MCAL Configuration Verification Manual for PORT

32-bit TriCore™ AURIX™ TC3xx microcontroller family

About this document

Scope and purpose

This Configuration Data Reference document is applicable to all TC3xx devices in the TriCore™ AURIX™ family of 32-bit microcontrollers.

The purpose of this document is to facilitate the integrator to verify the generated code based on the input configuration parameters. This document describes details of structures, defines, macros and variables generated from the configuration parameters.

Intended audience

This document is intended for integrators who need to understand the logic of the generated configuration code of AURIX™ AUTOSAR MCAL.

Reference documents

This document should be read in conjunction with the following documents:

- AURIXTM TC3xx MCAL User Manual Port

Table of contents

1 PORT driver

This chapter describes the details of the configuration data generated from the PORT driver.

1.1 File: Port_Cfg.h

The generated header file contains all pre-compile configuration parameters. Pre-compile time configuration allows decoupling of the static configuration from implementation. The file is generated in 'inc' folder.

1.1.1 Macro: PORT_AR_RELEASE_MAJOR_VERSION

Table 1 PORT_AR_RELEASE_MAJOR_VERSION

Name	PORT_AR_RELEASE_MAJOR_VERSION	
Description	Major version number of AUTOSAR release on which the PORT implementation is based on.	
Verification method	The macro is generated with the value present in 'CommonPublishedInformation/ArMajorVersion'. *Note: The macro is not user configurable.*	
Example(s)	**Action**	**Generated output**
	Generate Port_Cfg.h file	`#define PORT_AR_RELEASE_MAJOR_VERSION (4U)`

1.1.2 Macro: PORT_AR_RELEASE_MINOR_VERSION

Table 2 PORT_AR_RELEASE_MINOR_VERSION

Name	PORT_AR_RELEASE_MINOR_VERSION	
Description	Minor version number of AUTOSAR release on which the PORT implementation is based on.	
Verification method	The macro is generated with the value present in 'CommonPublishedInformation/ArMinorVersion'. *Note: The macro is not user configurable.*	
Example(s)	**Action**	**Generated output**
	Generate Port_Cfg.h file	`#define PORT_AR_RELEASE_MINOR_VERSION (2U)`

1.1.3 Macro: PORT_AR_RELEASE_REVISION_VERSION

Table 3 PORT_AR_RELEAE_REVISION_VERSION

Name	PORT_AR_RELEASE_REVISION_VERSION

Description	Revision version number of AUTOSAR release on which the PORT implementation is based on.	
Verification method	The macro is generated with the value present in 'CommonPublishedInformation/ArPatchVersion'. *Note: The macro is not user configurable.*	
Example(s)	**Action**	**Generated output**
	Generate Port_Cfg.h file	`#define PORT_AR_RELEASE_REVISION_VERSION (2U)`

1.1.4 Macro: PORT_SW_MAJOR_VERSION

Table 4 PORT_SW_MAJOR_VERSION

Name	PORT_SW_MAJOR_VERSION	
Description	Major version number of the PORT module.	
Verification method	The macro is generated with the value present in 'CommonPublishedInformation/SwMajorVersion *Note: The macro is not user configurable.*	
Example(s)	**Action**	**Generated output**
	Generate Port_Cfg.h file	`#define PORT_SW_MAJOR_VERSION (10U)`

1.1.5 Macro: PORT_SW_MINOR_VERSION

Table 5 PORT_SW_MINOR_VERSION

Name	PORT_SW_MINOR_VERSION	
Description	Minor version number of the PORT module.	
Verification method	The macro is generated with the value present in 'CommonPublishedInformation/SwMinorVersion *Note: The macro is not user configurable.*	
Example(s)	**Action**	**Generated output**
	Generate Port_Cfg.h file	`#define PORT_SW_MINOR_VERSION (10U)`

1.1.6 Macro: PORT_SW_PATCH_VERSION

Table 6 PORT_SW_PATCH_VERSION

Name	PORT_SW_PATCH_VERSION	
Description	Patch level version number of the PORT module.	
Verification method	The macro is generated with the value present in ‘CommonPublishedInformation/SwPatchVersion *Note: The macro is not user configurable.*	
Example(s)	**Action**	**Generated output**
	Generate Port_Cfg.h file	`#define PORT_SW_PATCH_VERSION (0U)`

1.1.7 Macro: PORT_DEV_ERROR_DETECT

Table 7 PORT_DEV_ERROR_DETECT

Name	PORT_DEV_ERROR_DETECT	
Description	Enables/disables development error detection	
Verification method	The macro is generated as STD_ON if PortDevErrorDetect configuration parameter is set to ‘True’ else the macro is generated as STD_OFF.	
Example(s)	**Action**	**Generated output**
	PortDevErrorDetect = True	`#define PORT_DEV_ERROR_DETECT (STD_ON)`
	PortDevErrorDetect = False	`#define PORT_DEV_ERROR_DETECT (STD_OFF)`

1.1.8 Macro: PORT_VERSION_INFO_API

Table 8 PORT_VERSION_INFO_API

Name	PORT_VERSION_INFO_API	
Description	Enables/disables Port_GetVersionInfo API	
Verification method	The macro is generated as STD_ON if PortVersionInfoApi configuration parameter is set to ‘True’ else the macro is generated as STD_OFF.	
Example(s)	**Action**	**Generated output**
	PortVersionInfoApi = True	`#define PORT_VERSION_INFO_API (STD_ON)`
	PortVersionInfoApi = False	`#define PORT_VERSION_INFO_API (STD_OFF)`

1.1.9 Macro: PORT_SET_PIN_MODE_API

Table 9 PORT_SET_PIN_MODE_API

Name	PORT_SET_PIN_MODE_API	
Description	Enables/disables Port_SetPinMode API	
Verification method	The macro is generated as STD_ON if PortSetPinModeApi configuration parameter is set to 'True' else the macro is generated as STD_OFF.	
Example(s)	**Action**	**Generated output**
	PortSetPinModeApi = True	`#define PORT_SET_PIN_MODE_API (STD_ON)`
	PortSetPinModeApi = False	`#define PORT_SET_PIN_MODE_API (STD_OFF)`

1.1.10 Macro: PORT_SET_PIN_DIRECTION_API

Table 10 PORT_SET_PIN_DIRECTION_API

Name	PORT_SET_PIN_DIRECTION_API	
Description	Enables/disables Port_SetPinDirection API	
Verification method	The macro is generated as STD_ON if PortSetPinDirectionApi configuration parameter is set to 'True' else the macro is generated as STD_OFF	
Example(s)	**Action**	**Generated output**
	PortSetPinDirectionApi = True	`#define PORT_SET_PIN_DIRECTION_API (STD_ON)`
	PortSetPinDirectionApi = False	`#define PORT_SET_PIN_DIRECTION_API (STD_OFF)`

1.1.11 Macro: PORT_INIT_API_MODE

Table 11 PORT_INIT_API_MODE

Name	PORT_INIT_API_MODE	
Description	Decides the mode of execution of Init API.	
Verification method	The macro is generated as PORT_MCAL_USER1 if PortInitApiMode configuration parameter is set to 'PORT_MCAL_USER1' else the macro is generated as PORT_MCAL_SUPERVISOR.	
Example(s)	**Action**	**Generated output**
	PortInitApiMode = PORT_MCAL_USER1	`#define PORT_INIT_API_MODE (PORT_MCAL_USER1)`
	PortInitApiMode = PORT_MCAL_SUPERVISOR	`#define PORT_INIT_API_MODE (PORT_MCAL_SUPERVISOR)`

1.1.12 Macro: PORT_SAFETY_ENABLE

Table 12 PORT_SAFETY_ENABLE

Name	PORT_SAFETY_ENABLE	
Description	Enables/disables safety features	
Verification method	The macro is generated as STD_ON if PortSafetyEnable configuration parameter is set to 'True' else the macro is generated as STD_OFF.	
Example(s)	**Action**	**Generated output**
	PortSafetyEnable= True	`#define PORT_SAFETY_ENABLE (STD_ON)`
	PortSafetyEnable= False	`#define PORT_SAFETY_ENABLE (STD_OFF)`

1.1.13 Macro: PORT_INIT_CHECK_API

Table 13 PORT_INIT_CHECK_API

Name	PORT_INIT_CHECK_API	
Description	Enables/disables the Port_InitCheck API.	
Verification method	The macro is generated as STD_ON if PortInitCheckApi configuration parameter is set to 'True' else the macro is generated as STD_OFF.	
Example(s)	**Action**	**Generated output**
	PortInitCheckApi = True	`#define PORT_INIT_CHECK_API (STD_ON)`
	PortInitCheckApi = False	`#define PORT_INIT_CHECK_API (STD_OFF)`

1.1.14 Macro: PORTS_AVAILABLE_00_31

Table 14 PORTS_AVAILABLE_00_31

Name	PORTS_AVAILABLE_00_31	
Description	Specifies the ports between 0 – 31 which are available on the microcontroller. *Note: The macro is not configurable by the user.*	
Verification method	The macro is generated as a bit-wise numeric representation of ports (Port0 to Port31) available in hardware. Bit0 correspond to Port0 up to Bit31 which corresponds to Port31. 1 represents a port is available and 0 represents as not available.	
Example(s)	**Action**	**Generated output**
	Port Available = 3 4 5 6 7 8 9 16 17 18 19 30 31	`#define PORTS_AVAILABLE_00_31 (0xc00f03f8U)`

	Port Available = 0 1 2 10 11 12 13 14 15 20 21 22 23 24 25 26 30 31	`#define PORTS_AVAILABLE_00_31 (0xc7f0fc07U)`

1.1.15 Macro: PORTS_AVAILABLE_32_63

Table 15 PORTS_AVAILABLE_32_63

Name	PORTS_AVAILABLE_32_63	
Description	Specifies the ports available between 32 – 63 on the microcontroller. *Note: The macro is not configurable by the user.*	
Verification method	The macro is generated as a bit-wise numeric representation of ports (Port32 to Port63) available in hardware. Bit0 correspond to Port32 up to Bit31 which corresponds to Port63. 1 represents a port is available and 0 represents as not available.	
Example(s)	**Action**	**Generated output**
	Port Available: 35 36 37 38 39 40 41	`#define PORTS_AVAILABLE_32_63 (0x000003f8U)`
	Port Available: 32 33 34 40 41	`#define PORTS_AVAILABLE_32_63 (0x00000307U)`

1.1.16 Macro: PORTS_READONLY_00_31

Table 16 PORTS_READONLY_00_31

Name	PORTS_READONLY_00_31	
Description	Specifies the ports between 0 -31 which are read only on the microcontroller. *Note: The macro is not configurable by the user.*	
Verification method	The macro is generated as a bit-wise numeric representation of ports (Port0 to Port31) which are read only in hardware. Bit0 correspond to Port0 up to Bit31 which corresponds to Port31. 1 represents a port is read only else 0.	
Example(s)	**Action**	**Generated output**
	Read only ports: 40 41	`#define PORTS_READONLY_00_31 (0x00000000U)`
	Read only ports: 0 1 2 3 4 5 6 7 8 9 10	`#define PORTS_READONLY_00_31 (0x000007ffU)`

