

Document Title	Specification of Platform Types
Document Owner	AUTOSAR
Document Responsibility	AUTOSAR
Document Identification No	048
Document Classification	Standard

Document Version	2.3.0
Document Status	Final
Part of Release	3.0
Revision	7

	Document Change History		
Date	Version	Changed by	Change Description
15.09.2010	2.3.0	AUTOSAR Administration	 Replaced generic <module> by "PLATFORM" in chapter 10</module> Legal disclaimer revised
13.11.2007	2.2.0	AUTOSAR Administration	 Chapter 8.2: "AUTOSAR supports for compiler and target implementation only 2 complement arithmetic" Chapter 12.10: changed the basic type for *_least types (optimized types) from 'int' to 'long' for SHx processors Removal the explicit cast to boolean in the precompile definition (#define) for macros TRUE and FALSE ("#define TRUE ((boolean) 1)" has become "#define TRUE 1") Document meta information extended Small layout adaptations made
31.01.2007	2.1.0	AUTOSAR Administration	 Boolean type has been defined as an eight bit long unsigned integer Legal disclaimer revised Release Notes added "Advice for users" revised "Revision Information" added
12.07.2006	2.0.0	AUTOSAR Administration	Second release
30.06.2005	1.0.0	AUTOSAR Administration	Initial Release



Disclaimer

This specification and the material contained in it, as released by AUTOSAR is for the purpose of information only. AUTOSAR and the companies that have contributed to it shall not be liable for any use of the specification.

The material contained in this specification is protected by copyright and other types of Intellectual Property Rights. The commercial exploitation of the material contained in this specification requires a license to such Intellectual Property Rights.

This specification may be utilized or reproduced without any modification, in any form or by any means, for informational purposes only.

For any other purpose, no part of the specification may be utilized or reproduced, in any form or by any means, without permission in writing from the publisher.

The AUTOSAR specifications have been developed for automotive applications only. They have neither been developed, nor tested for non-automotive applications.

The word AUTOSAR and the AUTOSAR logo are registered trademarks.

Advice for users

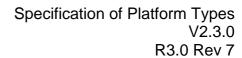
AUTOSAR Specification Documents may contain exemplary items (exemplary reference models, "use cases", and/or references to exemplary technical solutions, devices, processes or software).

Any such exemplary items are contained in the Specification Documents for illustration purposes only, and they themselves are not part of the AUTOSAR Standard. Neither their presence in such Specification Documents, nor any later documentation of AUTOSAR conformance of products actually implementing such exemplary items, imply that intellectual property rights covering such exemplary items are licensed under the same rules as applicable to the AUTOSAR Standard.



Table of Contents

1	Intro	duction and functional overview	5
2	Acro	nyms and abbreviations	6
3	Rela	ted documentation	7
		Input documentsRelated standards and norms	
4	Cons	straints and assumptions	8
	4.2	Limitations Applicability to car domains Applicability to safety related environments	8
5	Depe	endencies to other modules	9
	5.1.1 5.1.2	Header file structure1	9 9 10
6	Requ	uirements traceability1	11
	6.1	Linkage items for requirements management1	15
7	Fund	ctional specification1	16
	7.2 7.3 7.3.1 7.3.2	2 Byte Ordering (Memory)1	16 16 16
		Optimized integer data types1 boolean data type1	
8		specification	
		2 uint8	20 20 20
	8.2.4 8.2.5	l uint32	20 22
	8.2.6 8.2.7 8.2.8	7 sint32	22
	8.2.9 8.2.1	9 uint16_least2	22
	8.2.1 8.2.1	1 sint8_least 2 2 sint16_least 2	23 23
	8.2.1	3 sint32_least	23





8.2.	14 float32	23
8.2.	15 float64	23
8.3	Symbol definitions	
8.3.		
8.3.		
8.3.		
8.3.	,	
8.4	Function definitions	
8.5	Call-back notifications	
8.6	Scheduled functions	
8.7	Expected Interfaces	25
9 Seq	uence diagrams	26
10 C	onfiguration specification	27
10.1	Published parameters	27
11 C	hanges to Release 1	28
11.1	Deleted SWS Items	28
11.2	Replaced SWS Items	28
11.3	Changed SWS Items	28
11.4	Added SWS Items	28
12 A	nnex	29
12.1	Type definitions – general	29
12.2	Type definitions – S12X	29
12.3	Type definitions – ST10	
12.4	Type definitions – ST30	
12.5	Type definitions – V850	
12.6	Type definitions – MPC5554	
12.7	Type definitions – TC1796/TC1766	32
12.8	Type definitions – MB91F	33
12.9	Type definitions – M16C/M32C	
12.10	Type definitions – SHx	



1 Introduction and functional overview

This document specifies the AUTOSAR platform types header file. It contains all platform dependent types and symbols. Those types must be abstracted in order to become platform and compiler independent.

It is required that all platform types files are unique within the AUTOSAR community to guarantee unique types per platform and to avoid type changes when moving a software module from platform A to B.



2 Acronyms and abbreviations

Acronyms and abbreviations that have a local scope are not contained in the AUTOSAR glossary. These must appear in a local glossary.

Acronym:	Description:
Rollover mechanism	The following example sequence is called 'rollover': • An unsigned char has the value of 255 • It is incremented by 1 • The result is 0
SDU	Service Data Unit (payload)

Abbreviation:	Description:
int	Integer



3 Related documentation

3.1 Input documents

- [1] General Requirements on Basic Software Modules, AUTOSAR_SRS_General.pdf
- [2] AUTOSAR Basic Software Module Description Template, AUTOSAR_BSW_Module_Description.pdf
- [3] Cosmic C Cross Compiler User's Guide for Motorola MC68HC12, V4.5
- [4] ARM ADS compiler manual
- [5] Greenhills MULTI for V850 V4.0.5:Building Applications for Embedded V800, V4.0, 30.1.2004
- [6] TASKING for ST10 V8.5: C166/ST10 v8.5 C Cross-Compiler User's Manual, V5.16 C166/ST10 v8.5 C Cross-Assembler, Linker/Locator, Utilities User's Manual, V5.16
- [7] Wind River (Diab Data) for PowerPC Version 5.2.1: Wind River Compiler for Power PC - Getting Started, Edition 2, 8.5.2004 Wind River Compiler for Power PC - User's Guide, Edition 2, 11.5.2004
- [8] TASKING for TriCore TC1796 V2.1R1: TriCore v2.0 C Cross-Compiler, Assembler, Linker User's Guide, V1.2
- [9] Metrowerks CodeWarrior 4.0 for Freescale HC9S12X/XGATE (V5.0.25): Motorola HC12 Assembler, 2.6.2004 Motorola HC12 Compiler, 2.6.2004 Smart Linker, 2.4.2004

3.2 Related standards and norms

- [10] ISO/IEC 9899:1990 Programming Language C
- [11] MISRA-C 2004: Guidelines for the use of the C language in critical systems, October 2004



4 Constraints and assumptions

4.1 Limitations

No limitations.

4.2 Applicability to car domains

No restrictions.

4.3 Applicability to safety related environments

The AUTOSAR boolean type may be used if the correct usage (see <u>PLATFORM027</u>) is proven by a formal code review or a static analysis by a validated static analysis tool.

