID

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

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

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

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

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

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

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

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

6.1.2 Macro Definition Documentation

6.1.2.1 PORT_VENDOR_ID

```
#define PORT_VENDOR_ID
```

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

Definition at line 60 of file Port.h.

6.1.2.2 PORT_E_PARAM_CONFIG

```
#define PORT_E_PARAM_CONFIG
```

The PORT module is not properly configured.

Definition at line 132 of file Port.h.

6.1.2.3 PORT_INSTANCE_ID

```
#define PORT_INSTANCE_ID
```

Instance ID of port driver.

Definition at line 139 of file Port.h.

6.1.2.4 PORT_INIT_ID

```
#define PORT_INIT_ID
```

API service ID for PORT Init function.

Parameters used when raising an error/exception.

Definition at line 153 of file Port.h.

6.1.2.5 PORT_SETPINDIRECTION_ID

```
#define PORT_SETPINDIRECTION_ID
```

API service ID for PORT set pin direction function.

Parameters used when raising an error/exception.

Definition at line 162 of file Port.h.

6.1.2.6 PORT_REFRESHPINDIRECTION_ID

```
#define PORT_REFRESHPINDIRECTION_ID
```

API service ID for PORT refresh pin direction function.

Parameters used when raising an error/exception.

Definition at line 170 of file Port.h.

6.1.2.7 PORT_GETVERSIONINFO_ID

```
#define PORT_GETVERSIONINFO_ID
```

API service ID for PORT get version info function.

Parameters used when raising an error/exception.

Definition at line 179 of file Port.h.

6.1.2.8 PORT_SETPINMODE_ID

```
#define PORT_SETPINMODE_ID
```

API service ID for PORT set pin mode.

Parameters used when raising an error/exception.

Definition at line 188 of file Port.h.

6.1.2.9 PORT_SETASUNUSEDPIN_ID

```
#define PORT_SETASUNUSEDPIN_ID
```

API service ID for PORT set as unused pin.

Parameters used when raising an error/exception.

Definition at line 194 of file Port.h.

6.1.2.10 PORT_SETASUSEDPIN_ID

```
#define PORT_SETASUSEDPIN_ID
```

API service ID for PORT set as used pin.

Parameters used when raising an error/exception.

Definition at line 200 of file Port.h.

6.1.2.11 PORT_RESETPINMODE_ID

```
#define PORT_RESETPINMODE_ID
```

API service ID for PORT reset pin mode.

Parameters used when raising an error/exception.

Definition at line 209 of file Port.h.

6.1.2.12 PORT_E_PARAM_PIN

```
#define PORT_E_PARAM_PIN
```

Error ID of port driver.

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

Invalid Port Pin ID requested.

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

Definition at line 239 of file Port.h.

6.1.2.13 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.

6.1.2.14 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.

6.1.2.15 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.

6.1.2.16 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.

6.1.2.17 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.

6.1.2.18 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.

6.1.3 Function Reference

6.1.3.1 Port_Init()

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

Port driver initialization function.

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

Parameters

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

Returns

void

6.1.3.2 Port_SetPinDirection()

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

Port_SetPinDirection.

Function used for changing the pin direction at runtime

Parameters

in		

pin id of the pin that needs to change the direction

Parameters

in		

new desired direction IN OUT IN_OUT

Returns

void

6.1.3.3 Port_SetPinMode()

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

Port_SetPinMode.

Function used to change the pin mode at runtime.

Parameters

in		

pin id of the pin that needs to change the direction

Parameters

in		

new mode

Returns

void

6.1.3.4 Port_GetVersionInfo()

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

Port_GetVersionInfo.

Function used to read the driver version information

Parameters

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

Returns

void

6.1.3.5 Port_RefreshPortDirection()

```
void Port_RefreshPortDirection (
            void  )
```

Port_RefreshPortDirection.

function used to reset the direction of the pin

Returns

void

6.1.3.6 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

6.1.3.7 Port_SetAsUsedPin()

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

Port_SetAsUsedPin.

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

Returns

void

6.1.3.8 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

6.2 Port IPL

6.2.1 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.

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_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_PortReceiverSelect

  Configures the Receiver Select. Implements : Siul2_Port_Ip_PortReceiverSelect.
- enum Siul2_Port_Ip_PortOpenDrain

  Configures the Open Drain Enable field. Implements : Siul2_Port_Ip_PortOpenDrain.
- enum Siul2_Port_Ip_PortDirectionType

  Configures port direction.

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_PortDirectionTyp 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.

6.2.2 Data Structure Documentation

6.2.2.1 struct Siul2_Port_Ip_PortType

PORT - Register Layout Typedef

Definition at line 387 of file Siul2_Port_Ip_Types.h.

6.2.2.2 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 408 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_PortReceiverSelect	receiverSel	Configures the Receiver Select, apply for SIUL2_0/1
Siul2_Port_Ip_PortOpenDrain	openDrain	Configures open drain, apply for SIUL2_0/1
Siul2_Port_Ip_PortOutputBuffer	outputBuffer	Configures the Output Buffer Enable.
Siul2_Port_Ip_PortInputBuffer	inputBuffer	Configures the Input Buffer Enable.
Siul2_Port_Ip_PortInputMux	inputMux[(8U)]	Configures the input muxing
uint32	inputMuxReg[(8U)]	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

6.2.3 Macro Definition Documentation

6.2.3.1 PORT_PIN_LEVEL_NOTCHANGED_U8

```
#define PORT_PIN_LEVEL_NOTCHANGED_U8
```

Not changed port pin logic.

