
Integration Manual

for MPC574XG PORT Driver

Document Number: IM35PORTASR4.2 Rev002 R1.0.0
Rev. 1.0





Contents

Section number	Title	Page
Chapter 1		
Revision History		
Chapter 2		
Introduction		
2.1	Supported Derivatives.....	7
2.2	Overview.....	8
2.3	About this Manual.....	8
2.4	Acronyms and Definitions.....	9
2.5	Reference List.....	9
Chapter 3		
Building the Driver		
3.1	Build Options.....	11
3.1.1	DIAB Compiler/Linker/Assembler Options.....	11
3.1.2	GHS Compiler/Linker/Assembler Options.....	13
3.2	Files required for Compilation.....	15
3.3	Setting up the Plug-ins.....	17
Chapter 4		
Function calls to module		
4.1	Function Calls during Start-up.....	19
4.2	Function Calls during Shutdown.....	19
4.3	Function Calls during Wake-up.....	19
Chapter 5		
Module requirements		
5.1	Exclusive areas to be defined in BSW scheduler.....	21
5.2	Peripheral Hardware Requirements.....	22
5.3	ISR to configure within OS – dependencies.....	22
5.4	ISR Macro.....	22
5.5	Other AUTOSAR modules - dependencies.....	22

Section number	Title	Page
5.6	Data cache restriction.....	23
<p style="text-align: center;">Chapter 6 Main API Requirements</p>		
6.1	Main functions calls within BSW scheduler.....	25
6.2	API Requirements.....	25
6.3	Calls to Notification Functions, Callbacks, Callouts.....	25
<p style="text-align: center;">Chapter 7 Memory Allocation</p>		
7.1	Sections to be defined in Port_MemMap.h.....	27
7.2	Linker command file.....	28
<p style="text-align: center;">Chapter 8 Configuration parameters considerations</p>		
8.1	Configuration Parameters.....	29
<p style="text-align: center;">Chapter 9 Integration Steps</p>		
<p style="text-align: center;">Chapter 10 External Assumptions for PORT driver</p>		

Chapter 1

Revision History

Table 1-1. Revision History

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



Chapter 2

Introduction

This integration manual describes the integration requirements for Port Driver for MPC574XG microcontrollers.

2.1 Supported Derivatives

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

Table 2-1. MPC574XG Derivatives

NXP Semiconductor	MPC5748G_LQFP176, MPC5748G_MAPBGA256, MPC5748G_MAPBGA324, MPC5747G_LQFP176, MPC5747G_MAPBGA256, MPC5747G_MAPBGA324, MPC5746G_LQFP176, MPC5746G_MAPBGA256, MPC5746G_MAPBGA324, MPC5748C_LQFP176, MPC5748C_MAPBGA256, MPC5748C_MAPBGA324, MPC5747C_LQFP176, MPC5747C_MAPBGA256, MPC5747C_MAPBGA324, MPC5746C_LQFP176, MPC5746C_MAPBGA256, MPC5746C_MAPBGA324, MPC5746C_MAPBGA100, MPC5745C_LQFP176, MPC5745C_MAPBGA256, MPC5745C_MAPBGA100, MPC5744C_LQFP176, MPC5744C_MAPBGA256, MPC5744C_MAPBGA100, MPC5746B_LQFP176, MPC5746B_MAPBGA256, MPC5746B_MAPBGA100, MPC5744B_LQFP176, MPC5744B_MAPBGA256,
-------------------	--

Table 2-1. MPC574XG Derivatives

	MPC5744B_MAPBGA100, MPC5745B_LQFP176, MPC5745B_MAPBGA256, MPC5745B_MAPBGA100
--	---

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

2.2 Overview

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

AUTOSAR

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

2.3 About this Manual

This Technical Reference employs the following typographical conventions:

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

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

Notes and warnings are shown as below:

Note

This is a note.

2.4 Acronyms and Definitions

Table 2-2. Acronyms and Definitions

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

2.5 Reference List

Table 2-3. Reference List

#	Title	Version
1	AUTOSAR 4.2 Rev0002Port Driver Software Specification Document.	V3.2.0
2	MPC5748G Reference Manual	Rev. 5, 12/2016

Table continues on the next page...