The optimized AUTOSAR integer data types (*_least) may be used if the correct usage (see <u>PLATFORM005</u>) is proven by a formal code review or a static analysis by a validated static analysis tool.



5 Dependencies to other modules

None.

5.1 File structure

5.1.1 Code file structure

None

5.1.2 Header file structure

Two header file structures are applicable. One is depending on communication related basic software modules and the second is depending on non-communication related basic software modules.



5.1.2.1 Communication related basic software modules

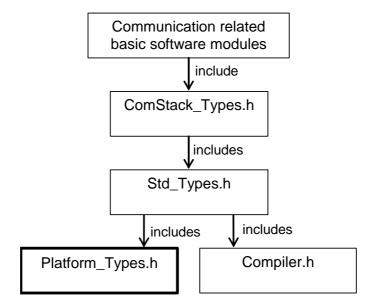


Figure 1: Include File Structure for communication related basic software modules

- <user>_Types.h shall include ComStack_Types.h and <user> is a communication related basic software module (e.g. Com, PduR, Can...)
- ComStack_Types.h shall include Std_Types.h
- Std_Types.h shall include Platform_Types.h
- Std Types.h shall include Compiler.h

5.1.3 Non-communication related basic software modules

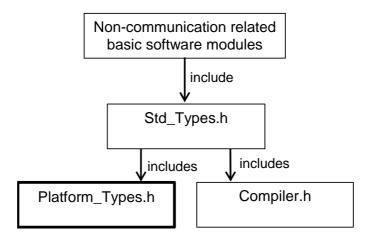


Figure 2: Include File Structure for non-communication related basic software modules

- <user>_Types.h shall include ComStack_Types.h and user is a noncommunication related basic software module(e.g. Mcu, WdgM ...)
- Std_Types.h shall include Platform_Types.h
- Std_Types.h shall include Compiler.h



