Integration Manual

for MPC574XG Eth Driver

Document Number: IM35ETHASR4.2 Rev0002 R1.0.0
Rev. 1.0.0

Contents

Section number **Title** **Page**

Chapter 1
Revision History

Chapter 2
Introduction

2.1 Supported Derivatives....7

2.2 Acronyms and Definitions....8

2.3 Reference List....8

Chapter 3
Building the Driver

3.1 Build Options....11

3.1.1 GHS Compiler/Linker/Assembler Options....11

3.1.2 DIAB Compiler/Linker/Assembler Options....13

3.2 Files Required for the Compilation....15

3.3 Setting up the Plugins....16

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....21

5.2 Critical Region Exclusivity Matrix....21

5.3 Peripheral Hardware Requirements....22

5.4 ISR to Configure within OS - Dependencies....22

5.5 Other AUTOSAR Modules - Dependencies....22

5.6 User Mode Support....23

Chapter 6
Main API Requirements

Section number Title Page

6.1 Main functions calls within BSW scheduler....25

6.2 Calls to Notification Functions, Callbacks, Callouts....25

Chapter 7
Memory Allocation

7.1 Sections to Be Defined in MemMap.h....27

7.2 Linker command file....29

Chapter 8
Configuration parameters considerations

8.1 Configuration parameters....31

Chapter 9
Integration Steps

Chapter 10
External Assumptions for ETH driver

Chapter 1
Revision History

Table 1-1. Revision History

Revision	Date	Author	Description
1.0.0	18th-Feb-2017	Vu Van Thinh	Initial version for Calypso RTM 1.0.0 ASR 4.2 Release

Chapter 2
Introduction

This integration manual describes the integration requirements for ETH 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 Acronyms and Definitions

Table 2-2. Acronyms and Definitions

Term	Definition
API	Application Programming Interface
AUTOSAR	Automotive Open System Architecture
DEM	Diagnostic Event Manager
DET	Development Error Tracer
ENET	Ethernet MAC (Ethernet Controller)
ETH	Ethernet
ETHIF	Ethernet Interface
FEC	Fast Ethernet Controlled (Ethernet Controller)
FIFO	First-In First-Out Reception Storage
N/A	Not Applicable
MCU	Micro Controller Unit
MII	Media Independent Interface
RAM	Random Access Memory
RMII	Reduced Media Independent Interface
VLE	Variable Length Encoding

- The term "Ethernet Controller" is related to the hardware module providing the Ethernet functionality.
- The term "Ethernet Driver" is related to the software handling the Ethernet Controller.
- The term "Application" is used for the software utilizing the Ethernet Driver.

2.3 Reference List

Table 2-3. Reference List

#	Title	Version
1	AUTOSAR 4.2 Rev0002ETH Driver Software Specification Document.	4.2 Rev002
2	MPC5748G Reference Manual	Rev. 5, 12/2016
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 ETH driver for NXP SemiconductorMPC574XG . It also explains the EB Tresos Studio plugin setup procedure.

3.1 Build Options

The ETH 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 GHS Compiler/Linker/Assembler Options

Table 3-1. 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 3-2. 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 3-3. 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

3.1.2 DIAB Compiler/Linker/Assembler Options

Table 3-4. 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.

Table continues on the next page...

Table 3-4. Compiler Options (continued)

Option	Description
-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.
-DDIAB	-D defines a preprocessor symbol and optionally can set it to a value. This one defines the DIAB preprocessor symbol.
-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.
-c	Stop after assembly, produce object file.

Table 3-5. 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

Table continues on the next page...

Table 3-5. Assembler Options (continued)

Option	Description
-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 3-6. 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

3.2 Files Required for the Compilation

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

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

Ethernet Driver Files

- Eth_TS_T2D35M10I0R0\src\Eth.c
- Eth_TS_T2D35M10I0R0\src\Eth_Irq.c
- Eth_TS_T2D35M10I0R0\src\Eth_Buffers.c
- Eth_TS_T2D35M10I0R0\src\Eth_Ipw.c
- Eth_TS_T2D35M10I0R0\src\Eth_Enet.c