Table 2-3. Reference List (continued)

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

Chapter 3

Building the Driver

This section describes the source files and various compilers, linker options used for building the Autosar Port driver for NXP SemiconductorMPC574XG . It also explains the EB Tresos Studio plugin setup procedure.

3.1 Build Options

The Port driver files are compiled using

- Windriver DIAB DIAB_5_9_6_2
- Green Hills Multi 7.1.4 / Compiler 2015.1.6

The compiler, linker flags used for building the driver are explained below:

Note

The TS_T2D35M10I0R0 plugin name is composed as follow:

TS_T = Target_Id

D = Derivative_Id

M = SW_Version_Major

I = SW_Version_Minor

R = Revision

(i.e. Target_Id = 2 identifies PA architecture and Derivative_Id = 35 identifies the MPC574XG)

3.1.1 DIAB Compiler/Linker/Assembler Options

Table 3-1. Compiler Options

Option	Description
-tPPCE200Z4204N3VEN:simple	Sets target processor to PPCE200Z4204N3VEN, generates ELF using EABI conventions, No floating point support (minimizes the required runtime), selects simple environment settings for Startup Module and Libraries
-tPPCE200Z210N3VEN:simple	Sets target processor to PPCE200Z210N3VEN, generates ELF using EABI conventions, No floating point support (minimizes the required runtime), selects simple environment settings for Startup Module and Libraries
-Xdialect-ansi	Follow the ANSI C standard with some additions
-XO	Enables extra optimizations to produce highly optimized code
-g3	Generate symbolic debugger information and do all optimizations.
-Xsize-opt	Optimize for size rather than speed when there is a choice
-Xsmall-data=0	Set Size Limit for 'small data' Variables to zero.
-Xsmall-const=0	Set Size Limit for "small const" Variables to zero.
-Xaddr-sconst=0x11	Specify addressing for constant static and global variables with size less than or equal to -Xsmall-const to far-absolute.
-Xaddr-sdata=0x11	Specify addressing for non-constant static and global variables with size less than or equal to -Xsmall-data in size to far-absolute.
-Xno-common	Disable use of the 'COMMON' feature so that the compiler or assembler will allocate each uninitialized public variable in the .bss section for the module defining it, and the linker will require exactly one definition of each public variable
-Xnested-interrupts	Allow nested interrupts
-Xdebug-dwarf2	Generate symbolic debug information in dwarf2 format
-Xdebug-local-all	Force generation of type information for all local variables
-Xdebug-local-cie	Create common information entry per module
-Xdebug-struct-all	Force generation of type information for all typedefs, struct, union and class types
-Xforce-declarations	Generates warnings if a function is used without a previous declaration
-ee1481	Generate an error when the function was used before it has been declared
-Xmacro-undefined-warn	Generates a warning when an undefined macro name occurs in a #if preprocessor directive
-Xlink-time-lint	Enable the checking of object and function declarations across compilation units, as well as the consistency of compiler options used to compile source files
-W:as,-l	Pass the option '-l' (lower case letter L) to the assembler to get an assembler listing file
-Wa,-Xisa-vle	Instruct the assembler to expect and assemble VLE (Variable Length Encoding) instructions rather than BookE instructions.
-DAUTOSAR_OS_NOT_USED	-D defines a preprocessor symbol and optionally can set it to a value. AUTOSAR_OS_NOT_USED: By default in the package, the drivers are compiled to be used without Autosar OS. If the drivers are used with Autosar OS, the compiler option '-DAUTOSAR_OS_NOT_USED' must be removed from project options
-DUSE_SW_VECTOR_MODE	-D defines a preprocessor symbol and optionally can set it to a value. USE_SW_VECTOR_MODE: By default in the package, drivers are compiled to be used with interrupt controller configured to be in hardware vector mode. In case of AUTOSAR_OS_NOT_USED, the compiler option "-DUSE_SW_VECTOR_MODE" must be added to the list of compiler options to be used with interrupt controller configured to be in software vector mode.

Table continues on the next page...

Table 3-1. Compiler Options (continued)

Option	Description
-DDIAB	-D defines a preprocessor symbol and optionally can set it to a value. This one defines the DIAB preprocessor symbol.
-DDISABLE_MCAL_INTERMODULE_ASRCHECK	-D defines a preprocessor symbol to disable the inter-module version check for AR_RELEASE versions. DISABLE_MCAL_INTERMODULE_ASRCHECK: By default in the package, drivers are compiled to perform the inter-module version check as per Autosar BSW004. When the inter-module version check needs to be disabled then the DISABLE_MCAL_INTERMODULE_ASRCHECK global define must be added to the list of compiler options.
-c	Stop after assembly, produce object file.