6 Requirements traceability

Document: General Requirements on Basic Software Modules,

BSW00344 Reference to link-time configuration (i.e. PlatformTypes SWS specifies a header file) (IBSW00404 Reference to post build time configuration (i.e. PlatformTypes SWS specifies a header file) (IBSW00345 Reference to multiple configuration sets (IBSW00345 Pre-compile-time configuration (I.e. PlatformTypes SWS specifies a header file) (IBSW00345 Pre-compile-time configuration (I.e. PlatformTypes SWS specifies a header file) (I.e. PlatformTypes SWS specifies a header file) (I.e. PlatformTypes SWS specifies a header file) (I.e. PlatformTypes SWS is not a module specific configuration file) (I.e. PlatformTypes SWS is not a BSW module) (I.e. PlatformTypes SWS specifies a header file) (I.e. PlatformTypes SWS is not a BSW module) (I.e. PlatformTypes SWS specifies a header file) (I.e. PlatformTypes) (I.e	Requirement	Satisfied by
BSW00405 Reference to post build time configuration (i.e. PlatformTypes SWS specifies a header file) (ii.e. PlatformTypes SWS is not a module specific configuration file) (ii.e. PlatformTypes SWS is not a module specific configuration file) (ii.e. PlatformTypes SWS is not a BSW module) (ii.e. PlatformTypes SWS specifies a header file) (ii.e. PlatformTypes SWS is not a BSW module) (ii.e. PlatformTypes SWS specifies a header file) (ii.e. PlatformTypes SWS specifies a header file) (ii.e. PlatformTypes SWS specifies a header file) (ii.e. PlatformTypes SWS is not a BSW module) (ii.e. PlatformTypes SWS is not a BSW module) (ii.e. PlatformTypes SWS specifies a header file) (ii.e. PlatformTypes SWS is not a BSW module) (ii.e. PlatformTypes SWS is not a BSW module) (ii.e. PlatformTypes SWS specifies a header file) (ii.e. PlatformTypes SWS is not a BSW module) (ii.e. PlatformTypes SWS is not a BSW module) (ii.e. PlatformTypes SWS specifies a header file) (ii.e. PlatformTypes) (ii.e. PlatformTypes) (ii.e.	[BSW00344] Reference to link-time configuration	Not applicable
Configuration (i.e. PlatformTypes SWS specifies a header file)		(i.e. PlatformTypes SWS specifies a header file)
Configuration (i.e. PlatformTypes SWS specifies a header file)	[BSW00404] Reference to post build time	Not applicable
BSW00405] Reference to multiple configuration sets (i.e. PlatformTypes SWS specifies a header file)	1-	
Sets		
BSW00345 Pre-compile-time configuration Not applicable (i.e. PlatformTypes SWS is not a module specific configuration file)		
(i.e. PlatformTypes SWS is not a module specific configuration file) [BSW159] Tool-based configuration [BSW167] Static configuration checking [BSW171] Configurability of optional functionality [BSW170] Data for reconfiguration of AUTOSAR SW-Components [BSW030] Separate C-Files for configuration parameters [BSW00341] Separate C-Files for pre-compile time configuration parameters [BSW00381] Separate configuration header file for pre-compile time configuration parameters [BSW00381] Separate C-Files for pre-compile time parameters [BSW00381] Separate C-Files for pre-compile time parameters [BSW00381] Separate C-Files for pre-compile time parameters [BSW00383] List dependencies of configuration parameters available for platform types) [BSW00383] List dependencies to other modules [BSW00384] List dependencies to other modules [BSW00387] Specify the configuration class of callback function [BSW00388] Introduce containers [BSW00389] Containers shall have names [BSW00389] Containers shall have names [BSW00390] Parameter content shall be unique within the module [BSW00391] Parameter shall have unique names [BSW00392] Parameters shall have unique names [BSW00392] Parameters shall have a type	[BSW00345] Pre-compile-time configuration	
BSW159] Tool-based configuration Not applicable (i.e. PlatformTypes SWS is not a BSW module)		
BSW159 Tool-based configuration		
(i.e. PiatformTypes SWS is not a BSW module) [BSW170] Static configuration checking [BSW170] Data for reconfiguration of AUTOSAR SW-Components [BSW00380] Separate C-Files for configuration parameters [BSW00380] Separate C-Files for pre-compile time configuration parameters [BSW00419] Separate C-Files for pre-compile time configuration parameters [BSW00381] Separate configuration header file for pre-compile time parameters [BSW00381] Separate configuration header file for pre-compile time parameters [BSW00381] Separate th-File for configuration parameters available for platform types) [BSW00381] Separate C-Files for pre-compile (i.e. no configuration parameters available for platform types) [BSW00381] Separate configuration header file for pre-compile time parameters [BSW00381] Separate H-File for configuration parameters available for platform types) [BSW00383] List dependencies of configuration files [BSW00384] List dependencies to other modules [BSW00384] List dependencies to other modules [BSW00384] Specify the configuration class of callback function [BSW00388] Introduce containers [BSW00388] Introduce containers [BSW00389] Containers shall have names [BSW00389] Containers shall have names [BSW00390] Parameter content shall be unique within the module [BSW00391] Parameter shall have unique names [BSW00392] Parameters shall have a type	[BSW159] Tool-based configuration	
BSW167 Static configuration checking Not applicable (i.e. PlatformTypes SWS is not a BSW module)		
(i.e. PlatformTypes SWS is not a BSW module) (i.e. PlatformTypes SWS is not a BSW module) (i.e. PlatformTypes SWS is not a BSW module) (i.e. PlatformTypes SWS specifies a header file) (i.e. PlatformTypes SWS specifies a header file) (i.e. no reconfiguration available for platform types) (i.e. no reconfiguration available for platform types) (i.e. no configuration available for platform types) (i.e. no configuration parameters available for platform types) (i.e. no configur	[BSW167] Static configuration checking	
BSW171] Configurability of optional functionality (i.e. PlatformTypes SWS specifies a header file) (i.e. no reconfiguration available for platform types) (i.e. no configuration parameters available for platform types) (i.e. no config	The state of the s	1
(i.e. PlatformTypes SWS specifies a header file) Not applicable (i.e. no reconfiguration available for platform types) (i.e. no reconfiguration available for platform types) Not applicable (i.e. no reconfiguration available for platform types) Not applicable (i.e. no reconfiguration parameters available for platform types) (i.e. no configuration par	[BSW171] Configurability of optional functionality	
BSW170 Data for reconfiguration of AUTOSAR SW-Components	[]	1 ' '
SW-Components (i.e. no reconfiguration available for platform types)	[BSW170] Data for reconfiguration of AUTOSAR	
BSW00380 Separate C-Files for configuration parameters Not applicable (i.e no configuration parameters available for platform types)		
December 2 (i.e. no configuration parameters available for platform types)		
BSW00419 Separate C-Files for pre-compile time configuration parameters Not applicable (i.e no configuration parameters available for platform types)	1-	
BSW00381 Separate C-Files for pre-compile time configuration parameters Not applicable (i.e no configuration parameters available for platform types)	parameters	
time configuration parameters [BSW00381] Separate configuration header file for pre-compile time parameters [BSW00412] Separate H-File for configuration parameters available for platform types) [BSW00383] List dependencies of configuration files [BSW00383] List dependencies of configuration files [BSW00384] List dependencies to other modules [BSW00387] Specify the configuration class of callback function [BSW00388] Introduce containers [BSW00389] Containers shall have names [BSW00390] Parameter content shall be unique within the module [BSW00391] Parameter shall have unique names [BSW00392] Parameters shall have a type [BSW00392] Parameters shall have a type [BSW00392] Parameters shall have a type [BSW00390] Parameters shall have a type	IBSW004191 Separate C-Files for pre-compile	7. /
BSW00381] Separate configuration header file for pre-compile time parameters Not applicable (i.e no configuration parameters available for platform types)		
BSW00381 Separate configuration header file for pre-compile time parameters	anno configuration parameters	
for pre-compile time parameters [BSW00412] Separate H-File for configuration parameters [BSW00383] List dependencies of configuration files [BSW00384] List dependencies to other modules [BSW00384] List dependencies to other modules [BSW00387] Specify the configuration class of callback function [BSW00388] Introduce containers [BSW00389] Containers shall have names [BSW00390] Parameter content shall be unique within the module [BSW00391] Parameter shall have unique names [BSW00392] Parameters shall have a type [BSW00390] Parameters shall have a type	IBSW003811 Separate configuration header file	
BSW00412] Separate H-File for configuration parameters Not applicable (i.e no configuration parameters available for platform types)	1	1
BSW00383 List dependencies of configuration files Not applicable (i.e no configuration parameters available for platform types)	To pro compile umo parametere	
Description Description	[BSW00412] Separate H-File for configuration	
platform types) BSW00383] List dependencies of configuration files		1
BSW00383 List dependencies of configuration files	parameters	
files (i.e no configuration parameters available for platform types) [BSW00384] List dependencies to other modules (i.e no configuration parameters available for platform types) [BSW00387] Specify the configuration class of callback function (i.e no callback function available for platform types) [BSW00388] Introduce containers Not applicable (i.e no configuration parameters available for platform types) [BSW00389] Containers shall have names Not applicable (i.e no configuration parameters available for platform types) [BSW00390] Parameter content shall be unique within the module (i.e no configuration parameters available for platform types) [BSW00391] Parameter shall have unique names Not applicable (i.e no configuration parameters available for platform types) [BSW00392] Parameters shall have a type Not applicable (i.e no configuration parameters available for platform types)	IBSW003831 List dependencies of configuration	
BSW00384] List dependencies to other modules Not applicable (i.e no configuration parameters available for platform types)		
[BSW00384] List dependencies to other modules (i.e no configuration parameters available for platform types) [BSW00387] Specify the configuration class of callback function (i.e no callback function available for platform types) [BSW00388] Introduce containers Not applicable (i.e no configuration parameters available for platform types) [BSW00389] Containers shall have names Not applicable (i.e no configuration parameters available for platform types) [BSW00390] Parameter content shall be unique within the module (i.e no configuration parameters available for platform types) [BSW00391] Parameter shall have unique names Not applicable (i.e no configuration parameters available for platform types) [BSW00392] Parameters shall have a type Not applicable		
(i.e no configuration parameters available for platform types) [BSW00387] Specify the configuration class of callback function (i.e no callback function available for platform types) [BSW00388] Introduce containers Not applicable (i.e no configuration parameters available for platform types) [BSW00389] Containers shall have names Not applicable (i.e no configuration parameters available for platform types) [BSW00390] Parameter content shall be unique within the module (i.e no configuration parameters available for platform types) [BSW00391] Parameter shall have unique names Not applicable (i.e no configuration parameters available for platform types) [BSW00392] Parameters shall have a type Not applicable [BSW00392] Parameters shall have a type Not applicable	[BSW00384] List dependencies to other modules	
platform types)	[
[BSW00387] Specify the configuration class of callback function (i.e no callback function available for platform types) [BSW00388] Introduce containers Not applicable (i.e no configuration parameters available for platform types) [BSW00389] Containers shall have names Not applicable (i.e no configuration parameters available for platform types) [BSW00390] Parameter content shall be unique within the module (i.e no configuration parameters available for platform types) [BSW00391] Parameter shall have unique names Not applicable (i.e no configuration parameters available for platform types) [BSW00392] Parameters shall have a type Not applicable		
[BSW00388] Introduce containers [BSW00388] Introduce containers [BSW00389] Containers shall have names [BSW00389] Containers shall have names [BSW00390] Parameter content shall be unique within the module [BSW00391] Parameter shall have unique names [BSW00392] Parameters shall have a type (i.e no callback function available for platform types) Not applicable (i.e no configuration parameters available for platform types) Not applicable (i.e no configuration parameters available for platform types) Not applicable (i.e no configuration parameters available for platform types)	IBSW003871 Specify the configuration class of	
[BSW00388] Introduce containers [BSW00389] Containers shall have names [BSW00390] Parameter content shall be unique within the module [BSW00391] Parameter shall have unique names [BSW00391] Parameter shall have unique names [BSW00392] Parameters shall have a type [BSW00392] Parameters shall have a type [BSW00388] Introduce containers Not applicable (i.e no configuration parameters available for platform types) Not applicable (i.e no configuration parameters available for platform types)	1	
[BSW00388] Introduce containers Not applicable (i.e no configuration parameters available for platform types) [BSW00389] Containers shall have names Not applicable (i.e no configuration parameters available for platform types) [BSW00390] Parameter content shall be unique within the module within the module [BSW00391] Parameter shall have unique names [BSW00392] Parameters shall have a type Not applicable (i.e no configuration parameters available for platform types) Not applicable (i.e no configuration parameters available for platform types)		1 `
(i.e no configuration parameters available for platform types) [BSW00389] Containers shall have names Not applicable (i.e no configuration parameters available for platform types) [BSW00390] Parameter content shall be unique within the module within the module [BSW00391] Parameter shall have unique names [BSW00392] Parameters shall have a type (i.e no configuration parameters available for platform types) [BSW00392] Parameters shall have a type Not applicable Not applicable	IBSW003881 Introduce containers	
[BSW00389] Containers shall have names [BSW00390] Parameter content shall be unique within the module [BSW00391] Parameter shall have unique names [BSW00391] Parameter shall have unique names [BSW00392] Parameters shall have a type Dot applicable (i.e no configuration parameters available for platform types) Not applicable (i.e no configuration parameters available for platform types) Not applicable (i.e no configuration parameters available for platform types) Dot applicable (i.e no configuration parameters available for platform types) Dot applicable (i.e no configuration parameters available for platform types) Dot applicable (i.e no configuration parameters available for platform types) Dot applicable (i.e no configuration parameters available for platform types)		
[BSW00389] Containers shall have names Not applicable (i.e no configuration parameters available for platform types) [BSW00390] Parameter content shall be unique within the module [BSW00391] Parameter shall have unique names [BSW00391] Parameter shall have unique names [BSW00392] Parameters shall have a type Not applicable (i.e no configuration parameters available for platform types) [BSW00392] Parameters shall have a type Not applicable		
(i.e no configuration parameters available for platform types) [BSW00390] Parameter content shall be unique within the module Within the module [BSW00391] Parameter shall have unique names [BSW00391] Parameter shall have unique names [BSW00392] Parameters shall have a type (i.e no configuration parameters available for platform types) [BSW00392] Parameters shall have a type Not applicable	[BSW00389] Containers shall have names	
[BSW00390] Parameter content shall be unique within the module [BSW00391] Parameter shall have unique names [BSW00391] Parameter shall have unique names [BSW00392] Parameters shall have a type [BSW00392] Parameters shall have a type [BSW00390] Parameters shall have a type		
[BSW00390] Parameter content shall be unique within the module (i.e no configuration parameters available for platform types) Not applicable (i.e no configuration parameters available for platform types) [BSW00392] Parameters shall have a type Not applicable		
within the module (i.e no configuration parameters available for platform types) [BSW00391] Parameter shall have unique names Not applicable (i.e no configuration parameters available for platform types) [BSW00392] Parameters shall have a type Not applicable	[BSW00390] Parameter content shall be unique	
[BSW00391] Parameter shall have unique names Not applicable (i.e no configuration parameters available for platform types) [BSW00392] Parameters shall have a type Not applicable		
[BSW00391] Parameter shall have unique names		
(i.e no configuration parameters available for platform types) [BSW00392] Parameters shall have a type Not applicable	[BSW00391] Parameter shall have unique names	
platform types) [BSW00392] Parameters shall have a type Not applicable		
[BSW00392] Parameters shall have a type Not applicable		
	[BSW00392] Parameters shall have a type	
		(i.e no configuration parameters available for