- Eth_TS_T2D35M10I0R0\include\Eth.h
- Eth_TS_T2D35M10I0R0\include\Eth_Enet_Counters.h
- Eth_TS_T2D35M10I0R0\include\Eth_Irq.h
- Eth_TS_T2D35M10I0R0\include\Eth_Ipw.h
- Eth_TS_T2D35M10I0R0\include\Eth_Enet.h

- Eth_variant_PBcfg.c - This file should be generated by the user using a configuration tool for compilation. The file name depend on naming of variant define in ECUC.
- Eth_Cfg.h - This file should be generated by the user using a configuration tool for compilation
- Eth_Cfg.c - This file should be generated by the user using a configuration tool for compilation

Files from Base common folder
- Base_TS_T2D35M10I0R0\include\*.h

Files from Det folder:
- Det_TS_T2D35M10I0R0\include\Det.h
- Det_TS_T2D35M10I0R0\src\Det.c

Files from Dem folder:
- Dem_TS_T2D35M10I0R0\include\Dem.h
- Dem_TS_T2D35M10I0R0\src\Dem.c

Files from EthIf folder:
- EthIf_TS_T2D35M10I0R0\include\EthIf_Cbk.h
- EthIf_TS_T2D35M10I0R0\src\EthIf_Cbk.c

Files from EthTrcv folder:
- EthTrcv_TS_T2D35M10I0R0\include\EthTrcv.h
- EthTrcv_TS_T2D35M10I0R0\src\EthTrcv.c

Files from Rte folder:
- Rte_TS_T2D35M10I0R0\include\SchM_Eth.h

3.3 Setting up the Plugins

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

Location of various files inside the Eth module plug-in folder is explained below.
- Module Parameter Definition:
    - Eth_TS_T2D35M10I0R0\config\Eth.xdm

- Code Generation Templates for Pre-Compile, Link-Time, PostBuild configuration parameters:
    - Eth_TS_T2D35M10I0R0\generate\include\Eth_Cfg.h
    - Eth_TS_T2D35M10I0R0\generate\src\Eth_variant_PBcfg.c
    - Eth_TS_T2D35M10I0R0\generate\src\Eth_Cfg.c

Steps to generate the configuration:

1. Copy both the module folder (Eth_TS_T2D35M10I0R0) into EB tresos Studio installation path under "\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.

Using alternative plugins directory:

1. Copy the folder (Eth_TS_T2D35M10I0R0) into desired alternative directory like "<MyDirectoryPath>\<MyDirectory>" under the folders "<MyDirectoryPath>\<MyDirectory>\eclipse\plugins".
2. Create a text file with the extension ".link" into the EB tresos installation in a folder called "links" (e.g. <MyTresosIntallation>\links\Eth_TS_T2D35M10I0R0.link).
3. Put the following text in that Eth_TS_T2D35M10I0R0.link file: "path=<MyDirectoryPath>/<MyDirectoryName>". Please make sure the path is described with forward slashes.

Chapter 4
Function calls to module

4.1 Function Calls during Start-up

The ETH module shall be initialized by the Eth_Init() function call during the start-up. Please note that GPIO pins used to connect the Ethernet Transceiver have to be properly configured to the appropriate mode prior the ETH module initialization. Configure fast slew-rate for all ethernet pins (otherwise packet losts may occure).

4.2 Function Calls during Shutdown

The application should ensure that all the Ethernet frame transmissions have been completed before the module is shut down by the Eth_SetControllerMode(0, ETH_MODE_DOWN) function call. Note that all not transmitted buffers are flushed and all locked transmit buffers are unlocked when the controller is disabled. All receive buffers are also discarded when the controller is stopped so the Eth_Receive function should be called before the shut down.

4.3 Function Calls during Wake-up

None.

Chapter 5
Module requirements

5.1 Exclusive Areas

The `ETH_EXCLUSIVE_AREA_00` is used in Eth_Transmit to protect Eth_u8LockedTxBufCount (counter to store the number of messages sent need confirmation). When one of the functions using the shared resources enters the exclusive area, the other function must not enter the exclusive area until the first function leaves it.

The `ETH_EXCLUSIVE_AREA_01` is used in Eth_TxConfirmation to protect Eth_Enet_ERR006358() (Process the errata 6358 workaround), Eth_au8TxBufFlags (software flag), Eth_u8LockedTxBufCount (counter to store the number of messages sent need confirmation). When one of the functions using the shared resources enters the exclusive area, the other function must not enter the exclusive area until the first function leaves it.

