User Manual

for S32K1XX BASE Driver

Rev. 1.0 — 11 January 2022 User manual

UM2BASEASR4.2 Rev0002R1.0.5

Revision History

Revision	Date	Author	Description
1.0	11/01/2022	NXP MCAL Team	Updated version for ASR 4.2.2 S32K1XX R1.0.5

1 Introduction

This User Manual describes NXP Semiconductors AUTOSAR Base (BASE) for S32K1XX .

AUTOSAR BASE driver configuration parameters and deviations from the specification are described in BASE Driver chapter of this document. AUTOSAR BASE driver requirements and APIs are described in the AUTOSAR BASE driver software specification document.

1.1 Supported Derivatives

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

Table 1. S32K1XX Derivatives

NXP Semiconductors	s32k142_lqfp100, s32k142_lqfp64, s32k142_lqfp48, s32k144_lqfp100, s32k144_mapbga100, s32k144_lqfp64, s32k144_lqfp48, s32k146_lqfp144, s32k146_lqfp100, s32k146_mapbga100, s32k146_lqfp64, s32k148_lqfp176, s32k148_lqfp144, s32k148_lqfp100, s32k148_mapbga100, s32k118_lqfp48, s32k118_lqfp64, s32k116_lqfp48, s32k116_qfn32, s32k144w_lqfp48, s32k144w_lqfp64, s32k142w_lqfp64, s32k142w_lqfp48

All of the above microcontroller devices are collectively named as S32K1XX .

1.2 Overview

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

AUTOSAR

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

1.3 About this Manual

This Technical Reference employs the following typographical conventions:

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

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

Notes and warnings are shown as below:

Note: *This is a note.*

1.4 Acronyms and Definitions

Table 2. Acronyms and Definitions

Term	Definition
API	Application Programming Interface
ASM	Assembler Language
AUTOSAR	Automotive Open System Architecture
BSMI	Basic Software Make file Interface
C/CPP	C and C++ Source Code
DEM	Diagnostic Event Manager
DET	Development Error Tracer
N/A	Not Applicable
MCU	Micro Controller Unit
VLE	Variable Length Encoding

1.5 Reference List

Table 3. Reference List

#	Title	Version
1	General Specification of Basic Software Modules	AUTOSAR Release 4.2.2
2	Specification of Communication Stack Types	AUTOSAR Release 4.2.2
3	Specification of Compiler Abstraction	AUTOSAR Release 4.2.2
4	Specification of Platform Types	AUTOSAR Release 4.2.2
5	Specification of Standard Types	AUTOSAR Release 4.2.2
6	S32K1xx Series Reference Manual	Rev. 13, 04/2020
7	S32K1xx Data Sheet	Rev. 13, 04/2020
8	S32K142 Mask Set Errata for Mask 0N33V (0N33V)	Rev. 20/APR/2020
9	S32K144 Mask Set Errata for Mask 0N57U (0N57U)	Rev. 20/APR/2020
10	S32K146 Mask Set Errata for Mask 0N73V (0N73V)	Rev. 20/APR/2020
11	S32K148 Mask Set Errata for Mask 0N20V (0N20V)	Rev. 20/APR/2020
12	S32K118 Mask Set Errata for Mask 0N97V (0N97V)	Rev. 20/APR/2020

Table 3. Reference List*...continued*

#	Title	Version
13	S32K116 Mask Set Errata for Mask 0N96V (0N96V)	Rev. 20/APR/2020
14	S32K144W Mask Set Errata for Mask 0P64A (0P64A)	Rev. 14 FEB 2020

2 Driver

2.1 Requirements

BASE is an custom module, so AUTOSAR only specifies some guidelines for the design and configuration. Other details for this module can be found in EB tresos Studio developer's guide. This module contains stubs from several AutoSAR components. The requirements used for the files present in this module are available in the Software Specification documents from Table Section 1.5 .

2.2 Driver Design Summary

The BASE module contains the common files/definitions needed by the MCAL. This means that it is a dependency for all other MCAL modules.

The BASE module consists from a list of C header files that can be split into 3 categories:

- AutoSAR required files (that AutoSAR specifies and must not be modified)
- Stubs - files that are required by AutoSAR but are provided as examples in the NXP Semiconductors S32K1XX MCAL release. They must be re-written by the integrator.
- Files that are required by the NXP Semiconductors S32K1XX MCAL and must not be modified.