	platform types)
[BSW00393] Parameters shall have a range	Not applicable
[BOVVOOCOO] i didiiiotolo ondii navo a rango	(i.e no configuration parameters available for
	platform types)
[BSW00394] Specify the scope of the parameters	Not applicable
[Bettered if epoons are deepe of the parameter	(i.e no configuration parameters available for
	platform types)
[BSW00395] List the required parameters (per	Not applicable
parameter)	(i.e no configuration parameters available for
parametery	platform types)
[BSW00396] Configuration classes	Not applicable
[BOTTOGOOG] Cormiguration diacodo	(i.e no configuration parameters available for
	platform types)
[BSW00397] Pre-compile-time parameters	Not applicable
[Bettesser] i to semple time parameters	(i.e no configuration parameters available for
	platform types)
[BSW00398] Link-time parameters	Not applicable
[DOW00390] Link-time parameters	(i.e no configuration parameters available for
	platform types)
[BSW00399] Loadable Post-build time	Not applicable
parameters	(i.e no configuration parameters available for
parameters	
[PSW/00400] Salastable Boot build time	platform types) Not applicable
[BSW00400] Selectable Post-build time	
parameters	(i.e no configuration parameters available for
IDCM/004001 Dublish addinformation	platform types)
[BSW00402] Published information	[PLATFORM012]
[BSW00375] Notification of wake-up reason	Not applicable
[DOMAGA11 32 12 42 4 4	(i.e. no functionality defined with platform types)
[BSW101] Initialization interface	Not applicable
	(i.e. no functionality defined with platform types)
[BSW00416] Sequence of Initialization	Not applicable
	(i.e. no functionality defined with platform types)
[BSW00406] Check module initialization	Not applicable
	(i.e. no functionality defined with platform types)
[BSW168] Diagnostic Interface of SW	Not applicable
components	(i.e. no testing of platform types defined)
[BSW00407] Function to read out published	Not applicable
parameters	(i.e. no functionality defined with platform types)
[BSW00423] Usage of SW-C template to	Not applicable
describe BSW modules with AUTOSAR	(i.e platform types is not a module)
Interfaces	
[BSW00429] Restricted BSW OS functionality	Not applicable
access	(i.e. no functionality defined with platform types. It's
	a header file)
[BSW00432] Modules should have separate main	Not applicable
processing functions for read/receive and	(i.e. no functionality defined with platform types. It's
write/transmit data path	a header file)
[BSW00336] Shutdown interface	Not applicable
	(i.e. no functionality defined in platform types. It's a
	header file)
[BSW00337] Classification of errors	Not applicable
, -	(i.e. no error classification defined with platform
	types)
[BSW00338] Detection and Reporting of	Not applicable
development errors	(i.e. no error classification defined with platform
	types)
[BSW00369] Do not return development error	Not applicable
codes via API	(i.e. no functionality defined in platform types. It's a
10000 110 111	header file)
	noddor moj