The `ETH_EXCLUSIVE_AREA_02` is used in Eth_SetGlobalTime to protect Eth_LocalTime (store the value for current time). When one of the functions using the shared resources enters the exclusive area, the other function must not enter the exclusive area until the first function leaves it.

5.2 Critical Region Exclusivity Matrix

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

Table 5-1. Exclusive Areas

	ETH_EXCLUSIVE_AREA_00	ETH_EXCLUSIVE_AREA_01	ETH_EXCLUSIVE_AREA_02
ETH_EXCLUSIVE_AREA_00		x	
ETH_EXCLUSIVE_AREA_01	x		
ETH_EXCLUSIVE_AREA_02			

5.3 Peripheral Hardware Requirements

The Ethernet Transceiver should be connected to the Ethernet Controller module pins which should be configured to be used by the Ethernet Controller. In addition these pins should have configured the highest possible slew rate.

5.4 ISR to Configure within OS - Dependencies

The following ISRs (see Table 5-2 table) are used by the ETH Driver and they need to be assigned to a priority level when the interrupts are switched on (the Ethernet Driver can be run in poll-driven mode).

Table 5-2. Ethernet Driver ISR

ISR Name	Hardware Interrupt Vector
`Eth_RxIrqHdlr_0`	211
`Eth_TxIrqHdlr_0`	212
`Eth_TxRxIrqHdlr_0 *)`	N/A
`Eth_RxIrqHdlr_1`	202
`Eth_TxIrqHdlr_1`	203
`Eth_TxRxIrqHdlr_1 *)`	N/A

5.5 Other AUTOSAR Modules - Dependencies

- **Port:** This module shall configure the Port pins that are used by the Enet.
- **Det** (only if EthDevErrorDetect=true): This module is necessary for enabling the Development Error Detection. The API function used is Det_ReportError(). The activation/deactivation of the Development Error Detection is configurable using the "EthDevErrorDetect" configuration parameter.

- **Dem:** This module is necessary for enabling reporting of the production relevant error status. The API function used is Dem_ReportErrorStatus().
- **Rte:** This module is used for protection of shared internal resources.
- **EthIf:** This module callbacks EthIf_RxIndication and EthIf_TxConfirmation are used by the ETH Driver to notify the upper layers about the Ethernet frame transmission and/or reception.
- **EthTrcv:** This module is used for report indication to transceiver layer by functions: EthTrcv_ReadMiiIndication or EthTrcv_WriteMiiIndication
- **Mcu:** This module is used for get the clock reference for ENET timestamps and initialize clocks
- **Resource:** Sub-Derivative model is selected from Resource configuration.
- **ECUC:** This module is necessary for handle Postbuild Variant.
- **Base:** This module is used for refer type, general define, macro which is used in general.

5.6 User Mode Support

Eth drivers can run in both supervisor mode and user mode. For this platform, there is not any specific measure required for running in user mode.

Chapter 6
Main API Requirements

6.1 Main functions calls within BSW scheduler

ETH Driver support main functions that can be configured to be scheduled by BSW scheduler:

- `FUNC(void, ETH_CODE) Eth_MainFunction(VAR(void, AUTOMATIC))`

The function checks for controller errors and lost frames. Used for polling state changes. Calls EthIf_CtrlModeIndication when the controller mode changed.

6.2 Calls to Notification Functions, Callbacks, Callouts

Notification functions

- None

Call-backs

- `EthIf_RxIndication` This EthIf interface is called when one or more unread Ethernet frames are waiting in the receive buffers to be processed. Call is made from the `Eth_Receive` function or the `Eth_RxIrqHdlr_0` interrupt handler.
- `EthIf_TxConfirmation` This EthIf interface is called when one or more Ethernet frames were transmitted and had been set to confirm the transmission. Call is made from the `Eth_TxConfirmation` function or the `Eth_TxIrqHdlr_0` interrupt handler.