Table 3-2. Assembler Options

Option	Description
-tPPCE200Z4204N3VEN:simple	Sets target processor to PPCE200Z4204N3VEN, generates ELF using EABI conventions, No floating point support (minimizes the required runtime), selects simple environment settings for Startup Module and Libraries
-tPPCE200Z210N3VEN:simple	Sets target processor to PPCE200Z210N3VEN, generates ELF using EABI conventions, No floating point support (minimizes the required runtime), selects simple environment settings for Startup Module and Libraries
-g	Dump the symbols in the global symbol table in each archive file.
-Xisa-vle	Expect and assemble VLE (Variable Length Encoding) instructions rather than Book E instructions. The default code section is named .text_vle instead of .text, and the default code section fill "character" is set to 0x44444444 instead of 0. The .text_vle code section will have ELF section header flags marking it as VLE code, not Book E code.
-Xasm-debug-on	Generate debug line and file information
-Xdebug-dwarf2	Generate symbolic debug information in dwarf2 format
-Xsemi-is-newline	Treat the semicolon (;) as a statement separator instead of a comment character.

Table 3-3. Linker Options

Option	Description
-tPPCE200Z4204N3VEN:simple	Sets target processor to tPPCE200Z4204N3VEN, generates ELF using EABI conventions, No floating point support (minimizes the required runtime), selects simple environment settings for Startup Module and Libraries
-tPPCE200Z210N3VEN:simple	Sets target processor to tPPCE200Z210N3VEN, generates ELF using EABI conventions, No floating point support (minimizes the required runtime), selects simple environment settings for Startup Module and Libraries
-Xelf	Generates ELF object format for output file
-m6	Generates a detailed link map and cross reference table
-Xlink-time-lint	Enable the checking of object and function declarations across compilation units, as well as the consistency of compiler options used to compile source files

3.1.2 GHS Compiler/Linker/Assembler Options

Table 3-4. Compiler Options

Option	Description
-cpu=ppc5748gz4204	Selects target processor: ppc5748gz4204
-cpu=ppc5748gz210	Selects target processor: ppc5748gz210
-ansi	Specifies ANSI C with extensions. This mode extends the ANSI X3.159-1989 standard with certain useful and compatible constructs.
-noSPE	Disables the use of SPE and vector floating point instructions by the compiler.
-Ospace	Optimize for size.
-sda=0	Enables the Small Data Area optimization with a threshold of 0.
-vle	Enables VLE code generation
-dual_debug	Enables the generation of DWARF, COFF, or BSD debugging information in the object file
-G	Generates source level debugging information and allows procedure call from debugger's command line.
--no_exceptions	Disables support for exception handling
-Wundef	Generates warnings for undefined symbols in preprocessor expressions
-Wimplicit-int	Issues a warning if the return type of a function is not declared before it is called
-Wshadow	Issues a warning if the declaration of a local variable shadows the declaration of a variable of the same name declared at the global scope, or at an outer scope
-Wtrigraphs	Issues a warning for any use of trigraphs
--prototype_errors	Generates errors when functions referenced or called have no prototype
--incorrect_pragma_warnings	Valid #pragma directives with wrong syntax are treated as warnings
-noslashcomment	C++ like comments will generate a compilation error
-preprocess_assembly_files	Preprocesses assembly files
-nostartfile	Do not use Start files
--short_enum	Store enumerations in the smallest possible type
--diag_error 223	Sets the specified compiler diagnostic messages to the level of error
-DAUTOSAR_OS_NOT_USED	-D defines a preprocessor symbol and optionally can set it to a value. AUTOSAR_OS_NOT_USED: By default in the package, the drivers are compiled to be used without Autosar OS. If the drivers are used with Autosar OS, the compiler option '-DAUTOSAR_OS_NOT_USED' must be removed from project options
-DUSE_SW_VECTOR_MODE	-D defines a preprocessor symbol and optionally can set it to a value. USE_SW_VECTOR_MODE: By default in the package, drivers are compiled to be used with interrupt controller configured to be in hardware vector mode. In case of AUTOSAR_OS_NOT_USED, the compiler option "-DUSE_SW_VECTOR_MODE" must be added to the list of compiler options to be used with interrupt controller configured to be in software vector mode.
-DDISABLE_MCAL_INTERMODULE_ASR_CHECK	-D defines a preprocessor symbol to disable the inter-module version check for AR_RELEASE versions. DISABLE_MCAL_INTERMODULE_ASR_CHECK: By default in the package, drivers are compiled to perform the inter-module version check as per Autosar BSW004. When the inter-module version check needs to be disabled then the DISABLE_MCAL_INTERMODULE_ASR_CHECK global define must be added to the list of compiler options.
-DGHS	-D defines a preprocessor symbol and optionally can set it to a value. This one defines the GHS preprocessor symbol.
-c	Produces an object file (called input-file.o) for each source file.