[BSW00339] Reporting of production relevant	Not applicable
error status	(i.e. no functionality defined in platform types. It's a
	header file)
[BSW00422] Debouncing of production relevant	Not applicable
error status	(i.e. no functionality defined in platform types. It's a
citor status	header file)
[BSW00420] Production relevant error event rate	Not applicable
	1
detection	(i.e. no functionality defined in platform types. It's a
	header file)
[BSW00417] Reporting of Error Events by Non-	Not applicable
Basic Software	(i.e. no functionality defined in platform types. It's a
	header file)
[BSW00323] API parameter checking	Not applicable
	(i.e. no functionality defined in platform types. It's a
	header file)
[BSW004] Version check	[PLATFORM012]
[BSW00409] Header files for production code	Not applicable
error IDs	(i.e. no error defined with platform types)
[BSW00385] List possible error notifications	Not applicable
[POW/2000] O. (f	(i.e. no error defined with platform types)
[BSW00386] Configuration for detecting an error	Not applicable
	(i.e. no error defined with platform types)
[BSW161] Microcontroller abstraction	Not applicable
	(i.e. no interface provided)
[BSW162] ECU layout abstraction	Not applicable
, ,	(i.e. no interface provided)
[BSW005] No hard coded horizontal interfaces	Not applicable
within MCAL	(i.e. no interface provided)
[BSW00415] User dependent include files	Not applicable (i.e. no interface provided)
[BSW164] Implementation of interrupt service	Not applicable
routines	(i.e. only types are defined here)
[BSW00325] Runtime of interrupt service routines	Not applicable
	(i.e. only types are defined here)
[BSW00326] Transition from ISRs to OS tasks	Not applicable
	(i.e. only types are defined here)
[BSW00342] Usage of source code and object	Not applicable
code	(i.e. only types are defined here)
[BSW00343] Specification and configuration of	Not applicable
time	(i.e. no time configuration provided)
[BSW160] Human-readable configuration data	Not applicable
[DOVV100] Framan readable configuration data	(i.e. only types are defined here)
[BSW007] HIS MISRA C	Not applicable
[BSWUU7] HIS WISKA C	
IDOM/second to the state of the	(i.e. only types are defined here)
[BSW00300] Module naming convention	Not applicable
	(i.e. only types are defined here)
[BSW00413] Accessing instances of BSW	Not applicable
modules	(i.e. only types are defined here)
[BSW00347] Naming separation of different	Not applicable
instances of BSW drivers	(i.e. instantiation of platform types required)
[BSW00305] Self-defined data types naming	Not applicable
convention	(i.e. platform types apply to all BSW modules)
[BSW00307] Global variables naming convention	Not applicable
1250001 Clobal valiables harming convention	(i.e. only types are defined here)
[PSW/00210] API naming convention	
[BSW00310] API naming convention	Not applicable
IDOM/00070114	(i.e. only types are defined here)
[BSW00373] Main processing function naming	Not applicable
convention	(i.e. only types are defined here)
[BSW00327] Error values naming convention	Not applicable
	(i.e. only types are defined here)



	10.01007
[BSW00335] Status values naming convention	Not applicable
IDOM/200701D	(i.e. only types are defined here)
[BSW00350] Development error detection	Not applicable
keyword	(i.e. only types are defined here)
[BSW00408] Configuration parameter naming	Not applicable
convention	(i.e. only types are defined here)
[BSW00410] Compiler switches shall have	Not applicable
defined values	(i.e. compiler switches not provided)
[BSW00411] Get version info keyword	Not applicable
	(i.e. only types are defined here)
[BSW00346] Basic set of module files	Not applicable
	(i.e. only types are defined here)
[BSW158] Separation of configuration from	Not applicable
implementation	(i.e. no configuration provided with platform types)
[BSW00314] Separation of interrupt frames and	Not applicable
service routines	(i.e. only types are defined here)
[BSW00370] Separation of callback interface	Not applicable
from API	(i.e. only types are defined here)
	Not applicable
[BSW00348] Standard type header	
IDOMOGOTOLDI (C	(i.e. platform types are defined here)
[BSW00353] Platform specific type header	PLATFORM001, PLATFORM003
	Chapter 8.2 Type definitions
[BSW00361] Compiler specific language	Not applicable
extension header	(i.e. only types are defined here)
[BSW00301] Limit imported information	Not applicable
	(i.e. only types are defined here)
[BSW00302] Limit exported information	Not applicable
	(i.e. only types are defined here)
[BSW00328] Avoid duplication of code	Not applicable
	(i.e. only types are defined here)
[BSW00312] Shared code shall be reentrant	Not applicable
	(i.e. only types are defined here)
[BSW006] Platform independency	All SWS items present in this document
[BSW00357] Standard API return type	Not applicable
[Bowood7] Standard At Tretum type	(i.e. only types are defined here)
[BSW00377] Module specific API return types	Not applicable (
[D3W00377] Woddie specific AFTTetaiti types	i.e. only types are defined here)
[BSW00304] AUTOSAR integer data types	PLATFORM001, PLATFORM003, PLATFORM005,
[b5vv00304] AUTOSAK integer data types	PLATFORMOO1, PLATFORMOO3, PLATFORMOO5, PLATFORMO15,
	PLATFORM016, PLATFORM017, PLATFORM018,
	PLATFORM020, PLATFORM021, PLATFORM022,
IDOM/SSSTELD	PLATFORM023, PLATFORM024, PLATFORM025
[BSW00355] Do not redefine AUTOSAR integer	Not applicable
data types	(i.e. only types are defined here)
[BSW00378] AUTOSAR boolean type	PLATFORM026, PLATFORM027, PLATFORM034
[BSW00306] Avoid direct use of compiler and	Not applicable
platform specific keywords	(i.e. this SWS does not specify a module)
[BSW00308] Definition of global data	Not applicable
	(i.e. only types are defined here)
[BSW00309] Global data with read-only	Not applicable
constraint	(i.e. only types are defined here)
[BSW00371] Do not pass function pointers via	Not applicable
API	(i.e. only types are defined here)
[BSW00358] Return type of init() functions	Not applicable
	(i.e. only types are defined here)
[BSW00414] Parameter of init function	Not applicable
[DOVVOO414] Farameter of finit function	
IDCM/002761 Deturn type and nevertees of making	(i.e. only types are defined here)
[BSW00376] Return type and parameters of main	Not applicable
processing functions	(i.e. only types are defined here)



[BSW00359] Return type of callback functions	Not applicable
	(i.e. only types are defined here)
[BSW00360] Parameters of callback functions	Not applicable
	(i.e. only types are defined here)
[BSW00329] Avoidance of generic interfaces	Not applicable
	(i.e. only types are defined here)
[BSW00330] Usage of macros / inline functions	Not applicable
instead of functions	(i.e. only types are defined here)
[BSW00331] Separation of error and status	Not applicable
values	(i.e. only types are defined here)
[BSW009] Module User Documentation	Not applicable
	(i.e. only types are defined here)
[BSW00401] Documentation of multiple instances	Not applicable
of configuration parameters	(i.e. only types are defined here)
[BSW172] Compatibility and documentation of	Not applicable
scheduling strategy	(i.e. only types are defined here)
[BSW010] Memory resource documentation	Not applicable
	(i.e. only types are defined here)
[BSW00333] Documentation of callback function	Not applicable
context	(i.e. only types are defined here)
[BSW00374] Module vendor identification	Not applicable
	(i.e. only types are defined here)
[BSW00379] Module identification	Not applicable
	(i.e. only types are defined here)
[BSW003] Version identification	PLATFORM012
[BSW00318] Format of module version numbers	PLATFORM012
[BSW00321] Enumeration of module version	Not applicable
numbers	(i.e. this SWS does not specify a module)
[BSW00341] Microcontroller compatibility	Not applicable
documentation	(i.e. this SWS is not a module documentation)
[BSW00334] Provision of XML file	Not applicable
	(i.e. only types are defined here)

6.1 Linkage items for requirements management

This chapter contents several items which are only used in the requirement management tool. The items are necessary to build up the linkage between requirements, specification, etc. (e.g. for impact and coverage analyses).

Not applicable

For release versions, this chapter has to be removed.



7 Functional specification

7.1 General issues

PLATFORM001: For each platform an own platform types header file has to be provided.

PLATFORM031: If a specific compiler (not listed in this specification) requires a different mapping of ANSI C types to the AUTOSAR standard integer types, an own platform types header file for this compiler has to be provided.

PLATFORM003: The file name of the platform types header file shall be for all platforms 'Platform_Types.h'.