Callouts

- None

Chapter 7 Memory Allocation

7.1 Sections to Be Defined in MemMap.h

Table 7-1. Sections to be defined in MemMap.h

Section name	Type of section	Description
ETH_START_SEC_CONFIG_DATA_UNSPECIFIED	Configuration Data	Start of Memory Section for Config Data.
ETH_STOP_SEC_CONFIG_DATA_UNSPECIFIED	Configuration Data	End of Memory Section for Config Data.
ETH_START_SEC_CODE	Code	Start of memory Section for Code.
ETH_STOP_SEC_CODE	Code	End of memory Section for Code.
ETH_START_SEC_VAR_NO_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 never cleared and never initialized by start-up code.
ETH_STOP_SEC_VAR_NO_INIT_UNSPECIFIED	Variables	End of above section.
ETH_START_SEC_VAR_NO_INIT_8	Variables	Used for variables which have to be aligned to 8 bit. For instance used for variables of size 8 bit or used for composite data types: arrays, structs containing elements of maximum 8 bits. These variables are never cleared and never initialized by start-up code.
ETH_STOP_SEC_VAR_NO_INIT_8	Variables	End of above section.
ETH_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.
ETH_STOP_SEC_VAR_INIT_UNSPECIFIED	Variables	End of above section.
ETH_START_SEC_VAR_INIT_32	Variables	Used for variables which have to be aligned to 32 bit. For instance used for variables of size 32 bit or used

Table continues on the next page...

Table 7-1. Sections to be defined in MemMap.h (continued)

Section name	Type of section	Description
		for composite data types: arrays, structs containing elements of maximum 32 bits. These variables are initialized with values after every reset.
ETH_STOP_SEC_VAR_INIT_32	Variables	End of above section.
ETH_START_SEC_VAR_INIT_16	Variables	Used for variables which have to be aligned to 16 bit. For instance used for variables of size 16 bit or used for composite data types: arrays, structs containing elements of maximum 16 bits. These variables are initialized with values after every reset.
ETH_STOP_SEC_VAR_INIT_16	Variables	End of above section.
ETH_START_SEC_VAR_INIT_8	Variables	Used for variables which have to be aligned to 8 bit. For instance used for variables of size 8 bit or used for composite data types: arrays, structs containing elements of maximum 8 bits. These variables are initialized with values after every reset.
ETH_STOP_SEC_VAR_INIT_8	Variables	End of above section.
ETH_START_SEC_VAR_INIT_BOOLEAN	Variables	Used for initialized boolean variables.
ETH_STOP_SEC_VAR_INIT_BOOLEAN	Variables	End of above section.
ETH_START_SEC_VAR_NO_INIT_UNSPECIFIED_NO_CACHEABLE	Non-Cacheable Variables	Used for variables, structures, arrays when the SIZE (alignment) does not fit the criteria of 8,16 or 32 bit. Normally, this section is used for store Eth Driver buffer memory. **CAUTION:** This section must be cache inhibited.
ETH_STOP_SEC_VAR_NO_INIT_UNSPECIFIED_NO_CACHEABLE	Non-Cacheable Variables	End of above section.
ETH_START_SEC_CONST_32	Constant	Used for constant which have to be aligned to 32 bit. For instance used for constant of size 32 bit or used for composite data types: arrays, structs containing elements of maximum 32 bits. These constant are initialized with values after every reset.
ETH_STOP_SEC_CONST_32	Constant	End of above section.
ETH_START_SEC_CONST_UNSPECIFIED	Constant	Used for constant when the SIZE (alignment) does not fit the criteria of 8,16 or 32 bit. For instance used for constant for composite data types: arrays, structs, etc . These

Table continues on the next page...

Table 7-1. Sections to be defined in MemMap.h (continued)

Section name	Type of section	Description
		constant are initialized with values after every reset.
ETH_STOP_SEC_CONST_UNSPECIFIED	Constant	End of above section.