1.1.17 Macro: PORTS_READONLY_32_63

Table 17 PORTS_READONLY_00_31

Name	PORTS_READONLY_32_63	
Description	Specifies the ports between 0 – 32 which are read only on the microcontroller. *Note: The macro is not configurable by the user.*	
Verification method	The macro is generated as a bit-wise numeric representation of ports (Port32 to Port63) read only in hardware. Bit0 correspond to Port32 up to Bit31 which corresponds to Port63. 1 represents a port is read only else 0.	
Example(s)	**Action**	**Generated output**
	Read only ports: 1 2 3 4 5 6 7 8 9 10	`#define PORTS_READONLY_32_63 (0x00000000U)`
	Read only ports: 40 41	`#define PORTS_READONLY_32_63 (0x00000300U)`

1.1.18 Macro: PORT_MAX_NUMBER

Table 18 PORT_MAX_NUMBER

Name	PORT_MAX_NUMBER	
Description	Number of the maximum port available in the microcontroller *Note: The macro is not configurable by the user.*	
Verification method	The macro is generated as a numeric value based on the maximum port which is available in the hardware	
Example(s)	**Action**	**Generated output**
	Maximum port available = 20	`#define PORT_MAX_NUMBER (20U)`
	Maximum port available = 41	`#define PORT_MAX_NUMBER (41U)`

1.1.19 Macro: PORT_MAX_PIN_ID

Table 19 PORT_MAX_PIN_ID

Name	PORT_MAX_PIN_ID	
Description	Number of the maximum port pin available in the microcontroller	
Verification method	The macro is generated as the (maximum pin available in hardware + (maximum port available in hardware* 16))	
Example(s)	**Action**	**Generated output**

	Maximum port available= 10 Maximum pin available = 8	`#define PORT_MAX_PIN_ID (0xA8U)`
	Maximum port available= 41 Maximum pin available = 15	`#define PORT_MAX_PIN_ID (0x29fU)`

1.1.20 Macro: PORT_TOTAL_AVAILABLE_PORTS

Table 20 PORT_TOTAL_AVAILABLE_PORTS

Name	PORT_TOTAL_AVAILABLE_PORTS	
Description	Maximum number of pin available across all ports.	
Verification method	The macro is generated based on the maximum number of pin available across all ports. *Note: The macro is not user configurable.*	
Example(s)	**Action**	**Generated output**
	Maximum pin available = 16	`#define PORT_TOTAL_AVAILABLE_PORTS (16U)`

1.1.21 Macro: PORT_MAX_LPCR_REG

Table 21 PORT_MAX_LPCR_REG

Name	PORT_MAX_LPCR_REG	
Description	Maximum number of LPCR registers	
Verification method	The macro is generated based on maximum number of LPCR registers available in hardware. *Note: The macro is not user configurable.*	
Example(s)	**Action**	**Generated output**
	Generate Port_Cfg.h file	`#define PORT_MAX_LPCR_REG (7U)`

1.1.22 Macro: PORT14_PDISC_MASK

Table 22 PORT14_PDISC_MASK

Name	PORT14_PDISC_MASK
Description	Macro that defines the derivate for errata PORTS_TC.H012. It is applicable only if Port14 pin9 and pin10 is configured as LVDS enabled and device is TC397, TC397_ADAS or TC387.
Verification method	The macro is generated as 0x800U if the errata PORTS_TC.H012 is applicable for the microcontroller derivative, that is if Port14 pin9 and pin10 is configured as LVDS

	enabled and device is TC397, TC397_ADAS or TC387, else macro is generated as 0x0 *Note: The macro is not user configurable.*	
Example(s)	**Action**	**Generated output**
	Errata PORTS_TC.H012 is not applicable	`#define PORT14_PDISC_MASK (0U)`
	Errata PORTS_TC.H012 is applicable	`#define PORT14_PDISC_MASK (0x800U)`

1.1.23 Macro: PORTS_LVDS_00_31

Table 23 PORTS_LVDS_00_31

Name	PORTS_LVDS_00_31	
Description	Specify the LVDS ports between 0 – 31 available on the microcontroller	
Verification method	The macro is generated as a bit-wise numeric representation of LVDS ports (Port0 to Port31) in hardware. Bit0 correspond to Port0 up to Bit31 which corresponds to Port31. 1 represents a port that support LVDS and 0 represent the port which doesn't support LVDS. *Note: The macro is not user configurable.*	
Example(s)	**Action**	**Generated output**
	LVDS ports: 0 1 2 3 4 5 6 7 8 9 10 11 30 31	`#define PORTS_LVDS_00_31 (0xc0000fffU)`
	LVDS ports: 13 14 15 21 22	`#define PORTS_LVDS_00_31 (0x0060e000U)`

1.1.24 Macro: PORTS_LVDS_32_63

Table 24 PORTS_LVDS_32_63

Name	PORTS_LVDS_32_63	
Description	Specify the LVDS ports between 32 – 63 available on the microcontroller	
Verification method	The macro is generated as a bit-wise numeric representation of LVDS ports (Port32 to Port63) in hardware. Bit0 correspond to Port32 up to Bit31 which corresponds to Port63. 1 represents a port that support LVDS and 0 represent the port which doesn't support LVDS. *Note: The macro is not user configurable.*	
Example(s)	**Action**	**Generated output**

	LVDS ports: 32 33 34 35 36 37 38 39 40 41 42 43 62 63	`#define PORTS_LVDS_32_63 (0xc0000fffU)`
	LVDS ports:44 45 46 47 52 53	`#define PORTS_LVDS_32_63 (0x0030f000U)`

1.1.25 Macro: PORTS_PCSR_00_31

Table 25 PORTS_PCSR_00_31

Name	PORTS_PCSR_00_31	
Description	Specify the PCSR ports between 00 – 31 available on the microcontroller	
Verification method	The macro is generated as a bit-wise numeric representation of PCSR ports (Port0 to Port31) in hardware. Bit0 correspond to Port0 up to Bit31 which corresponds to Port31. 1 represents a port that support PCSR and 0 represent the port which doesn't support PCSR. *Note: The macro is not user configurable.*	
Example(s)	**Action**	**Generated output**
	PCSR supported ports: 0 2 4 6 8 10 12 13 14 15 30 31	`#define PORTS_PCSR_00_31 (0xc000f555)`
	PCSR supported ports: 1 3 5 7 9 11 16 17 18 19 20 30 31	`#define PORTS_PCSR_00_31 (0xc00f0aaa)`

1.1.26 Macro: PORTS_PCSR_32_63

Table 26 PORTS_PCSR_32_63

Name	PORTS_PCSR_32_63	
Description	Specify the PCSR ports between 32 – 63 available on the microcontroller	
Verification method	The macro is generated as a bit-wise numeric representation of PCSR ports (Port32 to Port63) in hardware. Bit0 correspond to Port32 up to Bit31 which corresponds to Port63. 1 represents a port that support PCSR and 0 represent the port which doesn't support PCSR. *Note: The macro is not user configurable.*	
Example(s)	**Action**	**Generated output**
	PCSR supported ports: 32 34 36 38 40 42 44 46 48 50 52 54 55 56 57 62 63	`#define PORTS_PCSR_32_63 (0xc0d55555)`
	PCSR supported ports: 33 35 37 39 41 43 45 47 49 51 53 62 63	`#define PORTS_PCSR_32_63 (0xc02aaaaa)`

1.1.27 Macro: PORT_AVAILABLE_PINS_PORT<PortNumber>

Table 27 PORT_AVAILABLE_PINS_PORT<PortNumber>

Name	PORT_AVAILABLE_PINS_PORT<PortNumber>	
Description	Specifies the pins available in a port.	
Verification method	The macro is generated as bit-wise numeric representation of pins (Pin0 to Pin15) available in each ports of the microcontroller. Bit0 correspond to Pin0 up to Bit15 which corresponds to Pin15. Each macro is generated for each port up to the maximum port available in hardware. The macro will not be generated for the port which is not present in hardware. *Note: The macro is not user configurable.*	
Example(s)	**Action**	**Generated output**
	Maximum port available: 6 Pins available for Port 0: 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 Port 2: 1 2 3 4 6 7 8 9 11 12 13 Port 4: 8 9 10 11 12 13 14 15 Port 6: 0 1 2 3 4 5 6 7 8	`#define PORT_AVAILABLE_PINS_PORT0 (0xffff)` `#define PORT_AVAILABLE_PINS_PORT2 (0xcbde)` `#define PORT_AVAILABLE_PINS_PORT4 (0xff00)` `#define PORT_AVAILABLE_PINS_PORT6 (0x00ff)`
	Maximum port available: 7 Pins available for Port 1: 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 Port 3: 1 2 3 4 6 7 8 9 11 12 13 Port 5: 8 9 10 11 12 13 14 15 Port 7: 0 1 2 3 4 5 6 7 8	`#define PORT_AVAILABLE_PINS_PORT1 (0xffff)` `#define PORT_AVAILABLE_PINS_PORT3 (0xcbde)` `#define PORT_AVAILABLE_PINS_PORT5 (0xff00)` `#define PORT_AVAILABLE_PINS_PORT7 (0x00ff)`

1.1.28 Macro: PORTS_PDISC_00_31

Table 28 PORTS_PDISC_00_31

Name	PORTS_PDISC_00_31
Description	Specify the analog/digital input ports between 00 – 31 available on the microcontroller.
Verification method	The macro is generated as a bit-wise numeric representation of analog/digital ports (Port0 to Port31) in hardware. Bit0 correspond to Port0 up to Bit31 which corresponds to Port31. 1 represents a port that support analog/digital inputs and 0 represent the port

	which doesn't support both analog/digital inputs. *Note: The macro is not user configurable.*	
Example(s)	**Action**	**Generated output**
	Analog/digital ports : 0 1 2 10 11 30 31	`#define PORTS_PDISC_00_31 (0xc0000c07U)`
	Analog/digital ports : 12 13 15 19 20 30 31	`#define PORTS_PDISC_00_31 (0xc018b000U)`

1.1.29 Macro: PORTS_PDISC_32_63

Table 29 PORTS_PDISC_32_63

Name	PORTS_PDISC_32_63	
Description	Specify the analog/digital input ports available between 32 – 63 on the microcontroller.	
Verification method	The macro is generated as a bit-wise numeric representation of analog/digital ports (Port32 to Port63) in hardware. Bit0 correspond to Port32 up to Bit31 which corresponds to Port63. 1 represents a port that support analog/digital inputs and 0 represent the port which doesn't support both analog/digital inputs. *Note: The macro is not user configurable.*	
Example(s)	**Action**	**Generated output**
	Analog/digital ports : 32 33 34 42 43 62 63	`#define PORTS_PDISC_32_63 (0xc0000c07U)`
	Analog/digital ports : 44 45 46 51 52 62 63	`#define PORTS_PDISC_32_63 (0xc018b000U)`