Table 3-5. Assembler Options

Option	Description
-cpu=ppc5748gz4204	Selects target processor: ppc5748gz4204
-cpu=ppc5748gz210	Selects target processor: ppc5748gz210
-G	Generates source level debugging information and allows procedure call from debugger's command line.
-list	Creates a listing by using the name of the object file with the .lst extension

Table 3-6. Linker Options

Option	Description
-cpu=ppc5748gz4204	Selects target processor: ppc5748gz4204
-cpu=ppc5748gz210	Selects target processor: ppc5748gz210
-nostartfiles	Do not use Start files.
-vle	Enables VLE code generation
--nocpp	Do not Generate Constructors/Destructors
-Mn	sort numerically the MAP file
-delete	The -delete option instructs the linker to remove functions that are not referenced in the final executable.
-ignore_debug_references	Ignores relocations from DWARF debug sections when using -delete. DWARF debug information will contain references to deleted functions that may break some third-party debuggers.
-keepmap	keeps the MAP file in case of link error

3.2 Files required for Compilation

This section describes the include files required to compile, assemble (if assembler code) and link the Port driver for MPC574XG microcontrollers.

To avoid integration of incompatible files, all the include files from other modules shall have the same AR_RELEASE_MAJOR_VERSION and AR_RELEASE_MINOR_VERSION, i.e. only files with the same AUTOSAR major and minor versions can be compiled.

Port Files

- ..\Port_TS_T2D35M10I0R0\src\Port.c
- ..\Port_TS_T2D35M10I0R0\src\Port_Ipw.c
- ..\Port_TS_T2D35M10I0R0\src\Port_Siul2.c
- ..\Port_TS_T2D35M10I0R0\include\Port.h
- ..\Port_TS_T2D35M10I0R0\include\Port_EnvCfg.h
- ..\Port_TS_T2D35M10I0R0\include\Port_Ipw.h

Files required for Compilation

- ..\Port_TS_T2D35M10I0R0\include\Port_Siul2.h
- ..\Port_TS_T2D35M10I0R0\include\Reg_eSys_Siul2.h
- ..\Port_TS_T2D35M10I0R0\include\Siul2_IpVersion.h
- ..\Port_TS_T2D35M10I0R0\include\Port_Siul2_Types.h

Port Generated Files

- Port_Cfg.c - This file should be generated by the user using a configuration tool for compilation.
- Port_PBCfg_[VariantName].c - This file should be generated by the user using a configuration tool for compilation. The file contains the definition of the init pointer for the respective variant.
- Port_Cfg.h - This file should be generated by the user using a configuration tool for compilation.

Files from Base common folder

- ..\Base_TS_T2D35M10I0R0\include\Compiler.h
- ..\Base_TS_T2D35M10I0R0\include\Compiler_Cfg.h
- ..\Base_TS_T2D35M10I0R0\include\CompilerDefinition.h
- ..\Base_TS_T2D35M10I0R0\include\ComStack_Types.h
- ..\Base_TS_T2D35M10I0R0\include\Dio_MemMap.h
- ..\Base_TS_T2D35M10I0R0\include\Mcal.h
- ..\Base_TS_T2D35M10I0R0\include\Platform_Types.h
- ..\Base_TS_T2D35M10I0R0\include\Reg_eSys.h
- ..\Base_TS_T2D35M10I0R0\include\Reg_Macros.h
- ..\Base_TS_T2D35M10I0R0\include\Reg_LockMacros.h
- ..\Base_TS_T2D35M10I0R0\include\SilRegMacros.h
- ..\Base_TS_T2D35M10I0R0\include\Soc_Ips.h
- ..\Base_TS_T2D35M10I0R0\include\Std_Types.h
- ..\Base_TS_T2D35M10I0R0\include\StdRegMacros.h