PLATFORM002: It is not allowed to add any extension to this file. Any extension invalidates the AUTOSAR conformity.

7.2 CPU Type

PLATFORM044: For each platform the register width of the CPU used shall be indicated by defining CPU_TYPE.

PLATFORM045: According to the register width of the CPU used, CPU_TYPE shall be assigned to one of the symbols CPU_TYPE_8, CPU_TYPE_16 or CPU_TYPE_32.

7.3 Endianess

The pattern for bit, byte and word ordering in native types, such as integers, is called endianess.

PLATFORM043: For each platform the appropriate bit order on register level shall be indicated in the platform types header file using the symbol CPU_BIT_ORDER.

PLATFORM046: For each platform the appropriate byte order on memory level shall be indicated in the platform types header file using the symbol CPU_BYTE_ORDER.

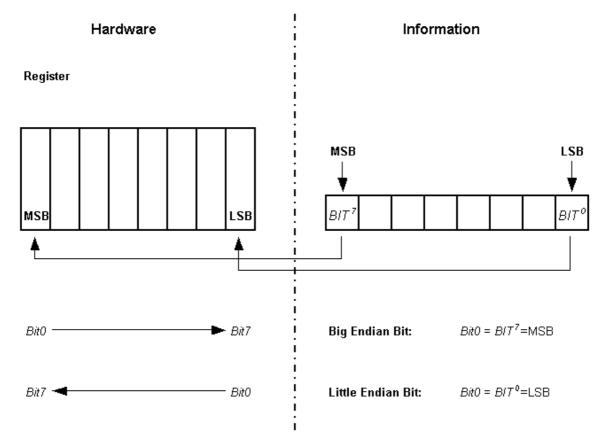
7.3.1 Bit Ordering (Register)

PLATFORM048: In case of big endian bit ordering CPU_BIT_ORDER shall be assigned to MSB_FIRST in the platform types header file.

PLATFORM049: In case of little endian bit ordering CPU_BIT_ORDER shall be assigned to LSB_FIRST in the platform types header file.



Illustrations:



Important Note:

The *naming* convention Bit0, Bit1, etc. and the bit's *significance* within a byte, word, etc. are different topics and shall not be mixed. The counting scheme of bits in Motorola μ C-architecture's (Big Endian Bit Order) starts with Bit0 indicating the Most Significant Bit, whereas all other μ C using Little Endian Bit Order assign Bit0 to be the Least Significant Bit!

The MSB in an accumulator is always stored as the left-most bit regardless of the CPU type. Hence, big and little endianess bit orders imply different bit-naming conventions.

7.3.2 Byte Ordering (Memory)

PLATFORM050: In case of big endian byte ordering CPU_BYTE_ORDER shall be assigned to HIGH_BYTE_FIRST in the platform types header file.

PLATFORM051: In case of little endian byte ordering CPU_BYTE_ORDER shall be assigned to LOW_BYTE_FIRST in the platform types header file.

Naming convention for illustration:

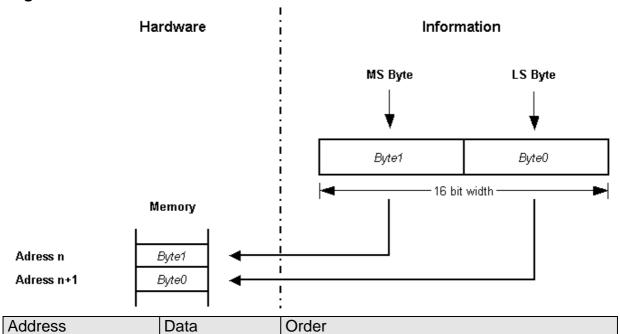
The Most Significant Byte within a 16 bit wide data is named Byte1. The Least Significant Byte within a 16 bit wide data is named Byte0.

Most Significant Byte (HIGH_BYTE_FIRST)

Least Significant Byte



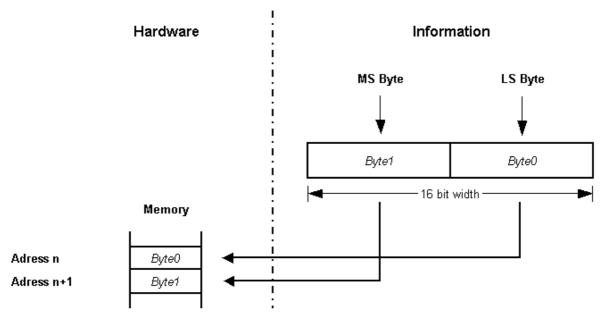
Big Endian(HIGH_BYTE_FIRST)



Little Endian (LOW_BYTE_FIRST)

Byte1

Byte0



Address	Data	Order
n	Byte0	Least Significant Byte (LOW_BYTE_FIRST)
n+1	Byte1	Most Significant Byte

Important Note:

The naming convention Byte0 and Byte1 is not unique and may be different in the manufacturer's reference documentation for a particular μ C.

n

n+1



7.4 Optimized integer data types

PLATFORM005: The optimized AUTOSAR integer data types (*_least) shall have at least the size given by the type name, but the types shall be implemented in a way that the best performance on the specific platform is achieved. 'Best performance' is defined in this context as 'least processor cycles for variable access as possible'. Example: on a TC1796, uint8_least is mapped to unsigned int (32 bit) because access to this type requires less processor cycles than e.g. unsigned char (8 bit).

PLATFORM032: The optimized AUTOSAR integer data types (*_least) shall only be used with a local scope inside a module. They are not allowed to be used within the API of a module.

PLATFORM033: Operations on the optimized AUTOSAR integer data types (*_least) shall not expect a specific size of this type. The size specified by the name is guaranteed, but can be larger. It is not allowed to use rollover mechanisms during counting and shifting.

Examples of usage:

- Loop counters (e.g. maximum loop count = 124 → use uint8_least)
- Switch case arguments (e.g. maximum number of states = 17 → use uint8_least)

7.5 boolean data type

PLATFORM027: The standard AUTOSAR type boolean shall be implemented on basis of an eight bits long unsigned integer.

PLATFORM034: The standard AUTOSAR type boolean shall only be used in conjunction with the standard symbols TRUE and FALSE. For value assignments of variables of type boolean no arithmetic or logical operators $(+, ++, -, --, *, /, \cdot, <<, >>, !, ~)$ must be used. The only allowed form of assignment is

```
boolean var;
...
var = TRUE;
var = FALSE;

The only allowed forms of comparison are boolean var;
...
if (var == TRUE)
if (var == FALSE)
if (var != TRUE)
if (var != TRUE)
```



8 API specification

8.1 Imported types

Not applicable.

8.2 Type definitions

Type definitions.PLATFORM061: Concerning the signed integer types, AUTOSAR supports for compiler and target implementation only 2 complement arithmetic. This directly impacts the chosen ranges for these types.

8.2.1 boolean

Type:	Unsigned integer
Range:	0 FALSE
	1 TRUE
Description:	PLATFORM026: This standard AUTOSAR type shall only be used together with the definitions TRUE and FALSE. See PLATFORM027 for implementation and usage.
	PLATFORM060: The boolean type shall always be mapped to a platform specific type where pointers can be applied to to enable a passing of parameters via API. There are specific BIT types of some HW platforms which are very efficient but where no pointers can point to.