7.2 Linker command file

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

Chapter 8
Configuration parameters considerations

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

8.1 Configuration parameters

Table 8-1. Configuration Parameters

Configuration Container	Configuration Parameters	Configuration Variant	Current Implementation
EthGeneral			
	EthDevErrorDetect	PC	PC
	EthIndex	PC	PC
	EthMaxCtrlsSupported	PC	PC
	EthVersionInfoApi	PC	PC
	EthVersionInfoApiMacro	PC	PC
	EthUpdatePhysAddrFilter	PC	PC
	EthEnableUserModeSupport	PC	PC
EthGeneral/EthVendorSpecific			
	EthMulticastPoolSize	PC	PC
	EthUseMultiBufferRxFrames	PC	PC
	EthEnableRxFrameWrap	PC	PC
	EthUseMultiBufferTxFrames	PC	PC
	EthDisableDemEventDetect	PC	PC
	EthMaxTXBuffersSupported	PC	PC
EthConfigSet_n/EthCtrlConfig/ EthCtrlConfig_n			
	EthCtrlEnableMii	PC/PB/LT	PC
	EthCtrlEnableRxInterrupt	PC/PB/LT	PC
	EthCtrlEnableTxInterrupt	PC/PB/LT	PC
	EthCtrlIdx	PC/PB/LT	PC
	EthCtrlRxBufLenByte	PC/PB/LT	PC/PB/LT
	EthCtrlTxBufLenByte	PC/PB/LT	PC/PB/LT

Table continues on the next page...

Table 8-1. Configuration Parameters (continued)

Configuration Container	Configuration Parameters	Configuration Variant	Current Implementation
	EthRxBufTotal	PC/PB/LT	PC/PB/LT
	EthTxBufTotal	PC/PB/LT	PC/PB/LT
	EthCtrlPhyAddress	PC/PB/LT	PC/PB/LT
EthConfigSet_n/EthCtrlConfig/ EthCtrlConfig_n/ EthDemEventParameterRefs			
	ETH_E_ACCESS	PC/PB/LT	PC/PB/LT
EthConfigSet_n/EthCtrlConfig/ EthCtrlConfig_n/EthVendorSpecific			
	EthPhyInterface	PC/PB/LT	PC/PB/LT
	EthCtrlSupportMDIO	PC/PB/LT	PC/PB/LT
	EthMIISpeedControl	PC/PB/LT	PC/PB/LT
	EthFullDuplexEnable	PC/PB/LT	PC/PB/LT
	EthReceiveBroadcast	PC/PB/LT	PC/PB/LT
	EthEnablePromiscuousMode	PC/PB/LT	PC/PB/LT
	EthDropInvalidMAC	PC/PB/LT	PC/PB/LT
	EthInterPacketGap	PC/PB/LT	PC/PB/LT
	EthMDIOHoldTime	PC/PB/LT	PC/PB/LT
EthConfigSet_n/EthCtrlConfig/ EthCtrlConfig_n/EthVendorSpecific/ EthEnableLoopbackMode		PC/PB/LT	PC/PB/LT
	EthInternalLoopbackMode	PC/PB/LT	PC/PB/LT

Chapter 9
Integration Steps

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

- Generate the required ETH configurations. For more details refer to section Files Required for the Compilation
- Allocate proper memory sections in ETH_MemMap.h and linker command file. For more details refer to section Sections to Be Defined in MemMap.h
- Compile & build the ETH with all the dependent modules. For more details refer to section Building the Driver

Chapter 10
External Assumptions for ETH driver

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

[SMCAL_CPR_EXT163]

<< If interrupts are locked a centralized function pair to lock and unlock interrupts shall be used. >>

[SMCAL_CPR_EXT177]

<< When caches are enabled and data buffers are allocated in cachable memory regions the buffers involved in DMA transfer shall be aligned with both start and end to cache line size.

>>

NOTE

Rationale: This ensures that no other buffers/variables to compete for the same cache lines.

[SWS_Eth_00149]

<< The types specified in SWS_EthernetDriver shall be declared in Eth_GeneralTypes.h. >>

NOTE

Under control of BASE module

[SWS_Eth_00157]

<< Name: Eth_ReturnType

Type: Enumeration

Range: ETH_OK success

ETH_E_NOT_OK general failure

ETH_E_NO_ACCESS Ethernet hardware access failure

Description: Ethernet Driver specific return type.

>>

NOTE

Under control of BASE module

[SWS_Eth_00158]

<< Name: Eth_ModeType

Type: Enumeration

Range: ETH_MODE_DOWN Controller disabled

ETH_MODE_ACTIVE Controller enabled

Description: This type defines the controller modes

>>

NOTE

Under control of BASE module

[SWS_Eth_00159]

<< Name: Eth_StateType

Type: Enumeration

Range: ETH_STATE_UNINIT Driver is not yet configured

ETH_STATE_INIT Driver is configured

Description: Status supervision used for Development Error Detection. The state shall be available for debugging.