Below you can find the descriptions for each file present in the BASE module:

Table 4. Description of files inside the BASE module

File Name	File Type	Description
Can_ GeneralTypes.h	Stub file. Must be replaced by all integrators.	This file is a stub. Its name and content is specified by AutoSAR but in the NXP Semiconductors S32K1XX MCAL release it contains only the defines/typedefs/constants that are needed by the MCAL drivers. Note: The following files need to be included prior to include Can_GeneralTypes.h - ComStack_Cfg.h and Can_Cfg.h
Compiler.h	AutoSAR specified file - must not be modified.	This is a file with content fully defined by the AutoSAR standard. AutoSAR requires that no modification must be done to the contents of this file. During integration this file can be overwritten with another one with the same C content. The NXP Semiconductors S32K1XX MCAL release provides this file and can be used as-is.
Compiler_ Cfg.h	Stub file. Must be replaced by all integrators.	This file is a stub. Its name and content is specified by AutoSAR but in the NXP Semiconductors S32K1XX MCAL release it contains only the defines that are needed by the MCAL drivers. This file defines the compiler memory and pointer classes to be used for MCAL. The value of the defines must be set by each integrator.

Table 4. Description of files inside the BASE module*...continued*

File Name	File Type	Description
ComStack_ Cfg.h	Stub file. Must be replaced by all integrators.	This file is a stub. Its name and content is specified by AutoSAR but in the NXP Semiconductors S32K1XX MCAL release it contains only the defines/typedefs/constants that are needed by the MCAL drivers.
ComStack_ Types.h	Stub file. Must be replaced by all integrators.	This file is a stub. Its name and content is specified by AutoSAR but in the NXP Semiconductors S32K1XX MCAL release it contains only the defines/typedefs/constants that are needed by the MCAL drivers.
Eth_General Types.h	Stub file. Must be replaced by all integrators.	This file is a stub. Its name and content is specified by AutoSAR but in the NXP Semiconductors S32K1XX MCAL release it contains only the defines/typedefs/constants that are needed by the MCAL drivers.
Fr_General Types.h	Stub file. Must be replaced by all integrators.	This file is a stub. Its name and content is specified by AutoSAR but in the NXP Semiconductors S32K1XX MCAL release it contains only the defines/typedefs/constants that are needed by the MCAL drivers.
Lin_General Types.h	Stub file. Must be replaced by all integrators.	This file is a stub. Its name and content is specified by AutoSAR but in the NXP Semiconductors S32K1XX MCAL release it contains only the defines/typedefs/constants that are needed by the MCAL drivers.
Mcal.h	MCAL specific file.	This is a file that specific to NXP Semiconductors S32K1XX MCAL release. It contains defines and macros needed by MCAL drivers. It contains several macros defined for every compiler supported by MCAL (but not all compilers are available for all releases - for a list of compilers supported by this release please check the release note document). If no operating system is used, the following 4 macros can be overwritten by the integrators depending on their environment: • ISR • EXIT_INTERRUPT • SuspendAllInterrupts • ResumeAllInterrupts If the integrated project uses an AutoSAR operating system, this file must be used as-is.
MemMap.h	Stub file. Must be replaced by all integrators.	This file is a stub. Its name and content is specified by AutoSAR but in the NXP Semiconductors S32K1XX MCAL release it contains only the defines/typedefs/constants that are needed by the MCAL drivers. This file contains the memory mapping instructions/pragmas needed for every memory section from the MCAL code. The default content of this file only renames some sections and has the pragmas to clearly mark the RAM code sections. Depending on the integrating environment, this entire file must be updated.

Table 4. Description of files inside the BASE module*...continued*

File Name	File Type	Description
Platform_ Types.h	AutoSAR specified file - must not be modified.	This is a file with content fully defined by the AutoSAR standard. AutoSAR requires that no modification must be done to the contents of this file. During integration this file can be overwritten with another one with the same C content. The NXP Semiconductors S32K1XX MCAL release provides this file and can be used as-is.
RegLock Macros.h	MCAL specific file - to be used as-is.	This is a file that specific to NXP Semiconductors S32K1XX MCAL release. It contains defines needed by MCAL drivers.
Reg_eSys.h	MCAL specific file - to be used as-is.	This is a file that specific to NXP Semiconductors S32K1XX MCAL release. It contains defines needed by MCAL drivers.
SilReg Macros.h	MCAL specific file - to be used as-is.	This is a file that specific to NXP Semiconductors S32K1XX MCAL release. It contains defines and macros needed by MCAL drivers.
Soc_Ips.h	MCAL specific file - to be used as-is.	This is a file that specific to NXP Semiconductors S32K1XX MCAL release. It contains defines and macros needed by MCAL drivers.
StdReg Macros.h	MCAL specific file - to be used as-is.	This is a file that specific to NXP Semiconductors S32K1XX MCAL release. It contains defines and macros needed by MCAL drivers.
Std_Types.h	AutoSAR specified file - must not be modified.	This is a file with content fully defined by the AutoSAR standard. AutoSAR requires that no modification must be done to the contents of this file. During integration this file can be overwritten with another one with the same C content. The NXP Semiconductors S32K1XX MCAL release provides this file and can be used as-is.
modules.h	MCAL specific file - to be used as-is.	This is a file that is generated by Base plugin and contains defines needed by MCAL drivers.