8.2.2 uint8

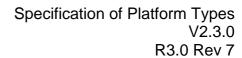
Type:	Unsigned integer
Range:	0255 8 bit 0x000xFF
Description:	PLATFORM013: This standard AUTOSAR type shall be of 8 bit unsigned.

8.2.3 uint16

Type:	Unsigned integer
Range:	065535 16 bit 0x00000xFFFF
Description:	PLATFORM014: This standard AUTOSAR type shall be of 16 bit unsigned.

8.2.4 uint32

Type:	Unsigned integer
Range:	04294967295 32 bit 0x000000000xFFFFFFF
Description:	PLATFORM015: This standard AUTOSAR type shall be 32 bit unsigned.







8.2.5 sint8

Type:	Signed integer
Range:	-128+127 7 bit + 1 bit sign 0x800x7F
Description:	PLATFORM016: This standard AUTOSAR type shall be 8 bit signed.

8.2.6 sint16

Type:	Signed integer
Range:	-32768+32767 15 bit + 1 bit sign 0x80000x7FFF
Description:	PLATFORM017: This standard AUTOSAR type shall be 16 bit signed.

8.2.7 sint32

Type:	Signed integer
Range:	-2147483648 +2147483647 31 bit + 1 bit sign 0x800000000x7FFFFFFF
Description:	PLATFORM018:

8.2.8 uint8_least

Туре:	Unsigned integer
Range:	At least 0255 At least 8 bit
Description:	PLATFORM020: This optimized AUTOSAR type shall be at least of 8 bit
-	unsigned. See PLATFORM005 for implementation and usage.

8.2.9 uint16_least

Туре:	Unsigned integer
Range:	At least 065535 At least 16 bit
Description:	PLATFORM021: This standard AUTOSAR type shall be at least 16 bit unsigned.
•	See PLATFORM005 for implementation and usage.

8.2.10 uint32_least

Туре:	Unsigned integer
Range:	At least 04294967295 At least 32 bit
Description:	PLATFORM022. See PLATFORM005 for implementation and usage.



8.2.11 sint8_least

Type:	Signed integer
Range:	At least -128+127 At least 7 bit + 1 bit sign
Description:	PLATFORM023. See PLATFORM005 for implementation and usage.

8.2.12 sint16_least

Туре:	Signed integer	
Range:	At least -32768+32767 At least 15 bit + 1 bit sign	
Description:	PLATFORM024. See PLATFORM005 for implementation and usage.	

8.2.13 sint32_least

Type:	Signed integer
	At least -2147483648 At least 31 bit + 1 bit sign +2147483647
Description:	PLATFORM025 See PLATFORM005 for implementation and usage.

8.2.14 float32

Type:	Float
Range:	- 32 bit
Description:	PLATFORM041

8.2.15 float64

Туре:	Double
Range:	- 64 bit
Description:	PLATFORM042:



8.3 Symbol definitions

8.3.1 CPU_TYPE

Symbol	CPU_TYPE	
Range	CPU_TYPE_8	Indicating a 8 bit processor
	CPU_TYPE_16	Indicating a 16 bit processor
	CPU_TYPE_32	Indicating a 32 bit processor
Description:	This symbol shall be defined as #define having one of the values CPU_TYPE_8, CPU_TYPE_16 or CPU_TYPE_32 according to the platform.	

8.3.2 CPU_BIT_ORDER

Symbol	CPU_BIT_ORDER	
Range	MSB_FIRST The most significant bit is the first bit of the bit field	
	LSB_FIRST	The least significant bit is the first bit of the bit field
Description:	PLATFORM038: MSB_FIRST_LSB_FIRST	

8.3.3 CPU_BYTE_ORDER

Symbol	CPU_BYTE_ORDER		
Range	HIGH_BYTE_FIRST	Within a uint16, the high byte is located before the low byte.	
	LOW_BYTE_FIRST	OW_BYTE_FIRST Within uint16, the low byte is located before the hig	
		byte.	
Description:	PLATFORM039: This symbol shall be defined as #define having one of the		
	values HIGH_BYTE_FIRST or LOW_BYTE_FIRST according to the platform.		

8.3.4 TRUE, FALSE

Symbol/Value:	TRUE	1
Symbol/Value:	FALSE	0
Description:	shall be avoided using a c	of in-built compiler support of the symbols, redefinitions onditional check. bols TRUE and FALSE shall be defined as follows:
	<pre>#ifndef TRUE #define TRUE 1 #endif</pre>	
	#ifndef FALSE #define FALSE 0 #endif PLATFORM055: These boolean type defined in l	symbols shall only be used in conjunction with the Platform_Types.h.



8.4 Function definitions

Not applicable.

8.5 Call-back notifications

Not applicable.

8.6 Scheduled functions

Not applicable.

8.7 Expected Interfaces

Not applicable.



9 Sequence diagrams

Not applicable.



10 Configuration specification

10.1 Published parameters

Published information contains data defined by the implementer of the SW module that does not change when the module is adapted (i.e. configured) to the actual HW/SW environment. It thus contains version and manufacturer information.

The standard common published information like

```
vendorld (PLATFORM_VENDOR_ID), moduleld (PLATFORM_MODULE_ID), arMajorVersion (PLATFORM_AR_MAJOR_VERSION), arMinorVersion (PLATFORM_AR_MINOR_VERSION), arPatchVersion (PLATFORM_AR_PATCH_VERSION), swMajorVersion (PLATFORM_SW_MAJOR_VERSION), swMinorVersion (PLATFORM_SW_MINOR_VERSION), swPatchVersion (PLATFORM_SW_PATCH_VERSION), vendorApiInfix (PLATFORM_VENDOR_API_INFIX)
```

is provided in the BSW Module Description Template (see [2] Figure 4.1 and Figure 7.1).

Additional published parameters are listed below if applicable for this module.



11 Changes to Release 1

11.1 Deleted SWS Items

PLATFORM052 PLATFORM053 PLATFORM040 PLATFORM047

11.2 Replaced SWS Items

Not applicable

11.3 Changed SWS Items

PLATFORM012

11.4 Added SWS Items

Not applicable



12 Annex

12.1 Type definitions – general

PLATFORM057: The platform type files for all platforms shall contain the following symbols:

```
#define CPU_TYPE_8 8
#define CPU_TYPE_16 16
#define CPU_TYPE_32 32
#define MSB_FIRST 0
#define LSB_FIRST 1
#define HIGH_BYTE_FIRST 0
#define LOW_BYTE_FIRST 1
```

12.2 Type definitions - S12X

PLATFORM006: The platform types for Freescale S12X shall have the following mapping to the ANSI C types:

#define	CPU_TYPE CPU_BIT_ORDER CPU_BYTE_ORDER	CPU_TYPE_16 LSB_FIRST HIGH_BYTE_FIRST
Types: typedef	unsigned char	boolean;
typedef typedef typedef typedef	signed char unsigned char signed short unsigned short signed long unsigned long	<pre>sint8; uint8; sint16; uint16; sint32; uint32;</pre>
typedef typedef typedef typedef typedef	signed char unsigned char signed short unsigned short signed long unsigned long	<pre>sint8_least; uint8_least; sint16_least; uint16_least; sint32_least; uint32_least;</pre>
typedef typedef		float32; float64;