1.1.30 Macro: PortConf_PortContainer_<ContainerNumber>_PORT_<PortNumber>_PIN_<PinNumber>

Table 30 PortConf_PortContainer_<ContainerNumber>_PORT_<PortNumber>_PIN_<PinNumber>

Name	PortConf_PortContainer_<ContainerNumber>_PORT_<PortNumber>_PIN_<PinNumber>
Description	Specify the symbolic name for available port pin in the micro controller. <ContainerNumber> represents the suffixed value of container name in PortConfigSet\PortContainer. <PortNumber> represent the actual port number. <PinNumber> represents the pin available in the port.
Verification method	The macro is generated as the (<PinNumber> + (<PortNumber> * 16))

Example(s)	Action	Example(s)
	Available ports: 0 2 4 Pins available in port 0: 0 15 Pins available in port 2: 2 14 Pins available in port 4: 7 11	`#define PortConf_PortContainer_0_PORT_0_PIN_0 (Port_PinType )(0x0000U)` `#define PortConf_PortContainer_0_PORT_0_PIN_15 (Port_PinType )(0x000fU)` `#define PortConf_PortContainer_1_PORT_2_PIN_2 (Port_PinType )(0x0022U)` `#define PortConf_PortContainer_1_PORT_2_PIN_14 (Port_PinType )(0x002eU)` `#define PortConf_PortContainer_2_PORT_4_PIN_7 (Port_PinType )(0x0047U)` `#define PortConf_PortContainer_2_PORT_4_PIN_11 (Port_PinType )(0x004bU)`
	Available ports: 1 3 5 Pins available in port 1: 2 12 Pins available in port 3: 5 9 Pins available in port 5: 8 11	`#define PortConf_PortContainer_0_PORT_1_PIN_2 (Port_PinType )(0x0012U)` `#define PortConf_PortContainer_0_PORT_1_PIN_12 (Port_PinType )(0x001cU)` `#define PortConf_PortContainer_1_PORT_3_PIN_5 (Port_PinType )(0x0035U)` `#define PortConf_PortContainer_1_PORT_3_PIN_9 (Port_PinType )(0x0039U)` `#define PortConf_PortContainer_2_PORT_5_PIN_8 (Port_PinType )(0x0058U)` `#define PortConf_PortContainer_2_PORT_5_PIN_11 (Port_PinType )(0x004bU)`

1.2 File: Port[_<variant>]_PBcfg.c

The generated source file contains all post-build configuration parameters. Post-build time configuration mechanism allows configurable functionality of PORT driver that is deployed as object code. The file is generated in ‘src’ folder.

1.2.1 Macro: Portx_lPdrConfig1(Pd0,Pd1,Pd2,Pd3,Pd4,Pd5,Pd6,Pd7)

Table 31 Portx_lPdrConfig1(Pd0,Pd1,Pd2,Pd3,Pd4,Pd5,Pd6,Pd7)

Name	Portx_lPdrConfig1(Pd0,Pd1,Pd2,Pd3,Pd4,Pd5,Pd6,Pd7)	
Description	Function line macro definition for PORT pad drive control register Pn_PDR0.	
Verification method	The macro is generated to set corresponding bits of Pn_PDR0. Each Pd<x> is 4 bits and corresponds to respective bit positions (x * 4) where x = 0 to 7. *Note: The macro is not user configurable.*	
Example(s)	**Action**	**Generated output**
	Generate Port_PBcfg.c file	`#define Portx_lPdrConfig1(Pd0,Pd1,Pd2,Pd3,Pd4,Pd5,Pd6,Pd7) \` `( ((uint32)(Pd7) << (uint32)28) | ((uint32)(Pd6) << (uint32)24) \ \` `((uint32)(Pd5) << (uint32)20) | ((uint32)(Pd4) << (uint32)16) \ \` `((uint32)(Pd3) << (uint32)12) | ((uint32)(Pd2) << (uint32)8) \ \` `((uint32)(Pd1) << (uint32)4) | (uint32)(Pd0) \` `)`

1.2.2 Macro: Portx_lPdrConfig2 (Pd8,Pd9,Pd10,Pd11,Pd12,Pd13,Pd14,Pd15)

Table 32 Portx_lPdrConfig2(Pd8,Pd9,Pd10,Pd11,Pd12,Pd13,Pd14,Pd15)

Name	Portx_lPdrConfig2(Pd8,Pd9,Pd10,Pd11,Pd12,Pd13,Pd14,Pd15)	
Description	Function like macro definition for PORT pad drive control register Pn_PDR1.	
Verification method	The macro is generated to set corresponding bits of Pn_PDR1. Each Pd<x> is 4 bits and corresponds to respective bit positions ((x-8) * 4) where x = 8 to 15. *Note: The macro is not user configurable.*	
Example(s)	**Action**	**Generated output**
	Generate Port_PBcfg.c file	`#define Portx_lPdrConfig2(Pd8,Pd9,Pd10,Pd11,Pd12,Pd13,Pd14,Pd15) \` `( ((uint32)(Pd15) << (uint32)28) | ((uint32)(Pd14) <<`

		(uint32)24) | \ ((uint32)(Pd13) << (uint32)20) | ((uint32)(Pd12) << (uint32)16) | \ ((uint32)(Pd11) << (uint32)12)| ((uint32)(Pd10) << (uint32)8) | \ ((uint32)(Pd9) << (uint32)4) | (uint32)(Pd8) \ \ )

1.2.3 Macro: Port_lDiscSet(b0,b1,b2,b3,b4,b5,b6,b7,b8,b9,b10,b11,b12,b13,b14,b15)

Table 33 Port_lDiscSet(b0,b1,b2,b3,b4,b5,b6,b7,b8,b9,b10,b11,b12,b13,b14,b15)

Name	Port_lDiscSet(b0,b1,b2,b3,b4,b5,b6,b7,b8,b9,b10,b11,b12,b13,b14,b15)	
Description	Function like macro definition for PORT Pad Decision Control Register PDISC.	
Verification method	The macro is generated to set the corresponding bits of PDISC register of available ports where b<x> corresponds to each bit and x = 0 – 15. *Note: The macro is not user configurable.*	
Example(s)	**Action**	**Generated output**
	Generate Port_PBcfg.c file	#define Port_lDiscSet(b0,b1,b2,b3,b4,b5,b6, b7,b8,b9,b10,b11,b12,b13,b14,b15) \ ((uint16)(\ (uint32)(b0) | ((uint32)(b1) << (uint32)1) | \ ((uint32)(b2) << (uint32)2) | ((uint32)(b3) << (uint32)3) | \ ((uint32)(b4) << (uint32)4) | ((uint32)(b5) << (uint32)5) | \ ((uint32)(b6) << (uint32)6) | ((uint32)(b7) << (uint32)7) | \ ((uint32)(b8) << (uint32)8) | ((uint32)(b9) << (uint32)9) | \

		``((uint32)(b10) << (uint32)10) | ((uint32)(b11) << (uint32)11) | \ ((uint32)(b12) << (uint32)12) | ((uint32)(b13)<< (uint32)13) | \ ((uint32)(b14) << (uint32)14) | ((uint32)(b15) << (uint32)15) \ ))```

1.2.4 Macro: Port_lPcsr(b0,b1,b2,b3,b4,b5,b6,b7,b8,b9,b10,b11,b12,b13,b14,b15)

Table 34 Port_lPcsr(b0,b1,b2,b3,b4,b5,b6,b7,b8,b9,b10,b11,b12,b13,b14,b15)

Name	Port_lPcsr(b0,b1,b2,b3,b4,b5,b6,b7,b8,b9,b10,b11,b12,b13,b14,b15)	
Description	Function like macro definition for PORT Pin Controller Select Register PCSR.	
Verification method	The macro is generated to set the corresponding bits of PCSR register of each available port where b<x> corresponds to each bit and x = 0 – 15. *Note: The macro is not user configurable.*	
Example(s)	**Action**	**Generated output**
	Generate Port_PBcfg.c file	```#define Port_lPcsr(b0,b1,b2,b3,b4,b5,b6,b7, b8,b9,b10,b11,b12,b13,b14,b15) \ ((uint32)(\ ((uint32)(b0)) | ((uint32)(b1) << (uint32)1) | \ ((uint32)(b2) << (uint32)2) | ((uint32)(b3) << (uint32)3) | \ ((uint32)(b4) << (uint32)4) | ((uint32)(b5) << (uint32)5) | \ ((uint32)(b6) << (uint32)6) | ((uint32)(b7) << (uint32)7) | \ ((uint32)(b8) << (uint32)8) | ((uint32)(b9) << (uint32)9) | \ ((uint32)(b10) << (uint32)10) | ((uint32)(b11) <<```

		(uint32)11) | \ ((uint32)(b12) \<\< (uint32)12) | ((uint32)(b13) \<\< (uint32)13) | \ ((uint32)(b14) \<\< (uint32)14) | ((uint32)(b15) \<\< (uint32)15) \ ))

1.2.5 Macro: Port_lLvds(REN_CTRL,RX_EN,TERM,LRTXERM,LVDSM,PS,TEN_CTRL,TX_EN, VIDFFADJ,VOSDYN,VOSEXT,TX_PD,TX_PWDPD)

Table 35 Port_lLvds(REN_CTRL,RX_EN,TERM,LRTXERM,LVDSM,PS,TEN_CTRL,TX_EN, VIDFFADJ,VOSDYN,VOSEXT,TX_PD,TX_PWDPD)

Name	Port_lLvds(REN_CTRL,RX_EN,TERM,LRTXERM,LVDSM,PS,TEN_CTRL,TX_EN, VIDFFADJ,VOSDYN,VOSEXT,TX_PD,TX_PWDPD)	
Description	Macro definition for configuring LPCR register for LVDS support.	
Verification method	The macro is generated to set the bits of LPCR register. ((uint32)(\ ((uint32)(REN_CTRL)) | ((uint32)(RX_EN) << (uint32)1) | \ ((uint32)(TERM) << (uint32)2) | ((uint32)(LRTXERM) << (uint32)3) | \ ((uint32)(LVDSM) << (uint32)6) |((uint32)(PS) << (uint32)7) | \ ((uint32)(TEN_CTRL) << (uint32)8) | \ ((uint32)(TX_EN) << (uint32)9) |((uint32)(VIDFFADJ) << (uint32)10) |\ ((uint32)(VOSDYN) << (uint32)12) | ((uint32)(VOSEXT) << (uint32)13)|\ ((uint32)(TX_PD) << (uint32)14) | ((uint32)(TX_PWDPD) << (uint32)15)\ )) *Note: The macro is not user configurable.*	
Example(s)	**Action**	**Generated output**
	Generate Port_PBcfg.c file	#define Port_lLvds(REN_CTRL,RX_EN,TERM,LRTXERM,LVDSM,PS,TEN_CTRL,TX_EN,\ VIDFFADJ,VOSDYN,VOSEXT,TX_PD,TX_PWDPD) \ ((uint32)(\ ((uint32)(REN_CTRL)) | ((uint32)(RX_EN) << (uint32)1) | \ ((uint32)(TERM) << (uint32)2) | ((uint32)(LRTXERM) <<