Files from Det folder:

- ..\Det_TS_T2D35M10I0R0\include\Det.h
- ..\Det_TS_T2D35M10I0R0\src\Det.c

Files from Rte folder:

- ..\Rte_TS_T2D35M10I0R0\include\SchM_Dio.h
- ..\Rte_TS_T2D35M10I0R0\src\SchM_Dio.c

Note:

<plugin_name>: TS_T<2>D<35>M<SW_Version_Major>I<SW_Version_Minor>R0

(i.e. Target_Id = 2 identifies PowerPC architecture and Derivative_Id = 35 identifies the MPC574XG)

3.3 Setting up the Plug-ins

The Port driver was designed to be configured by using the EB Tresos Studio (version EB tresos Studio 21.0.0 b160607-0933 or later.)

Location of various files inside the PORT module folder:

- VSMD (Vendor Specific Module Definition) file in EB tresos Studio XDM format:
 - ..\Port_TS_T2D35M10I0R0\config\Port.xdm
- VSMD (Vendor Specific Module Definition) file(s) in AUTOSAR compliant EPD format:
 - ..\Port_TS_T2D35M10I0R0\autosar\Port_mpc5744b_lqfp176.epd
 - ..\Port_TS_T2D35M10I0R0\autosar\Port_mpc5744b_mapbga100.epd
 - ..\Port_TS_T2D35M10I0R0\autosar\Port_mpc5744b_mapbga256.epd
 - ..\Port_TS_T2D35M10I0R0\autosar\Port_mpc5744c_lqfp176.epd
 - ..\Port_TS_T2D35M10I0R0\autosar\Port_mpc5744c_mapbga100.epd
 - ..\Port_TS_T2D35M10I0R0\autosar\Port_mpc5744c_mapbga256.epd
 - ..\Port_TS_T2D35M10I0R0\autosar\Port_mpc5745b_lqfp176.epd
 - ..\Port_TS_T2D35M10I0R0\autosar\Port_mpc5745b_mapbga100.epd
 - ..\Port_TS_T2D35M10I0R0\autosar\Port_mpc5745b_mapbga256.epd
 - ..\Port_TS_T2D35M10I0R0\autosar\Port_mpc5745c_lqfp176.epd
 - ..\Port_TS_T2D35M10I0R0\autosar\Port_mpc5745c_mapbga100.epd
 - ..\Port_TS_T2D35M10I0R0\autosar\Port_mpc5745c_mapbga256.epd
 - ..\Port_TS_T2D35M10I0R0\autosar\Port_mpc5746b_lqfp176.epd
 - ..\Port_TS_T2D35M10I0R0\autosar\Port_mpc5746b_mapbga100.epd
 - ..\Port_TS_T2D35M10I0R0\autosar\Port_mpc5746b_mapbga256.epd
 - ..\Port_TS_T2D35M10I0R0\autosar\Port_mpc5746c_lqfp176.epd
 - ..\Port_TS_T2D35M10I0R0\autosar\Port_mpc5746c_mapbga100.epd
 - ..\Port_TS_T2D35M10I0R0\autosar\Port_mpc5746c_mapbga256.epd
 - ..\Port_TS_T2D35M10I0R0\autosar\Port_mpc5746c_mapbga324.epd
 - ..\Port_TS_T2D35M10I0R0\autosar\Port_mpc5746g_lqfp176.epd
 - ..\Port_TS_T2D35M10I0R0\autosar\Port_mpc5746g_mapbga256.epd
 - ..\Port_TS_T2D35M10I0R0\autosar\Port_mpc5746g_mapbga324.epd
 - ..\Port_TS_T2D35M10I0R0\autosar\Port_mpc5747c_lqfp176.epd
 - ..\Port_TS_T2D35M10I0R0\autosar\Port_mpc5747c_mapbga256.epd
 - ..\Port_TS_T2D35M10I0R0\autosar\Port_mpc5747c_mapbga324.epd
 - ..\Port_TS_T2D35M10I0R0\autosar\Port_mpc5747g_lqfp176.epd
 - ..\Port_TS_T2D35M10I0R0\autosar\Port_mpc5747g_mapbga256.epd
 - ..\Port_TS_T2D35M10I0R0\autosar\Port_mpc5747g_mapbga324.epd
 - ..\Port_TS_T2D35M10I0R0\autosar\Port_mpc5748c_lqfp176.epd