12.3 Type definitions – ST10

PLATFORM007: The platform types for ST Microelectronics ST10 shall have the following mapping to the ANSI C types:

Symbols:

```
#define CPU_TYPE
                           CPU_TYPE_16
#define CPU_BIT_ORDER
                           LSB_FIRST
#define CPU BYTE ORDER
                           LOW BYTE FIRST
Types:
typedef unsigned char
                           boolean;
typedef signed char
                           sint8;
typedef unsigned char
                           uint8;
typedef signed short
                           sint16;
typedef unsigned short
                           uint16;
typedef signed long
                           sint32;
                           uint32;
typedef unsigned long
typedef unsigned short
                           uint8 least;
typedef unsigned short
                           uint16_least;
typedef unsigned long
                           uint32_least;
typedef signed short
                           sint8_least;
typedef signed short
                           sint16_least;
typedef signed long
                           sint32_least;
typedef float
                           float32;
typedef double
                           float64;
```

12.4 Type definitions - ST30

PLATFORM008: The platform types for STMicroelectronics ST30 shall have the following mapping to the ANSI C types:

Symbols:

30 of 34

#define	CPU_TYPE CPU_BIT_ORDER CPU_BYTE_ORDER	CPU_TYPE_32 LSB_FIRST LOW_BYTE_FIRST
Types: typedef	unsigned char	boolean;
typedef typedef typedef typedef	signed char unsigned char signed short unsigned short signed long unsigned long	<pre>sint8; uint8; sint16; uint16; sint32; uint32;</pre>



```
typedef unsigned long
                          uint8 least;
typedef unsigned long
                          uint16_least;
typedef unsigned long
                          uint32_least;
typedef signed long
                          sint8_least;
typedef signed long
                          sint16 least;
typedef signed long
                          sint32 least;
typedef float
                          float32;
typedef double
                          float64;
```

12.5 Type definitions – V850

PLATFORM009: The platform types for NEC V850 shall have the following mapping to the ANSI C types:

Symbols:

```
#define CPU_TYPE
                           CPU_TYPE_32
#define CPU_BIT_ORDER
                           LSB FIRST
                           LOW_BYTE_FIRST
#define CPU_BYTE_ORDER
Types:
typedef unsigned char
                          boolean;
typedef signed char
                          sint8;
typedef unsigned char
                          uint8;
                          sint16;
typedef signed short
typedef unsigned short
                         uint16;
typedef signed long
                           sint32;
typedef unsigned long
                          uint32;
typedef unsigned long
                          uint8 least;
typedef unsigned long
                          uint16 least;
typedef unsigned long
                          uint32_least;
typedef signed long
                          sint8 least;
typedef signed long
                          sint16_least;
typedef signed long
                          sint32_least;
typedef float
                          float32;
typedef double
                          float64;
```

12.6 Type definitions - MPC5554

PLATFORM010: The platform types for Freescale MPC5554 shall have the following mapping to the ANSI C types:

#define	CPU_TYPE	CPU_TYPE_32
#define	CPU_BIT_ORDER	MSB_FIRST



```
#define CPU_BYTE_ORDER
                           HIGH_BYTE_FIRST
Types:
typedef unsigned char
                            boolean;
typedef signed char
                           sint8;
typedef unsigned char
                           uint8;
typedef signed short
                           sint16;
typedef unsigned short
                           uint16;
typedef signed long
                           sint32;
typedef unsigned long
                           uint32;
typedef unsigned long
                           uint8 least;
typedef unsigned long typedef unsigned long
                           uint16 least;
                          uint32_least;
typedef signed long
                          sint8_least;
                          sint16_least;
typedef signed long
typedef signed long
                           sint32_least;
typedef float
                           float32;
typedef double
                           float64;
```

12.7 Type definitions – TC1796/TC1766

PLATFORM011: The platform types for Infineon TC1796/TC1766 shall have the following mapping to the ANSI C types:

```
#define CPU_TYPE
                            CPU TYPE 32
#define CPU_BIT_ORDER
                           LSB FIRST
#define CPU_BYTE_ORDER
                           LOW_BYTE_FIRST
Types:
typedef unsigned char
                           boolean;
typedef signed char
                            sint8;
typedef unsigned char
                           uint8;
typedef signed short
                           sint16;
typedef unsigned short
                          uint16;
typedef signed long
                           sint32;
typedef unsigned long
                           uint32;
typedef unsigned long
                           uint8 least;
typedef unsigned long
                           uint16_least;
typedef unsigned long typedef signed long
                           uint32 least;
                          sint8_least;
typedef signed long
                          sint16_least;
typedef signed long
                           sint32_least;
typedef float
                            float32;
```



typedef double float64;

12.8 Type definitions – MB91F

PLATFORM019: The platform types for Fujitsu MB91F shall have the following mapping to the ANSI C types:

Symbols:

```
#define CPU_TYPE
                         CPU_TYPE_32
#define CPU_BIT_ORDER
                        LSB_FIRST
#define CPU_BYTE_ORDER
                         HIGH_BYTE_FIRST
Types:
typedef unsigned char
                         boolean;
typedef signed char
                         sint8;
typedef unsigned char
                         uint8;
typedef signed short
                         sint16;
                       uint16;
typedef unsigned short
typedef signed long
                         sint32;
typedef unsigned long
                         uint32;
typedef unsigned long
typedef signed long
                         uint32_least;
                       sint8_least;
typedef signed long
                        sint16_least;
typedef signed long
                         sint32 least;
typedef float
                         float32;
typedef double
                         float64;
```

12.9 Type definitions - M16C/M32C

PLATFORM058: The platform types for Renesas M16C and M32C shall have the following mapping to the ANSI C types:

#define	CPU_TYPE CPU_BIT_ORDER CPU_BYTE_ORDER	CPU_TYPE_16 LSB_FIRST LOW_BYTE_FIRST
Types: typedef	unsigned char	boolean;
typedef	signed char unsigned char signed short	<pre>sint8; uint8; sint16;</pre>



```
typedef unsigned short
                           uint16;
typedef signed long
                            sint32;
typedef unsigned long
                           uint32;
typedef unsigned short
                           uint8_least;
typedef unsigned short
                           uint16 least;
typedef unsigned long
                           uint32 least;
typedef signed short
                           sint8 least;
typedef signed short
                           sint16_least;
typedef signed long
                            sint32_least;
typedef float
                            float32;
                           float64;
typedef double
```

12.10Type definitions – SHx

PLATFORM059: The platform types for Renesas SHx shall have the following mapping to the ANSI C types:

```
#define CPU_TYPE
                            CPU TYPE 32
#define CPU_BIT_ORDER
                           LSB_FIRST
#define CPU_BYTE_ORDER
                           HIGH_BYTE_FIRST
Types:
typedef unsigned char
                           boolean;
typedef signed char
                            sint8;
typedef unsigned char
                           uint8;
typedef signed short
                            sint16;
typedef unsigned short
                           uint16;
typedef signed int
                            sint32;
typedef unsigned int
                           uint32;
                           uint8_least;
typedef unsigned long
typedef unsigned long
                           uint16_least;
typedef unsigned long
                           uint32_least;
typedef signed long
                           sint8_least;
typedef signed long
                           sint16 least;
                           sint32 least;
typedef signed long
typedef float
                           float32;
typedef double
                           float64;
```