	(uint32)3) | \ ((uint32)(LVDSM) << (uint32)6) |((uint32)(PS) << (uint32)7) | \ ((uint32)(TEN_CTRL) << (uint32)8) | \ ((uint32)(TX_EN) << (uint32)9) |((uint32)(VIDFFADJ) << (uint32)10) |\ ((uint32)(VOSDYN) << (uint32)12) | ((uint32)(VOSEXT) << (uint32)13)|\ ((uint32)(TX_PD) << (uint32)14) | ((uint32)(TX_PWDPD) << (uint32)15)\ ))

1.2.6 Macro: PORT_PIN_DEFAULT

Table 36 PORT_PIN_DEFAULT

Name	PORT_PIN_DEFAULT	
Description	Macro to indicate invalid pins of valid ports	
Verification method	The macro is generated as 0x10U. *Note: The macro is not user configurable.*	
Example(s)	**Action**	**Generated output**
	Generate Port_PBcfg.c file	`#define PORT_PIN_DEFAULT (0x10U)`

1.2.7 Macro: PORT_PIN_LEVEL_LOW

Table 37 PORT_PIN_LEVEL_LOW

Name	PORT_PIN_LEVEL_LOW	
Description	Defines physical pin low level	
Verification method	The macro is generated as 0x00U. *Note: The macro is not user configurable.*	
Example(s)	**Action**	**Generated output**
	Generate Port_PBcfg.c file	`#define PORT_PIN_LEVEL_LOW (0x00U)`

1.2.8 Macro: PORT_PIN_LEVEL_HIGH

Table 38 PORT_PIN_LEVEL_HIGH

Name	PORT_PIN_LEVEL_HIGH	
Description	Defines physical pin high level	
Verification method	The macro is generated as 0x01U. *Note: The macro is not user configurable.*	
Example(s)	**Action**	**Generated output**
	Generate Port_PBcfg.c file	`#define PORT_PIN_LEVEL_HIGH (0x01U)`

1.2.9 Macro: PORT_PIN_IN_PULL_UP

Table 39 PORT_PIN_IN_PULL_UP

Name	PORT_PIN_IN_PULL_UP	
Description	Defines pull up characteristic of port pin	
Verification method	The macro is generated as 0x10U. *Note: The macro is not user configurable.*	
Example(s)	**Action**	**Generated output**
	Generate Port_PBcfg.c file	`#define PORT_PIN_IN_PULL_UP (0x10U)`

1.2.10 Macro: PORT_PIN_IN_PULL_DOWN

Table 40 PORT_PIN_IN_PULL_DOWN

Name	PORT_PIN_IN_PULL_DOWN	
Description	Define pull down characteristic of port pin.	
Verification method	The macro is generated as 0x08U. *Note: The macro is not user configurable.*	
Example(s)	**Action**	**Generated output**
	Generate Port_PBcfg.c file	`#define PORT_PIN_IN_PULL_DOWN (0x08U)`

1.2.11 Macro: PORT_PIN_IN_NO_PULL

Table 41 PORT_PIN_IN_NO_PULL

Name	PORT_PIN_IN_NO_PULL

Description	Defines no pull characteristic of port pin.	
Verification method	The macro is generated as 0x00U. *Note: The macro is not user configurable.*	
Example(s)	**Action**	**Generated output**
	Generate Port_PBcfg.c file	`#define PORT_PIN_IN_NO_PULL (0x00U)`

1.2.12 Macro: RFAST_PORT_PIN_STRONG_DRIVER_SHARP_EDGE

Table 42 RFAST_PORT_PIN_STRONG_DRIVER_SHARP_EDGE

Name	RFAST_PORT_PIN_STRONG_DRIVER_SHARP_EDGE	
Description	Defines the RFast pin mode as strong driver sharp edge	
Verification method	The macro is generated as 0x0U. *Note: The macro is not user configurable.*	
Example(s)	**Action**	**Generated output**
	Generate Port_PBcfg.c file	`#define RFAST_PORT_PIN_STRONG_DRIVER_SHARP_EDGE (0x0U)`

1.2.13 Macro: RFAST_PORT_PIN_STRONG_DRIVER_MEDIUM_EDGE

Table 43 RFAST_PORT_PIN_STRONG_DRIVER_MEDIUM_EDGE

Name	RFAST_PORT_PIN_STRONG_DRIVER_MEDIUM_EDGE	
Description	Defines the RFast pin mode as strong driver medium edge.	
Verification method	The macro is generated as 0x1U. *Note: The macro is not user configurable.*	
Example(s)	**Action**	**Generated output**
	Generate Port_PBcfg.c file	`#define RFAST_PORT_PIN_STRONG_DRIVER_MEDIUM_EDGE (0x1U)`

1.2.14 Macro: RFAST_PORT_PIN_MEDIUM_DRIVER

Table 44 RFAST_PORT_PIN_MEDIUM_DRIVER

Name	RFAST_PORT_PIN_MEDIUM_DRIVER
Description	Defines RFast pin mode as Medium edge.
Verification method	The macro is generated as 0x2U.

	Note: *The macro is not user configurable.*	
Example(s)	**Action**	**Generated output**
	Generate Port_PBcfg.c file	`#define RFAST_PORT_PIN_MEDIUM_DRIVER (0x2U)`

1.2.15 Macro: RFAST_PORT_PIN_RGMII_DRIVER

Table 45 RFAST_PORT_PIN_RGMII_DRIVER

Name	PORT_TOTAL_AVAILABLE_PORTS	
Description	Defines RGMII pad driver mode.	
Verification method	The macro is generated as 0x3U. *Note:* *The macro is not user configurable.*	
Example(s)	**Action**	**Generated output**
	Generate Port_PBcfg.c file	`#define RFAST_PORT_PIN_RGMII_DRIVER (0x3U)`

1.2.16 Macro: RFAST_PORT_PIN_DEFAULT_DRIVER

Table 46 RFAST_PORT_PIN_DEFAULT_DRIVER

Name	RFAST_PORT_PIN_MEDIUM_DRIVER	
Description	Defines RFast pin default mode.	
Verification method	The macro is generated as 0x0U. *Note:* *The macro is not user configurable.*	
Example(s)	**Action**	**Generated output**
	Generate Port_PBcfg.c file	`#define RFAST_PORT_PIN_DEFAULT_DRIVER (0x0U)`

1.2.17 Macro: FAST_PORT_PIN_STRONG_DRIVER_SHARP_EDGE

Table 47 FAST_PORT_PIN_STRONG_DRIVER_SHARP_EDGE

Name	FAST_PORT_PIN_STRONG_DRIVER_SHARP_EDGE
Description	Defines Fast pin mode as strong driver sharp edge.
Verification method	The macro is generated as 0x0U. *Note:* *The macro is not user configurable.*

Example(s)	Action	Generated output
	Generate Port_PBcfg.c file	`#define PORT_TOTAL_AVAILABLE_PORTS (16U)`

1.2.18 Macro: FAST_PORT_PIN_STRONG_DRIVER_MEDIUM_EDGE

Table 48 FAST_PORT_PIN_STRONG_DRIVER_MEDIUM_EDGE

Name	FAST_PORT_PIN_STRONG_DRIVER_MEDIUM_EDGE	
Description	Defines Fast pin mode as strong driver medium edge.	
Verification method	The macro is generated as 0x1U. *Note: The macro is not user configurable.*	
Example(s)	**Action**	**Generated output**
	Generate Port_PBcfg.c file	`#define FAST_PORT_PIN_STRONG_DRIVER_MEDIUM_EDGE (0x1U)`

1.2.19 Macro: FAST_PORT_PIN_MEDIUM_DRIVER

Table 49 FAST_PORT_PIN_MEDIUM_DRIVER

Name	FAST_PORT_PIN_MEDIUM_DRIVER	
Description	Defines Fast pin mode as driver medium driver.	
Verification method	The macro is generated as 0x2U. *Note: The macro is not user configurable.*	
Example(s)	**Action**	**Generated output**
	Generate Port_PBcfg.c file	`#define FAST_PORT_PIN_MEDIUM_DRIVER (0x2U)`

1.2.20 Macro: FAST_PORT_PIN_DEFAULT_DRIVER

Table 50 FAST_PORT_PIN_DEFAULT_DRIVER

Name	FAST_PORT_PIN_DEFAULT_DRIVER	
Description	Defines Fast pin default mode.	
Verification method	The macro is generated as 0x0U. *Note: The macro is not user configurable.*	
Example(s)	**Action**	**Generated output**
	Generate Port_PBcfg.c file	`#define FAST_PORT_PIN_DEFAULT_DRIVER (0x0U)`

1.2.21 Macro: SLOW_PORT_PIN_MEDIUM_DRIVER_SHARP_EDGE

Table 51 SLOW_PORT_PIN_MEDIUM_DRIVER_SHARP_EDGE

Name	SLOW_PORT_PIN_MEDIUM_DRIVER_SHARP_EDGE	
Description	Defines Slow pin mode as medium driver sharp edge.	
Verification method	The macro is generated as 0x0U. *Note: The macro is not user configurable.*	
Example(s)	**Action**	**Generated output**
	Generate Port_PBcfg.c file	`#define PORT_TOTAL_AVAILABLE_PORTS (16U)`

1.2.22 Macro: SLOW_PORT_PIN_MEDIUM_DRIVER

Table 52 SLOW_PORT_PIN_MEDIUM_DRIVER

Name	SLOW_PORT_PIN_MEDIUM_DRIVER	
Description	Defines Slow pin mode as medium driver.	
Verification method	The macro is generated as 0x1U. *Note: The macro is not user configurable.*	
Example(s)	**Action**	**Generated output**
	Generate Port_PBcfg.c file	`#define SLOW_PORT_PIN_MEDIUM_DRIVER (0x1U)`

1.2.23 Macro: SLOW_PORT_PIN_DEFAULT_DRIVER

Table 53 SLOW_PORT_PIN_DEFAULT_DRIVER

Name	SLOW_PORT_PIN_DEFAULT_DRIVER	
Description	Defines Slow pin default mode.	
Verification method	The macro is generated as 0x0U. *Note: The macro is not user configurable.*	
Example(s)	**Action**	**Generated output**
	Generate Port_PBcfg.c file	`#define SLOW_PORT_PIN_DEFAULT_DRIVER (0x0U)`