- ..\Port_TS_T2D35M10I0R0\autosar\Port_mpc5748c_mapbga256.epd
 - ..\Port_TS_T2D35M10I0R0\autosar\Port_mpc5748c_mapbga324.epd
 - ..\Port_TS_T2D35M10I0R0\autosar\Port_mpc5748g_lqfp176.epd
 - ..\Port_TS_T2D35M10I0R0\autosar\Port_mpc5748g_mapbga256.epd
 - ..\Port_TS_T2D35M10I0R0\autosar\Port_mpc5748g_mapbga324.epd
- Code Generation Templates for variant aware parameters:
 - ..\Port_TS_T2D35M10I0R0\generate_PB\src\Port_PBcfg.c

Steps to generate the configuration:

1. Copy the module folders Port_TS_T2D35M10I0R0 , Base_TS_T2D35M10I0R0 , Resource_TS_T2D35M10I0R0 , Det_TS_T2D35M10I0R0 , Rte_TS_T2D35M10I0R0 , Ecuc_TS_T2D35M10I0R0 into the Tresos plugins folder.
2. Set the desired Tresos Output location folder for the generated sources and header files.
3. Use the EB tresos Studio GUI to modify ECU configuration parameters values.
4. Generate the configuration files.



Chapter 4

Function calls to module

4.1 Function Calls during Start-up

None.

4.2 Function Calls during Shutdown

None.

4.3 Function Calls during Wake-up

None.

Chapter 5

Module requirements

5.1 Exclusive areas to be defined in BSW scheduler

In the current implementation, PORT is using the services of Schedule Manager (SchM) for entering and exiting the critical regions, to preserve a resource. SchM implementation is done by the integrators of the MCAL using OS or non-OS services. For testing the PORT, stubs are used for SchM. The following critical regions are used in the PORT driver:

PORT_EXCLUSIVE_AREA_00 To protect the SIUL2_MSCR_ADDR32(pinPad) resource from read/modify/write operation. It is used in the function Port_Siul2_SetPinMode, called from HLD Port_SetPinMode function;

PORT_EXCLUSIVE_AREA_01 To protect the SIUL2_MSCR_ADDR32(pinPad) resource from register operation: REG_BIT_SET32, REG_BIT_CLEAR32 and REG_RMW32. It is used in the function Port_Siul2_RefreshPortDirection, called from HLD Port_RefreshPortDirection function;

PORT_EXCLUSIVE_AREA_02 To protect the Port_GPIODirChangeability[] array during the read/modify/write action. It is used in the function Port_Siul2_SetGpioDirChangeability, called from HLD Port_SetPinMode function;

PORT_EXCLUSIVE_AREA_03 To protect the SIUL2_MSCR_ADDR32(pinPad) resource from register operation: REG_BIT_SET32, REG_BIT_CLEAR32 and REG_RMW32. It is used in the function Port_Siul2_SetPinDirection, called from HLD Port_SetPinDirection function;

Critical Region Exclusive Matrix

Below is the table depicting the exclusivity between different critical region IDs from the PORT driver. If there is an “X” in the table, it means that those 2 critical regions cannot interrupt each other.

Table 5-1. Exclusive Areas

	PORT_EA_00	PORT_EA_01	PORT_EA_02	PORT_EA_03
PORT_EA_00	X	X		X
PORT_EA_01	X			X
PORT_EA_02			X	
PORT_EA_03	X	X		X

Note

- PORT_EA_xx means PORT_EXCLUSIVE_AREA_xx

5.2 Peripheral Hardware Requirements

The PORT driver uses MPC574XG's peripheral: SIUL2.

5.3 ISR to configure within OS – dependencies

None.

5.4 ISR Macro

None.