Definition at line 166 of file Siul2_Port_Ip.h.

6.2.3.2 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.

6.2.4 Types Reference

6.2.4.1 Siul2_Port_Ip_PortPinsLevelType

```
typedef uint8 Siul2_Port_Ip_PortPinsLevelType
```

Type of a port levels representation. Implements : Siul2_Port_Ip_PortPinsLevelType.

Definition at line 115 of file Siul2_Port_Ip_Types.h.

6.2.5 Enum Reference

6.2.5.1 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 121 of file Siul2_Port_Ip_Types.h.

6.2.5.2 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

Definition at line 132 of file Siul2_Port_Ip_Types.h.

6.2.5.3 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

Definition at line 185 of file Siul2_Port_Ip_Types.h.

6.2.5.4 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

Definition at line 195 of file Siul2_Port_Ip_Types.h.

6.2.5.5 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

Enumerator

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 229 of file Siul2_Port_Ip_Types.h.

6.2.5.6 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

Definition at line 256 of file Siul2_Port_Ip_Types.h.

6.2.5.7 Siul2_Port_Ip_PortSlewRateControl

```
enum Siul2_Port_Ip_PortSlewRateControl
```

Configures the slew rate control. Implements : Siul2_Port_Ip_PortSlewRateControl.

Enumerator

PORT_SLEW_RATE_CONTROL0	Fmax=208 MHz (at 1.8V), 166 MHz (at 3.3V), apply for SIUL2_0/1
PORT_SLEW_RATE_CONTROL4	Fmax=166 MHz (at 1.8V), 150 MHz (at 3.3V, apply for SIUL2_0/1
PORT_SLEW_RATE_CONTROL5	Fmax=150 MHz (at 1.8V), 133 MHz (at 3.3V), apply for SIUL2_0/1
PORT_SLEW_RATE_CONTROL6	Fmax=133 MHz(at 1.8V), 100 MHz (at 3.3V), apply for SIUL2_0/1
PORT_SLEW_RATE_CONTROL7	Fmax=83 MHz (at 1.8V), 63 MHz (at 3.3V), apply for SIUL2_0/1

Definition at line 286 of file Siul2_Port_Ip_Types.h.

6.2.5.8 Siul2_Port_Ip_PortReceiverSelect

```
enum Siul2_Port_Ip_PortReceiverSelect
```

Configures the Receiver Select. Implements : Siul2_Port_Ip_PortReceiverSelect.

Enumerator

PORT_RECEIVER_ENABLE_DIFFERENTIAL_VREF	Enables the differential vref based receiver.
PORT_RECEIVER_ENABLE_SINGLE_ENDED	Enables the single ended receiver.

Definition at line 313 of file Siul2_Port_Ip_Types.h.

6.2.5.9 Siul2_Port_Ip_PortOpenDrain

```
enum Siul2_Port_Ip_PortOpenDrain
```

Configures the Open Drain Enable field. Implements : Siul2_Port_Ip_PortOpenDrain.

Enumerator

PORT_OPEN_DRAIN_DISABLED	Output is CMOS
PORT_OPEN_DRAIN_ENABLED	Output is open drain

Definition at line 373 of file Siul2_Port_Ip_Types.h.

6.2.5.10 Siul2_Port_Ip_PortDirectionType

```
enum Siul2_Port_Ip_PortDirectionType
```

Configures port direction.

Enumerator

SIUL2_PORT_IN	Sets port pin as input.
SIUL2_PORT_OUT	Sets port pin as output.
SIUL2_PORT_IN_OUT	Sets port pin as bidirectional.
SIUL2_PORT_HI_Z	Sets port pin as high_z.

Definition at line 394 of file Siul2_Port_Ip_Types.h.

6.2.6 Function Reference

6.2.6.1 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	*base*	Port base pointer (PORTA, PORTB, PORTA_AE, etc.)
in	*pin*	Port pin number
in	*pullConfig*	The pull configuration

6.2.6.2 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	*base*	Port base pointer (PORTA, PORTB, PORTA_AE, etc.)
in	*pin*	Port pin number
in	*enable*	Enable output buffer
in	*mux*	Pin muxing slot selection

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

Note

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

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

Returns

The status of the operation

6.2.6.5 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	*base*	Port base pointer
in	*pin*	Port pin number
in	*direction*	The direction of pin

Returns

void

6.2.6.6 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	*base*	Port base pointer
in	*pin*	Port pin number

Returns

MSCR register value

6.2.6.7 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	*base*	Port base pointer
in	*pin*	Port pin number
out	*config->pointer*	to output configuration structure information

Returns

MSCR register value

6.2.7 Variable Documentation

6.2.7.1 Port_au32Siul2BaseAddr

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

Base address array for Siul2 instances.

How to Reach Us:

Home Page:
nxp.com

Web Support:
nxp.com/support