1.2.24 Macro: PORT_PIN_PAD_STRENGTH_DEFAULT

Table 54 PORT_PIN_PAD_STRENGTH_DEFAULT

Name	PORT_PIN_PAD_STRENGTH_DEFAULT

Description	Defines default pad strength for input pins	
Verification method	The macro is generated as 0x0U. *Note: The macro is not user configurable.*	
Example(s)	**Action**	**Generated output**
	Generate Port_PBcfg.c file	`#define PORT_PIN_PAD_STRENGTH_DEFAULT (0x0U)`

1.2.25 Macro: PORT_PIN_PAD_LEVEL_DEFAULT

Table 55 PORT_PIN_PAD_LEVEL_DEFAULT

Name	PORT_PIN_PAD_LEVEL_DEFAULT	
Description	Defines default pad level for output pins.	
Verification method	The macro is generated as 0x0U. *Note: The macro is not user configurable.*	
Example(s)	**Action**	**Generated output**
	Generate Port_PBcfg.c file	`#define PORT_PIN_PAD_LEVEL_DEFAULT (0x0U)`

1.2.26 Macro: PORT_PIN_PAD_DEFAULT

Table 56 PORT_PIN_PAD_DEFAULT

Name	PORT_PIN_PAD_DEFAULT	
Description	Defines pad strength for port pin which is not available.	
Verification method	The macro is generated as 0x0U. *Note: The macro is not user configurable.*	
Example(s)	**Action**	**Generated output**
	Generate Port_PBcfg.c file	`#define PORT_PIN_PAD_DEFAULT (0x0U)`

1.2.27 Macro: PORT_PIN_OUT_PUSHPULL

Table 57 PORT_PIN_OUT_PUSHPULL

Name	PORT_PIN_OUT_PUSHPULL
Description	Defines port output pin push pull characteristic.
Verification method	The macro is generated as 0x00U.

	Note: *The macro is not user configurable.*	
Example(s)	**Action**	**Generated output**
	Generate Port_PBcfg.c file	`#define PORT_PIN_OUT_PUSHPULL (0x00U)`

1.2.28 Macro: PORT_PIN_OUT_OPENDRAIN

Table 58 PORT_PIN_OUT_OPENDRAIN

Name	PORT_PIN_OUT_OPENDRAIN	
Description	Defines port output pin open drain characteristic.	
Verification method	The macro is generated as 0x40U. *Note:* *The macro is not user configurable.*	
Example(s)	**Action**	**Generated output**
	Generate Port_PBcfg.c file	`#define PORT_PIN_OUT_OPENDRAIN (0x40U)`

1.2.29 Macro: PORT_INPUT_LEVEL_CMOS_AUTOMOTIVE

Table 59 PORT_INPUT_LEVEL_CMOS_AUTOMOTIVE

Name	PORT_INPUT_LEVEL_CMOS_AUTOMOTIVE	
Description	Defines automotive level input pad supply voltage	
Verification method	The macro is generated as 0x0U. *Note:* *The macro is not user configurable.*	
Example(s)	**Action**	**Generated output**
	Generate Port_PBcfg.c file	`#define PORT_INPUT_LEVEL_CMOS_AUTOMOTIVE (0x0U)`

1.2.30 Macro: PORT_INPUT_LEVEL_TTL_3_3V

Table 60 PORT_INPUT_LEVEL_TTL_3_3V

Name	PORT_INPUT_LEVEL_TTL_3_3V
Description	Defines TTL level for 3.3V pad supply.
Verification method	The macro is generated as 0xCU. *Note:* *The macro is not user configurable.*

Example(s)	Action	Generated output
	Generate Port_PBcfg.c file	`#define PORT_INPUT_LEVEL_TTL_3_3V (0xCU)`

1.2.31 Macro: PORT_INPUT_LEVEL_TTL_5_0V

Table 61 PORT_INPUT_LEVEL_TTL_5_0V

Name	PORT_INPUT_LEVEL_TTL_5_0V	
Description	Defines TTL level for 5V pad supply.	
Verification method	The macro is generated as 0x8U. *Note: The macro is not user configurable.*	
Example(s)	**Action**	**Generated output**
	Generate Port_PBcfg.c file	`#define PORT_INPUT_LEVEL_TTL_5_0V (0x8U)`

1.2.32 Macro: PORT_RGMII_INPUT

Table 62 PORT_RGMII_INPUT

Name	PORT_RGMII_INPUT	
Description	Defines only for pads with RGMII input buffer.	
Verification method	The macro is generated as 0x03U. *Note: The macro is not user configurable.*	
Example(s)	**Action**	**Generated output**
	Generate Port_PBcfg.c file	`#define PORT_RGMII_INPUT (0x03U)`

1.2.33 Macro: PORT_PIN_ANALOG_INPUT_ENABLE

Table 63 PORT_PIN_ANALOG_INPUT_ENABLE

Name	PORT_PIN_ANALOG_INPUT_ENABLE	
Description	Enables analog input for port pin	
Verification method	The macro is generated as 0x1U. *Note: The macro is not user configurable.*	
Example(s)	**Action**	**Generated output**
	Generate Port_PBcfg.c file	`#define PORT_PIN_ANALOG_INPUT_ENABLE (0x1U)`

1.2.34 Macro: PORT_PIN_ANALOG_INPUT_DISABLE

Table 64 PORT_PIN_ANALOG_INPUT_DISABLE

Name	PORT_PIN_ANALOG_INPUT_DISABLE	
Description	Disables analog input for port pin	
Verification method	The macro is generated as 0x0U. *Note: The macro is not user configurable.*	
Example(s)	**Action**	**Generated output**
	Generate Port_PBcfg.c file	`#define PORT_PIN_ANALOG_INPUT_DISABLE (0x0U)`

1.2.35 Macro: PORT_PCSR_ENABLE

Table 65 PORT_PCSR_ENABLE

Name	PORT_PCSR_ENABLE	
Description	Enables PCSR support for port pin	
Verification method	The macro is generated as 0x1U. *Note: The macro is not user configurable.*	
Example(s)	**Action**	**Generated output**
	Generate Port_PBcfg.c file	`#define PORT_PCSR_ENABLE (0x1U)`

1.2.36 Macro: PORT_PCSR_DISABLE

Table 66 PORT_PCSR_DISABLE

Name	PORT_PCSR_DISABLE	
Description	Disables PCSR support for port pin	
Verification method	The macro is generated as 0x0U. *Note: The macro is not user configurable.*	
Example(s)	**Action**	**Generated output**
	Generate Port_PBcfg.c file	`#define PORT_PCSR_DISABLE (0x0U)`

1.2.37 Macro: PORT_PCSR_DEFAULT

Table 67 PORT_PCSR_DEFAULT

Name	PORT_PCSR_DEFAULT
Description	Corresponds to port pins which do not have PCSR support

Verification method	The macro is generated as 0x0. *Note: The macro is not user configurable.*	
Example(s)	**Action**	**Generated output**
	Generate Port_PBcfg.c file	`#define PORT_PCSR_DEFAULT (0x0U)`

1.2.38 Macro: PORT_LVDS_PORT_CONTROLLED

Table 68 PORT_LVDS_PORT_CONTROLLED

Name	PORT_LVDS_PORT_CONTROLLED	
Description	LVDS pin pair is controlled by pad control register.	
Verification method	The macro is generated as 0x0U. *Note: The macro is not user configurable.*	
Example(s)	**Action**	**Generated output**
	Generate Port_PBcfg.c file	`#define PORT_LVDS_PORT_CONTROLLED (0x0U)`

1.2.39 Macro: PORT_LVDS_HSCT_CONTROLLED

Table 69 PORT_LVDS_HSCT_CONTROLLED

Name	PORT_LVDS_HSCT_CONTROLLED	
Description	LVDS pin pair is controlled by HSCT module.	
Verification method	The macro is generated as 0x1U. *Note: The macro is not user configurable.*	
Example(s)	**Action**	**Generated output**
	Generate Port_PBcfg.c file	`#define PORT_LVDS_HSCT_CONTROLLED (0x1U)`

1.2.40 Macro: PORT_LVDS_ENABLE

Table 70 PORT_LVDS_ENABLE

Name	PORT_LVDS_ENABLE	
Description	Enables LVDS supported pin pair	
Verification method	The macro is generated as 0x1U. *Note: The macro is not user configurable.*	
Example(s)	**Action**	**Generated output**

	Generate Port_PBcfg.c file	`#define PORT_LVDS_ENABLE (0x1U)`

1.2.41 Macro: PORT_LVDS_DISABLE

Table 71 PORT_LVDS_DISABLE

Name	PORT_LVDS_DISABLE	
Description	Disables LVDS supported pin pair	
Verification method	The macro is generated as 0x0U. *Note: The macro is not user configurable.*	
Example(s)	**Action**	**Generated output**
	Generate Port_PBcfg.c file	`#define PORT_LVDS_DISABLE (0x0U)`

1.2.42 Macro: PORT_LVDS_INTERNAL_TERMINATION

Table 72 PORT_LVDS_INTERNAL_TERMINATION

Name	PORT_LVDS_INTERNAL_TERMINATION	
Description	Receiver internal termination mode.	
Verification method	The macro is generated as 0x1U. *Note: The macro is not user configurable.*	
Example(s)	**Action**	**Generated output**
	Generate Port_PBcfg.c file	`#define PORT_LVDS_INTERNAL_TERMINATION (0x1U)`

1.2.43 Macro: PORT_LVDS_EXTERNAL_TERMINATION

Table 73 PORT_LVDS_EXTERNAL_TERMINATION

Name	PORT_LVDS_EXTERNAL_TERMINATION	
Description	Receiver external termination mode.	
Verification method	The macro is generated as 0x0U. *Note: The macro is not user configurable.*	
Example(s)	**Action**	**Generated output**
	Generate Port_PBcfg.c file	`#define PORT_LVDS_EXTERNAL_TERMINATION (0x0U)`

1.2.44 Macro: PORT_LVDS_POLY_RESISTOR_TERM

Table 74 PORT_LVDS_POLY_RESISTOR_TERM

Name	PORT_LVDS_POLY_RESISTOR_TERM	
Description	LVDS RX poly-resistor configuration value.	
Verification method	The macro is generated as 0x0U. *Note: The macro is not user configurable.*	
Example(s)	**Action**	**Generated output**
	Generate Port_PBcfg.c file	`#define PORT_LVDS_POLY_RESISTOR_TERM (0x0U)`

1.2.45 Macro: PORT_LVDS_LVDSM

Table 75 PORT_LVDS_LVDSM

Name	PORT_LVDS_LVDSM	
Description	Reduced frequency mode of the receiver.	
Verification method	The macro is generated as 0x1U. *Note: The macro is not user configurable.*	
Example(s)	**Action**	**Generated output**
	Generate Port_PBcfg.c file	`#define PORT_LVDS_LVDSM (0x1U)`

1.2.46 Macro: PORT_LVDS_LVDSH

Table 76 PORT_LVDS_LVDSH

Name	PORT_LVDS_LVDSH	
Description	High frequency mode of receiver	
Verification method	The macro is generated as 0x0U. *Note: The macro is not user configurable.*	
Example(s)	**Action**	**Generated output**
	Generate Port_PBcfg.c file	`#define PORT_LVDS_LVDSH (0x0U)`

1.2.47 Macro: PAD_SUPPLY_3_3V

Table 77 PAD_SUPPLY_3_3V

Name	PAD_SUPPLY_3_3V
Description	3.3V supply voltage for pad-pair.