5.5 Other AUTOSAR modules - dependencies

- **Det** This module is necessary for enabling Development error detection. The API function used is Det_ReportError(). The activation/deactivation of Development error detection is configurable using 'CanDevErrorDetect' configuration parameter.
- **Rte** This module is necessary for the infrastructure services that enable communication to occur between AUTOSAR software-components as well as acting as the means by which AUTOSAR software-components access basic software modules including the OS and communication service.

- **Resource** This module is required to select processor derivative. Current Port driver has support for the following derivatives, everyone having attached a Resource file: MPC5748G_LQFP176, MPC5748G_MAPBGA256, MPC5748G_MAPBGA324, MPC5747G_LQFP176, MPC5747G_MAPBGA256, MPC5747G_MAPBGA324, MPC5746G_LQFP176, MPC5746G_MAPBGA256, MPC5746G_MAPBGA324, MPC5748C_LQFP176, MPC5748C_MAPBGA256, MPC5748C_MAPBGA324, MPC5747C_LQFP176, MPC5747C_MAPBGA256, MPC5747C_MAPBGA324, MPC5746C_LQFP176, MPC5746C_MAPBGA256, MPC5746C_MAPBGA324, MPC5746C_MAPBGA100, MPC5745C_LQFP176, MPC5745C_MAPBGA256, MPC5745C_MAPBGA100, MPC5744C_LQFP176, MPC5744C_MAPBGA256, MPC5744C_MAPBGA100, MPC5746B_LQFP176, MPC5746B_MAPBGA256, MPC5746B_MAPBGA100, MPC5744B_LQFP176, MPC5744B_MAPBGA256, MPC5744B_MAPBGA100, MPC5745B_LQFP176, MPC5745B_MAPBGA256, MPC5745B_MAPBGA100 .
- **Ecuc** This module is required for configuring the variant handling in Tresos.
- **Mcu** The Microcontroller Unit Driver (MCU Driver) is primarily responsible for initializing and controlling the chips internal clock sources and clock prescalers. The clock frequency may affect the Trigger frequency, Conversion time and Sampling time.
- **Base** This module is necessary for a reference to the Wakeup source for this controller as defined in the ECU State Manager.

5.6 Data cache restriction

None



Chapter 6

Main API Requirements

6.1 Main functions calls within BSW scheduler

None.

6.2 API Requirements

None.

6.3 Calls to Notification Functions, Callbacks, Callouts

None.

Chapter 7

Memory Allocation

7.1 Sections to be defined in Port_MemMap.h

Table 7-1. MemMap sections present in the Port driver code

Section name	Section type	Description
PORT_START_SEC_CODE	Code	Start of Memory Section for Code
PORT_STOP_SEC_CODE	Code	End of Memory Section for Code
PORT_START_SEC_CONFIG_DATA_UNSPECIFIED	Configuration Data	Start of Memory Section for Config Data
PORT_STOP_SEC_CONFIG_DATA_UNSPECIFIED	Configuration Data	End of Memory Section for Config Data
PORT_START_SEC_CONFIG_DATA_16	Configuration Data	Start of Memory Section for Config Data
PORT_STOP_SEC_CONFIG_DATA_16	Configuration Data	End of Memory Section for Config Data
PORT_START_SEC_VAR_INIT_UNSPECIFIED	Variables	Used for variables, structures, arrays, when the SIZE (alignment) does not fit the criteria of 8,16 or 32 bit. These variables are initialized with values after every reset.
PORT_STOP_SEC_VAR_INIT_UNSPECIFIED	Variables	End of above section.
PORT_START_SEC_VAR_NO_INIT_16	Variables	Used for variables and constants which have to be aligned to 16 bit. These variables are never cleared and never initialized by start-up code.
PORT_STOP_SEC_VAR_NO_INIT_16	Variables	End of above section.
PORT_START_SEC_CONST_UNSPECIFIED	Constant data	The parameters that are not variant aware shall be stored in memory section for constants.
PORT_STOP_SEC_CONST_UNSPECIFIED	Constant data	End of above section.

Table continues on the next page...

Table 7-1. MemMap sections present in the Port driver code (continued)

PORT_START_SEC_CONST_16	Constant data	The parameters that are not variant aware shall be stored in memory section for constants.
PORT_STOP_SEC_CONST_16	Constant data	End of above section.