>>

NOTE

Under control of BASE module

[SWS_Eth_00160]

<< Name: Eth_FrameType

Type: --

Range: uint16 0x0000 - 0xFFFF See [21]

Description: This type defines the Ethernet frame type used in the Ethernet frame header

>>

NOTE

Under control of BASE module

[SWS_Eth_00161]

<< Name: Eth_DataType

Type: --

Range: uint8 0x00 - 0xFF 8, 16 or 32 bit CPU

uint16 0x0000 - 0xFFFF 8 or 16 bit CPU

uint32 0x00000000 - 0xFFFFFFFF 32 bit CPU

Description: This type defines the Ethernet data type used for data transmission. Its definition depends on the used CPU.

>>

NOTE

Under control of BASE module

[SWS_Eth_00175]

<< Ethernet buffer identifier type >>

NOTE

Under control of BASE module

[SWS_Eth_00162]

<< Name: Eth_RxStatusType

Type: Enumeration

Range: ETH_RECEIVED Ethernet frame has been received, no further frames available

ETH_NOT_RECEIVED Ethernet frame has not been received, no further frames available

ETH_RECEIVED_MORE_DATA_AVAILABLE Ethernet frame has been received, more frames are available

Description: Used as out parameter in Eth_Receive() indicates whether a frame has been received and if so, whether more frames are available or frames got lost.

>>

NOTE

Under control of BASE module

[SWS_Eth_00163]

<< Name: Eth_FilterActionType

Type: Enumeration

Range: ETH_ADD_TO_FILTER add the MAC address to the filter, meaning allow reception

ETH_REMOVE_FROM_FILTER remove the MAC address from the filter, meaning reception is blocked in the lower layer

Description: The Enumeration Type Eth_FilterActionType describes the action to be taklen for the MAC address given in *PhysAddrPtr.

>>

NOTE

Under control of BASE module

[SWS_Eth_00177]

<< Name: Eth_TimeStampQualType

Type: Enumeration

Range: ETH_VALID 0

ETH_INVALID 1

ETH_UNCERTAIN 2

Description: Depending on the HW, quality information regarding the evaluated time stamp might be supported. If not supported, the value shall be always Valid. For Uncertain and Invalid values, the upper layer shall discard the time stamp.

>>

NOTE

Under control of BASE module

[SWS_Eth_00178]

<< Name: Eth_TimeStampType

Type: Structure

Element: uint32 nanoseconds Nanoseconds part of the time

uint32 seconds 32 bit LSB of the 48 bits Seconds part of the time

uint16 secondsHi 16 bit MSB of the 48 bits Seconds part of the time

Description: Variables of this type are used for expressing time stamps including relative time and absolute calendar time. The absolute time starts acc. to "[5], Annex C/C1" specification at 1970-01-01.

0 to 281474976710655s

== 3257812230d

[0xFFFF FFFF FFFF]

0 to 999999999ns

[0x3B9A C9FF]

invalid value in nanoseconds: [0x3B9A CA00] to [0x3FFF FFFF]

Bit 30 and 31 reserved, default: 0

>>

NOTE

Under control of BASE module

[SWS_Eth_00179]

<< Name: Eth_TimeIntDiffType

Type: Structure

Element: Eth_TimeStampType diff time difference

boolean sign Positive (True) / negative (False) time

Description: Variables of this type are used to express time differences in a usual way. The described "TimeInterval" type referenced in "[5], chapter 6.3.3.3" will not be used and hereby slightly simplified.

>>

NOTE

Under control of BASE module

[SWS_Eth_00180]

<< Name: Eth_RateRatioType

Type: Structure

Element: Eth_TimeIntDiffType IngressTimeStampDelta IngressTimeStampSync2 - IngressTimeStampSync1

Eth_TimeIntDiffType OriginTimeStampDelta OriginTimeStampSync2[FUP2] - OriginTimeStampSync1[FUP1]

Description: Variables of this type are used to express frequency ratios.

>>

NOTE

Under control of BASE module

[SWS_Eth_00120]

<< This tabe defines all interfaces required to fulfill the core functionality of the DET module. >>

NOTE

This is not Eth requirement

How to Reach Us:

Home Page:
nxp.com

Web Support:
nxp.com/support

Document Number IM35ETHASR4.2 Rev0002 R1.0.0
Revision 1.0.0