Verification method	The macro is generated as 0x0U. *Note: The macro is not user configurable.*	
Example(s)	**Action**	**Generated output**
	Generate Port_PBcfg.c file	`#define PAD_SUPPLY_3_3V (0x0U)`

1.2.48 Macro: PAD_SUPPLY_5_0V

Table 78 PAD_SUPPLY_5_0V

Name	PAD_SUPPLY_5_0V	
Description	5V supply voltage for pad pair.	
Verification method	The macro is generated as 0x1U. *Note: The macro is not user configurable.*	
Example(s)	**Action**	**Generated output**
	Generate Port_PBcfg.c file	`#define PAD_SUPPLY_5_0V (0x1U)`

1.2.49 Macro: PORT_LVDS_LVDS_VDIFF_ADJ

Table 79 PORT_LVDS_LVDS_VDIFF_ADJ

Name	PORT_LVDS_LVDS_VDIFF_ADJ	
Description	LVDS output amplitude is adjusted to 3.2 mA	
Verification method	The macro is generated as 0x1U. *Note: The macro is not user configurable.*	
Example(s)	**Action**	**Generated output**
	Generate Port_PBcfg.c file	`#define PORT_LVDS_LVDS_VDIFF_ADJ (0x1U)`

1.2.50 Macro: PORT_LVDS_VOSDYN

Table 80 PORT_LVDS_VOSDYN

Name	PORT_LVDS_VOSDYN	
Description	Defines output voltage offset change mode as dynamic.	
Verification method	The macro is generated as 0x1U (Dynamic mode). *Note: The macro is not user configurable.*	
Example(s)	**Action**	**Generated output**

	Generate Port_PBcfg.c file	`#define PORT_LVDS_VOSDYN (0x1U)`

1.2.51 Macro: PORT_LVDS_VOSEXT

Table 81 PORT_LVDS_VOSEXT

Name	PORT_LVDS_VOSEXT	
Description	Defines tune bit to change the measurement point of output offset mode to internal replica.	
Verification method	The macro is generated as 0x0U. *Note: The macro is not user configurable.*	
Example(s)	**Action**	**Generated output**
	Generate Port_PBcfg.c file	`#define PORT_LVDS_VOSEXT (0x0U)`

1.2.52 Macro: PORT_LVDS_POWER_DOWN

Table 82 PORT_LVDS_POWER_DOWN

Name	PORT_LVDS_POWER_DOWN	
Description	LVDS power down	
Verification method	The macro is generated as 0x1U. *Note: The macro is not user configurable.*	
Example(s)	**Action**	**Generated output**
	Generate Port_PBcfg.c file	`#define PORT_LVDS_POWER_DOWN (0x1U)`

1.2.53 Macro: PORT_LVDS_TX_PWDPD_ENABLE

Table 83 PORT_LVDS_TX_PWDPD_ENABLE

Name	PORT_LVDS_TX_PWDPD_ENABLE	
Description	Enabled TX power down pull down resistor.	
Verification method	The macro is generated as 0x1U. *Note: The macro is not user configurable.*	
Example(s)	**Action**	**Generated output**
	Generate Port_PBcfg.c file	`#define PORT_LVDS_TX_PWDPD_ENABLE (0x1U)`

1.2.54 Macro: PORT_LVDS_TX_PWDPD_DISABLE

Table 84 PORT_LVDS_TX_PWDPD_DISABLE

Name	PORT_LVDS_TX_PWDPD_DISABLE	
Description	Disabled TX power down pull down resistor.	
Verification method	The macro is generated as 0x0U. *Note: The macro is not user configurable.*	
Example(s)	**Action**	**Generated output**
	Generate Port_PBcfg.c file	`#define PORT_LVDS_TX_PWDPD_DISABLE (0x0U)`

1.2.55 Macro: PORT_LVDS_DEFAULT

Table 85 PORT_LVDS_DEFAULT

Name	PORT_LVDS_DEFAULT	
Description	Default value for LVDS Pad control register bits.	
Verification method	The macro is generated as 0x0U. *Note: The macro is not user configurable.*	
Example(s)	**Action**	**Generated output**
	Generate Port_PBcfg.c file	`#define PORT_LVDS_DEFAULT (0x0U)`

1.2.56 Structure: Port_Config[_<variant>]

Table 86 Port_Config[_<variant>]

Name	Port_Config[_<variant>]	
Type	Port_ConfigType	
Description	Root configuration structure of PORT driver which will be used during initialization.	
Verification method	The generated structure is present in Port [_<variant>]_PBcfg.c file. The <variant> indicates the name of the post-build variant. For a variant-aware configuration the structure name is appended with the variant name. For variant-unaware configuration <variant> is ignored.	
Example(s)	**Action**	**Generated output**
	Configure PORT driver and generate. (variant-unaware)	`const Port_ConfigType Port_Config =` `{` `  /* Port Configuration set  */` `  &Port_kConfiguration[0],` `  /* Port 40-th Disc configuration set  */` `  &Port_DiscSet[0],` `  &Port_kPinSupportedModes[0],` `  /* LVDS configuration */` `  &Port_kLVDSConfig[0],` `  /* PCSR Configuration */` `  &Port_kPCSRConfig[0]` `};`
	Configure PORT driver and generate. (variant-aware. Variant name is ‘Petrol’)	`const Port_ConfigType Port_Config_Petrol =` `{` `  /* Port Configuration set  */` `  &Port_kConfiguration[0],` `  /* Port 40-th Disc configuration set  */` `  NULL_PTR,` `  /* LVDS configuration */` `  NULL_PTR,` `  /* PCSR Configuration */` `  &Port_kPCSRConfig[0]`

		};

1.2.56.1 Member: Port_kConfiguration[NumberOfPortsAvailable]

Table 87 Port_kConfiguration[NumberOfPortsAvailable]

Name	Port_kConfiguration[NumberOfPortsAvailable]	
Type	Port_n_ConfigType	
Description	Array of port specific configuration.	
Verification method	The generated structure member is present in the Port_Config[_<variant>] structure. Array size depends on the number of ports available in the microcontroller. For an available port the structure includes the pin configuration of that port. The generated structure member contains an array entry for each user-configured Port at 'PortContainer' container. IOCR register configuration is generated by doing OR operation on 'PortPinDirection', 'PortPinOutputPinDriveMode' and 'PortPinInitialMode' for output pins and 'PortPinInputPullResistor', 'PortPinDirection' and 'PortPinInitialMode' for input pins Configured value in 'PortPinLevelValue' is taken as the initial level configuration. Port pin driver strength for the pin 0 to 7 is derived in the function like macro 'Portx_lPdrConfig1' and pin 8 to 15 in 'Portx_lPdrConfig2'. And the parameters passed to the macro is derived by OR operation on the configuration parameter PortPinInputPadLevel and PortPinOutputPadDriveStrength values of respective pins. If a pin is configured with 'PortPinModeChangeable' enabled then mode of that pin can be changed runtime. If a pin is configured with 'PortPinEmergencyStop' enabled then that pin can be stopped at runtime	
Example(s)	**Action**	**Generated output**
	Available ports : 0 Input Pins: 3, 5, 6, 7, 8, 9, 10, 12, 13 Output Pins: 0, 1, 2, 4, 11,14,15 Pins configured PortPinOutputPinDriveMode as PORT_PIN_OUT_PUSHPULL: 0,	`static const Port_n_ConfigType Port_kConfiguration[] =` `{` `  /*  Port0 */` `  {` `  {` `  /* Port pins direction, characteristics and mode`