7.2 Linker command file

Memory shall be allocated for every section defined in Port_MemMap.h

Chapter 8

Configuration parameters considerations

Configuration parameter class for Autosar Port driver fall into the following variants as defined below:

8.1 Configuration Parameters

Specifies whether the configuration parameter shall be of configuration class Post Build

Table 8-1. Configuration Parameters

Configuration Container	Configuration Parameters	Configuration Variant	Current Implementation
PortGeneral	PortDevErrorDetect	Pre Compile Parameter for all configuration variants	Pre Compile
	PortSetPinDirectionApi	Pre Compile Parameter for all configuration variants	Pre Compile
	PortSetPinModeApi	Pre Compile Parameter for all configuration variants	Pre Compile
	PortVersionInfoApi	Pre Compile Parameter for all configuration variants	Pre Compile
	PortSetPinModeDoesNotTouchGpioLevel	VariantPC or VariantPB	Pre Compile
	PortEnableUserModeSupport	VariantPC or VariantPB	Pre Compile
UnUsedPortPin	PortPinWpe	VariantPC or VariantPB	Pre Compile
	PortPinWps	VariantPC or VariantPB	Pre Compile
	PortPinOde	VariantPC or VariantPB	Pre Compile
	PortPinDirection	VariantPC or VariantPB	Pre Compile
	PortPinLevelValue	VariantPC or VariantPB	Pre Compile
PortContainer	PortNumberOfPortPins	VariantPC or VariantPB	Pre Compile
	PortPinWpe	VariantPC or VariantPB	Pre Compile
	PortPinWps	VariantPC or VariantPB	Pre Compile
	PortPinOde	VariantPC or VariantPB	Pre Compile
	PortPinSafeMode	VariantPC or VariantPB	Pre Compile
	PortPinWithReadBack	VariantPC or VariantPB	Pre Compile

Table continues on the next page...

Table 8-1. Configuration Parameters (continued)

Configuration Container	Configuration Parameters	Configuration Variant	Current Implementation
	PortPinHysteresisControl	VariantPC or VariantPB	Pre Compile
	PortPinDirectionChangeable	VariantPC or VariantPB	Pre Compile
	PortPinModeChangeable	VariantPC or VariantPB	Pre Compile
	PortPinId	VariantPC or VariantPB	Pre Compile
	PortPinPcr	VariantPC or VariantPB	Pre Compile
	PortPinDirection	VariantPC or VariantPB	Pre Compile
	PortPinInitialMode	VariantPC or VariantPB	Pre Compile
	PortPinMode	VariantPC or VariantPB	Pre Compile
	PortPinLevelValue	VariantPC or VariantPB	Pre Compile
	PortPinOutputSlewRate	VariantPC or VariantPB	Pre Compile
CommonPublishedInformation	ArReleaseMajorVersion	VariantPC or VariantPB	Pre Compile
	ArReleaseMinorVersion	VariantPC or VariantPB	Pre Compile
	ArReleaseRevisionVersion	VariantPC or VariantPB	Pre Compile
	ModuleId	VariantPC or VariantPB	Pre Compile
	SwMajorVersion	VariantPC or VariantPB	Pre Compile
	SwMinorVersion	VariantPC or VariantPB	Pre Compile
	SwPatchVersion	VariantPC or VariantPB	Pre Compile
	VendorApiInfix	VariantPC or VariantPB	Pre Compile
	VendorId	VariantPC or VariantPB	Pre Compile

Chapter 9

Integration Steps

This section gives a brief overview of the steps needed for integrating Port :

- Generate the required Port configurations. For more details refer to section [Files required for Compilation](#)
- Allocate proper memory sections in Port_MemMap.h and linker command file. For more details refer to section
- Compile & build the Port with all the dependent modules. For more details refer to section [Building the Driver](#)



Chapter 10

External Assumptions for PORT driver

The section presents requirements that must be complied with when integrating PORT driver into the application.

[SMCAL_CPR_EXT60]

<< The application shall ensure that Port_Init() is not preempting itself or other PORT functions. >>

[SMCAL_CPR_EXT62]

<< The application shall ensure that Port_SetPinDirection() and Port_SetPinMode() are not preempting themselves or one each other when called on the same port. >>



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

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

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

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

© 2017 NXP B.V.

Document Number IM35PORTASR4.2 Rev002 R1.0.0
Revision 1.0