1, 2, 4,11 PORT_PIN_OUT_OPENDRAIN: 14, 15 Pins configured PortPinInputPullResistor as PORT_PIN_IN_PULL_UP: 3, 5, 6, 7, 8 PORT_PIN_IN_PULL_DOWN: 9, 10 PORT_PIN_IN_NO_PULL: 12, 13 Pins configured PortPinInitialMode as PORT_PIN_MODE_GPIO: 1, 3, 5 to 15 PORT_PIN_MODE_ALT1: 0, 2 PORT_PIN_MODE_ALT2: 4 Pins configured PortPinLevelValue as PORT_PIN_LEVEL_HIGH: 0 PORT_PIN_LEVEL_LOW: 1 to 15 Pin configured PortPinInputPadLevel as PORT_INPUT_LEVEL_CMOS_AUTOMOTIVE: 3, 5, 6, 7, 8, 9, 10 PORT_INPUT_LEVEL_TTL_3_3V: 12 PORT_INPUT_LEVEL_TTL_5_0V: 13 Pin configured PortPinModeChangeable as True: 0 False: 1 to 15 Pin configured PortPinEmergencyStop as True: 0 False: 1 to 15	configuration */ ((uint8)PORT_PIN_OUT | PORT_PIN_OUT_PUSHPULL | PORT_PIN_MODE_ALT1),/*Pin 0*/ ((uint8)PORT_PIN_OUT | PORT_PIN_OUT_PUSHPULL | PORT_PIN_MODE_GPIO),/*Pin 1*/ ((uint8)PORT_PIN_OUT | PORT_PIN_OUT_PUSHPULL | PORT_PIN_MODE_ALT1),/*Pin 2*/ ((uint8)PORT_PIN_IN | PORT_PIN_IN_PULL_UP | PORT_PIN_MODE_GPIO),/*Pin 3*/ ((uint8)PORT_PIN_OUT | PORT_PIN_OUT_PUSHPULL | PORT_PIN_MODE_ALT2),/*Pin 4*/ ((uint8)PORT_PIN_IN | PORT_PIN_IN_PULL_UP | PORT_PIN_MODE_GPIO),/*Pin 5*/ ((uint8)PORT_PIN_IN | PORT_PIN_IN_PULL_UP | PORT_PIN_MODE_GPIO),/*Pin 6*/ ((uint8)PORT_PIN_IN | PORT_PIN_IN_PULL_UP | PORT_PIN_MODE_GPIO),/*Pin 7*/ ((uint8)PORT_PIN_IN | PORT_PIN_IN_PULL_UP | PORT_PIN_MODE_GPIO),/*Pin 8*/ ((uint8)PORT_PIN_IN | PORT_PIN_IN_PULL_DOWN | PORT_PIN_MODE_GPIO),/*Pin 9*/ ((uint8)PORT_PIN_IN | PORT_PIN_IN_PULL_DOWN | PORT_PIN_MODE_GPIO),/*Pin 10*/ ((uint8)PORT_PIN_OUT | PORT_PIN_OUT_PUSHPULL | PORT_PIN_MODE_GPIO),/*Pin 11*/ ((uint8)PORT_PIN_IN | PORT_PIN_IN_NO_PULL | PORT_PIN_MODE_GPIO),/*Pin 12*/ ((uint8)PORT_PIN_IN | PORT_PIN_IN_NO_PULL | PORT_PIN_MODE_GPIO),/*Pin 13*/ ((uint8)PORT_PIN_OUT | PORT_PIN_OUT_OPENDRAIN | PORT_PIN_MODE_GPIO),/*Pin 14*/ ((uint8)PORT_PIN_OUT | PORT_PIN_OUT_OPENDRAIN |

```
PORT_PIN_MODE_GPIO)/*Pin 15*/
    },
    {
      /* Port pins initial level
configuration */
      PORT_PIN_LEVEL_HIGH,/* Pin 0 */
      PORT_PIN_LEVEL_LOW,/* Pin 1 */
      PORT_PIN_LEVEL_LOW,/* Pin 2 */
      PORT_PIN_LEVEL_LOW,/* Pin 3 */
      PORT_PIN_LEVEL_LOW,/* Pin 4 */
      PORT_PIN_LEVEL_LOW,/* Pin 5 */
      PORT_PIN_LEVEL_LOW,/* Pin 6 */
      PORT_PIN_LEVEL_LOW,/* Pin 7 */
      PORT_PIN_LEVEL_LOW,/* Pin 8 */
      PORT_PIN_LEVEL_LOW,/* Pin 9 */
      PORT_PIN_LEVEL_LOW,/* Pin 10 */
      PORT_PIN_LEVEL_LOW,/* Pin 11 */
      PORT_PIN_LEVEL_LOW,/* Pin 12 */
      PORT_PIN_LEVEL_LOW,/* Pin 13 */
      PORT_PIN_LEVEL_LOW,/* Pin 14 */
      PORT_PIN_LEVEL_LOW /* Pin 15 */
    },
  /* Port pins drive strength1
configuration */
    Portx_lPdrConfig1(

(PORT_PIN_PAD_LEVEL_DEFAULT|FAST_PORT
_PIN_DEFAULT_DRIVER)    ,/*Pin0*/

(PORT_PIN_PAD_LEVEL_DEFAULT|SLOW_PORT
_PIN_MEDIUM_DRIVER_SHARP_EDGE)
,/*Pin1*/

(PORT_PIN_PAD_LEVEL_DEFAULT|SLOW_PORT
_PIN_DEFAULT_DRIVER)    ,/*Pin2*/

(PORT_INPUT_LEVEL_CMOS_AUTOMOTIVE|POR
T_PIN_PAD_STRENGTH_DEFAULT)
,/*Pin3*/

(PORT_PIN_PAD_LEVEL_DEFAULT|SLOW_PORT
_PIN_MEDIUM_DRIVER_SHARP_EDGE)
,/*Pin4*/
```

```
(PORT_INPUT_LEVEL_CMOS_AUTOMOTIVE|POR
T_PIN_PAD_STRENGTH_DEFAULT)
,/*Pin5*/

(PORT_INPUT_LEVEL_CMOS_AUTOMOTIVE|POR
T_PIN_PAD_STRENGTH_DEFAULT)
,/*Pin6*/

(PORT_INPUT_LEVEL_CMOS_AUTOMOTIVE|POR
T_PIN_PAD_STRENGTH_DEFAULT)  /*Pin7*/
                      ),
    /* Port pins drive strength2
configuration */
    Portx_lPdrConfig2(

(PORT_INPUT_LEVEL_CMOS_AUTOMOTIVE|POR
T_PIN_PAD_STRENGTH_DEFAULT)
,/*Pin8*/

(PORT_INPUT_LEVEL_CMOS_AUTOMOTIVE|POR
T_PIN_PAD_STRENGTH_DEFAULT)
,/*Pin9*/

(PORT_INPUT_LEVEL_CMOS_AUTOMOTIVE|POR
T_PIN_PAD_STRENGTH_DEFAULT)
,/*Pin10*/

(PORT_PIN_PAD_LEVEL_DEFAULT|SLOW_PORT
_PIN_DEFAULT_DRIVER)    ,/*Pin11*/

(PORT_INPUT_LEVEL_TTL_3_3V|PORT_PIN_P
AD_STRENGTH_DEFAULT)  ,/*Pin12*/

(PORT_INPUT_LEVEL_TTL_5_0V|PORT_PIN_P
AD_STRENGTH_DEFAULT)  ,/*Pin13*/

(PORT_PIN_PAD_LEVEL_DEFAULT|SLOW_PORT
_PIN_MEDIUM_DRIVER)    ,/*Pin14*/

(PORT_PIN_PAD_LEVEL_DEFAULT|FAST_PORT
_PIN_MEDIUM_DRIVER)    /*Pin15*/
                      ),
    {/* Port pin run time mode
changeable or not configuration */


      PORT_PIN_MODE_CHANGEABLE ,/*
Pin 0 */
```

```
      PORT_PIN_MODE_NOT_CHANGEABLE
,/* Pin 1 */
      PORT_PIN_MODE_NOT_CHANGEABLE
,/* Pin 2 */
      PORT_PIN_MODE_NOT_CHANGEABLE
,/* Pin 3 */
      PORT_PIN_MODE_NOT_CHANGEABLE
,/* Pin 4 */
      PORT_PIN_MODE_NOT_CHANGEABLE
,/* Pin 5 */
      PORT_PIN_MODE_NOT_CHANGEABLE
,/* Pin 6 */
      PORT_PIN_MODE_NOT_CHANGEABLE
,/* Pin 7 */
      PORT_PIN_MODE_NOT_CHANGEABLE
,/* Pin 8 */
      PORT_PIN_MODE_NOT_CHANGEABLE
,/* Pin 9 */
      PORT_PIN_MODE_NOT_CHANGEABLE
,/* Pin 10 */
      PORT_PIN_MODE_NOT_CHANGEABLE
,/* Pin 11 */
      PORT_PIN_MODE_NOT_CHANGEABLE
,/* Pin 12 */
      PORT_PIN_MODE_NOT_CHANGEABLE
,/* Pin 13 */
      PORT_PIN_MODE_NOT_CHANGEABLE
,/* Pin 14 */
      PORT_PIN_MODE_NOT_CHANGEABLE
/* Pin 15 */


    },


    {/* Port pin run time mode
changeable or not configuration */



PORT_PIN_EMERGENCY_STOP_ENABLE,/* Pin
0 */

PORT_PIN_EMERGENCY_STOP_DISABLE,/*
Pin 1 */

PORT_PIN_EMERGENCY_STOP_DISABLE,/*
Pin 2 */
```

```
PORT_PIN_EMERGENCY_STOP_DISABLE,/*
Pin 3 */

PORT_PIN_EMERGENCY_STOP_DISABLE,/*
Pin 4 */

PORT_PIN_EMERGENCY_STOP_DISABLE,/*
Pin 5 */

PORT_PIN_EMERGENCY_STOP_DISABLE,/*
Pin 6 */

PORT_PIN_EMERGENCY_STOP_DISABLE,/*
Pin 7 */

PORT_PIN_EMERGENCY_STOP_DISABLE,/*
Pin 8 */

PORT_PIN_EMERGENCY_STOP_DISABLE,/*
Pin 9 */

PORT_PIN_EMERGENCY_STOP_DISABLE,/*
Pin 10 */

PORT_PIN_EMERGENCY_STOP_DISABLE,/*
Pin 11 */

PORT_PIN_EMERGENCY_STOP_DISABLE,/*
Pin 12 */

PORT_PIN_EMERGENCY_STOP_DISABLE,/*
Pin 13 */

PORT_PIN_EMERGENCY_STOP_DISABLE,/*
Pin 14 */
      PORT_PIN_EMERGENCY_STOP_DISABLE
/* Pin 15 */
    }
  }
}
```

1.2.56.2 Member: Port_DiscSet[NumberOfAnaogDigitalPortsAvailable]

Table 88 Port_DiscSet[NumberOfAnalogDigitalPortsAvailable]

Name	Port_DiscSet[NumberOfAnalogDigitalPortsAvailable]
Type	Uint32

Description	Array of port specific data, which stores input mode of port pad. Array is created only for the ports which support analog/digital functionality. If microcontroller do not have any port with analog/digital functionality the variable will not be defined.	
Verification method	The generated structure member is present in the Port_Config[_<variant>] structure. The generated structure member contains an array entry for each port which support analog/digital functionality. The function like macro 'Port_lDiscSet' is invoked at each array entry. The parameters of 'Port_lDiscSet' are the pin configuration at PortConfigSet/PortContainer/*/PortPin/*/PortPinEnableAnalogInputOnly. 'PortPinEnableAnalogInputOnly' can be configured as PORT_PIN_ANALOG_INPUT_DISABLE or PORT_PIN_ANALOG_INPUT_ENABLE.	
Example(s)	**Action**	**Generated output**
	Ports supporting analog/digital functionality : 0 Pins enabled with analog functionality: 0 4 8 12	`static const uint32 Port_DiscSet[] =` `{` `/*` `Port0 */` `Port_lDiscSet(` `PORT_PIN_ANALOG_INPUT_ENABLE,/* Pin 0 */` `PORT_PIN_ANALOG_INPUT_DISABLE,/* Pin 1 */` `PORT_PIN_ANALOG_INPUT_DISABLE,/* Pin 2 */` `PORT_PIN_ANALOG_INPUT_DISABLE,/* Pin 3 */` `PORT_PIN_ANALOG_INPUT_ENABLE,/* Pin 4 */` `PORT_PIN_ANALOG_INPUT_DISABLE,/* Pin 5 */` `PORT_PIN_ANALOG_INPUT_DISABLE,/* Pin 6 */` `PORT_PIN_ANALOG_INPUT_DISABLE,/* Pin 7 */` `PORT_PIN_ANALOG_INPUT_ENABLE,/* Pin 8 */` `PORT_PIN_ANALOG_INPUT_DISABLE,/* Pin`

		9 */ PORT_PIN_ANALOG_INPUT_DISABLE,/* Pin 10 */ PORT_PIN_ANALOG_INPUT_DISABLE,/* Pin 11 */ PORT_PIN_ANALOG_INPUT_ENABLE,/* Pin 12 */ PORT_PIN_ANALOG_INPUT_DISABLE,/* Pin 13 */ PORT_PIN_ANALOG_INPUT_DISABLE,/* Pin 14 */ PORT_PIN_ANALOG_INPUT_DISABLE /* Pin 15 */) }

1.2.56.3 Member: Port_kLVDSConfig[NumberOfAvailableLVDSPorts]

Table 89 Port_kLVDSConfig[NumberOfAvailableLVDSPorts]

Name	Port_kLVDSConfig[NumberOfAvailableLVDSPorts]	
Type	Port_n_LVDSConfigType	
Description	Array stores the LVDS pin pair configuration. If microcontroller do not have any port with LVDS functionality the variable will not be defined.	
Verification method	The generated structure member is present in the Port_Config[_<variant>] structure. The generated structure member contains an array entry for each port which support LVDS functionality. Each entry of array defined with the function like macro 'Port_lLvds' for each pin pair that support LVDS functionality. Parameters of 'Port_lLvds' are based on the configuration in PortConfigSet/PortContainer/*/PortLVDS/*. For Pin pair supporting Rx functionality values configured in PortLVDSRxEnController, PortLVDSRxPathEnable, PortLVDSRxTerminationMode, PortLVDSMode and PortLVDSPadSupply are considered. For Pin Pair supporting Tx functionality values configured in PortLVDSPadSupply, PortLVDSTxEnController, PortLVDSTxPathEnable and PortLVDSTxPowerDownPullDown are considered. Other parameters are set to default.	
Example(s)	**Action**	**Generated output**
	Port 21 pin pair 0,1 support LVDS Rx functionality and 4,5 support LVDS Tx functionality.	{ /* Port21 */ /* LPCR0 */ Port_lLvds(PORT_LVDS_PORT_CONTROLLED,

	Pin Pair 0,1 configured as PortLVDSRxEnController: PORT_CONTROLLED PortLVDSRxPathEnable: True PortLVDSRxTerminationMode: INTERNAL_TERMINATION PortLVDSMode: LVDSH PortLVDSPadSupply: V5_0 Pin Pair 4,5 configured as PortLVDSPadSupply: V3_3 PortLVDSTxEnController: HSCT_CONTROLLED PortLVDSTxPathEnable: False PortLVDSTxPowerDownPullDown: DISABLE	PORT_LVDS_ENABLE, PORT_LVDS_INTERNAL_TERMINATION, PORT_LVDS_POLY_RESISTOR_TERM, PORT_LVDS_LVDSH, PAD_SUPPLY_5_0V, PORT_LVDS_DEFAULT, PORT_LVDS_DEFAULT, PORT_LVDS_DEFAULT, PORT_LVDS_DEFAULT, PORT_LVDS_DEFAULT, PORT_LVDS_DEFAULT, PORT_LVDS_DEFAULT), /* LPCR1 */ PORT_LPCR_DEFAULT, /* LPCR2 */ Port_lLvds(PORT_LVDS_DEFAULT, PORT_LVDS_DEFAULT, PORT_LVDS_DEFAULT, PORT_LVDS_DEFAULT, PORT_LVDS_DEFAULT, PAD_SUPPLY_3_3V, PORT_LVDS_HSCT_CONTROLLED, PORT_LVDS_DISABLE, PORT_LVDS_LVDS_VDIFF_ADJ, PORT_LVDS_VOSDYN, PORT_LVDS_VOSEXT, PORT_LVDS_POWER_DOWN, PORT_LVDS_TX_PWDPD_DISABLE), /* LPCR3 */ PORT_LPCR_DEFAULT, /* LPCR4 */ PORT_LPCR_DEFAULT, /* LPCR5 */ PORT_LPCR_DEFAULT, /* LPCR6 */ PORT_LPCR_DEFAULT },

1.2.56.4 Member: Port_kPCSRConfig[NumberOfAvailablePCSRPorts]

Table 90 Port_kPCSRConfig[NumberOfAvailablePCSRPorts]

Name	Port_kPCSRConfig[NumberOfAvailablePCSRPorts]
Type	Port_n_PCSRConfigType
Description	Array stores the configuration of the pins which support PCSR functionality.
Verification method	The generated structure member is present in the Port_Config[_<variant>] structure. The generated structure member contains an array entry for each port which support

	PCSR functionality. Parameters of 'Port_lPcsr' are the status of port pin configured in PortConfigSet/PortContainer/*/PortPin/*/ PortPinControllerSelect. For the pin supporting PCSR functionality PortPinControllerSelect can be configured as ENABLE or DISABLE. And the pin that do not support will be configured to DEFAULT.	
Example(s)	**Action**	**Generated output**
	Port 0 pin 10 and 11 support PCSR functionality. Control selection is enabled for pin 10 and disabled for pin 11	`static const Port_n_PCSRConfigType Port_kPCSRConfig[] =` `{` `/* Port0 */` `Port_lPcsr(` `PORT_PCSR_DEFAULT, /*Pin0*/` `PORT_PCSR_DEFAULT, /*Pin1*/` `PORT_PCSR_DEFAULT, /*Pin2*/` `PORT_PCSR_DEFAULT, /*Pin3*/` `PORT_PCSR_DEFAULT, /*Pin4*/` `PORT_PCSR_DEFAULT, /*Pin5*/` `PORT_PCSR_DEFAULT, /*Pin6*/` `PORT_PCSR_DEFAULT, /*Pin7*/` `PORT_PCSR_DEFAULT, /*Pin8*/` `PORT_PCSR_DEFAULT, /*Pin9*/` `PORT_PCSR_ENABLE, /*Pin10*/` `PORT_PCSR_DISABLE, /*Pin11*/` `PORT_PCSR_DEFAULT, /*Pin12*/` `PORT_PCSR_DEFAULT, /*Pin13*/` `PORT_PCSR_DEFAULT, /*Pin14*/` `PORT_PCSR_DEFAULT /*Pin15*/` `)` `}`

1.2.56.5 Member: Port_kPinSupportedModes[NumberOfPortsAvailable]

Table 91 Port_kPinSupportedModes[NumberOfPortsAvailable]

Name	Port_kPinSupportedModes[NumberOfPortsAvailable]	
Type	Port_n_ModeType	
Description	Array stores the modes supported by the pins of the available ports.	
Verification method	The generated structure member is present in the Port_Config[_<variant>] structure. The generated structure member contains an array entry for each port available in microcontroller. Modes of the port pins are derived from the configured values in PortConfigSet/PortContainer/*/PortPin/PortPinMode/*. The port mode is generated by doing 'OR' operation between the modes selected.	
Example(s)	**Action**	**Generated output**

	Ports available: 0 Modes supported in Pin0:GPIO Pin1:ALT1 Pin2:ALT2 Pin3:ALT3 Pin4:ALT4 Pin5:ALT5 Pin6:ALT6 Pin7:ALT7 Pin8:GPIO, ALT1 Pin9:ALT2, ALT3 Pin10:ALT4, ALT5 Pin11:ALT6, ALT7 Pin12:GPIO, ALT1,ALT2 Pin13:ALT3,ALT4,ALT5 Pin14:GPIO,ALT2,ALT4 Pin15:ALL	static const Port_n_ModeType Port_kPinSupportedModes[] = { /* Port0 */ { { (uint8)(0x01U),/*Pin0*/ (uint8)(0x02U),/*Pin1*/ (uint8)(0x04U),/*Pin2*/ (uint8)(0x08U),/*Pin3*/ (uint8)(0x10U),/*Pin4*/ (uint8)(0x20U),/*Pin5*/ (uint8)(0x40U),/*Pin6*/ (uint8)(0x80U),/*Pin7*/ (uint8)(0x03U),/*Pin8*/ (uint8)(0x0cU),/*Pin9*/ (uint8)(0x30U),/*Pin10*/ (uint8)(0xc0U),/*Pin11*/ (uint8)(0x07U),/*Pin12*/ (uint8)(0x31U),/*Pin13*/ (uint8)(0x15U),/*Pin14*/ (uint8)(0xffU) /*Pin15*/ } }

1.3 File: Port[_<variant>]_PBcfg.h

The generated header file contains the declaration of the root configuration structure. Post-build time configuration mechanism allows configurable functionality of PORT driver that is deployed as object code. The file is generated in ‘inc’ folder.

1.3.1 Structure: Port_Config[_<variant>]

Table 92 Port_Config[_<varaint>]

Name	Port_Config[_<variant>]
Type	Port_ConfigType
Description	Declaration of root configuration structure of PORT driver which will be used during initialization.
Verification method	The generated structure is present in Port [_<variant>] _PBcfg.h file. The <variant> indicates the name of the post-build variant. For a variant-aware configuration the structure name is appended with the variant name. For variant-unaware

	configuration <variant> is ignored.	
Example(s)	**Action**	**Generated output**
	Configure PORT driver and generate(variant-unaware)	`extern const Port_ConfigType Port_Config;`
	Configure PORT driver and generate (variant-aware. Variant name is 'Petrol')	`extern const Port_ConfigType Port_Config_Petrol;`

Revision history

Major changes since the last revision

Date	Version	Description
24-03-2021	5.0	Document Released
23-03-2021	4.1	Description of Macro: PORT14_PDISC_MASK updated
14-10-2020	4.0	Document Released
13-10-2020	3.1	Port driver chapter moved from MC-ISAR_TC3xx_Config_Verification_Manual_BASIC.pdf to this document
19-07-2019	3.0	Document reviewed and released
19-07-2019	2.1	Added RGMII_INPUT macro.
26-02-2019	1.10.0_2.0	Added Pbcfg.h file
25-02-2019	1.10.0_1.0	Initial Release

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

Edition 2021-03-24
Published by

Infineon Technologies AG

81726 Munich, Germany

Do you have a question about this document?

Email: erratum@infineon.com

Document reference
Doc_Number

IMPORTANT NOTICE
The information given in this document shall in no event be regarded as a guarantee of conditions or characteristics (“Beschaffenheitsgarantie”) .

With respect to any examples, hints or any typical values stated herein and/or any information regarding the application of the product, Infineon Technologies hereby disclaims any and all warranties and liabilities of any kind, including without limitation warranties of non-infringement of intellectual property rights of any third party.

In addition, any information given in this document is subject to customer’s compliance with its obligations stated in this document and any applicable legal requirements, norms and standards concerning customer’s products and any use of the product of Infineon Technologies in customer’s applications.

The data contained in this document is exclusively intended for technically trained staff. It is the responsibility of customer’s technical departments to evaluate the suitability of the product for the intended application and the completeness of the product information given in this document with respect to such application.

For further information on the product, technology, delivery terms and conditions and prices please contact your nearest Infineon Technologies office (**www.infineon.com**).

WARNINGS
Due to technical requirements products may contain dangerous substances. For information on the types in question please contact your nearest Infineon Technologies office.

Except as otherwise explicitly approved by Infineon Technologies in a written document signed by authorized representatives of Infineon Technologies, Infineon Technologies’ products may not be used in any applications where a failure of the product or any consequences of the use thereof can reasonably be expected to result in personal injury.