2.3 Hardware Resources

None.

2.4 Deviation from Requirements

Since this is a custom module, it contains files from several AutoSAR components. The AUTOSAR provides some guidelines for design and configuration the BASE Module. The BASE module deviates from the AUTOSAR software specification documents from Reference List mainly for the files provided as stubs in the current release.

There are also some additional requirements (on top of requirements detailed in AUTOSAR software specification documents from Reference List) which need to be satisfied for correct operation.

Table 5. Deviations Status Column Description

Term	Definition
N/S	Out of scope
N/I	Not implemented
N/F	Not fully implemented

Below table identifies the AUTOSAR requirements that are not fully implemented, implemented differently, or out of scope for the module.

Table 6. Driver Deviations Table

Requirement	Status	Description	Notes
N/A	N/A	N/A	N/A

2.5 Driver limitations

None

2.6 Driver usage and configuration tips

None

2.7 Runtime Errors

The module does not generate any DEM errors at runtime.

Table 7. Runtime Errors

Function	Error Code	Condition triggering the error
N/A	N/A	N/A

2.8 Software specification

The following sections contains driver software specifications.

2.8.1 Define Reference

Constants supported by the driver are as per AUTOSAR BASE Driver software specification Version 4.2 Rev0002 .

2.8.1.1 Define COMPILER_VENDOR_ID

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

Implements : DBASE03023

Table 8. Define COMPILER_VENDOR_ID Description

Name	COMPILER_VENDOR_ID
Initializer	43

2.8.1.2 Define COMPILER_AR_RELEASE_MAJOR_VERSION

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

Implements : DBASE03023

Table 9. Define COMPILER_AR_RELEASE_MAJOR_VERSION Description

Name	COMPILER_AR_RELEASE_MAJOR_VERSION
Initializer	4

2.8.1.3 Define COMPILER_AR_RELEASE_MINOR_VERSION

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

Implements : DBASE03023

Table 10. Define COMPILER_AR_RELEASE_MINOR_VERSION Description

Name	COMPILER_AR_RELEASE_MINOR_VERSION
Initializer	2

2.8.1.4 Define COMPILER_AR_RELEASE_REVISION_VERSION

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

Implements : DBASE03023

Table 11. Define COMPILER_AR_RELEASE_REVISION_VERSION Description

Name	COMPILER_AR_RELEASE_REVISION_VERSION
Initializer	2

2.8.1.5 Define COMPILER_SW_MAJOR_VERSION

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

Implements : DBASE03023

Table 12. Define COMPILER_SW_MAJOR_VERSION Description

Name	COMPILER_SW_MAJOR_VERSION
Initializer	1

2.8.1.6 Define COMPILER_SW_MINOR_VERSION

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

Implements : DBASE03023

Table 13. Define COMPILER_SW_MINOR_VERSION Description

Name	COMPILER_SW_MINOR_VERSION
Initializer	0

2.8.1.7 Define COMPILER_SW_PATCH_VERSION

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

Implements : DBASE03023

Table 14. Define COMPILER_SW_PATCH_VERSION Description

Name	COMPILER_SW_PATCH_VERSION
Initializer	5

2.8.1.8 Define AUTOMATIC

The memory class AUTOMATIC shall be provided as empty definition, used for the declaration of local pointers.

Implements : DBASE03004

Table 15. Define AUTOMATIC Description

Name	AUTOMATIC
Initializer	

2.8.1.9 Define CONST

The compiler abstraction shall define the CONST macro for the declaration and definition of constants.

Implements : DBASE03012

Table 16. Define CONST Description

Name	CONST
Initializer	const consttype

2.8.1.10 Define CONSTP2CONST

The compiler abstraction shall define the CONSTP2CONST macro for the declaration and definition of constant pointers accessing constants.

Implements : DBASE03013

Table 17. Define CONSTP2CONST Description

Name	CONSTP2CONST
Initializer	const ptrtype * const

2.8.1.11 Define CONSTP2VAR

The compiler abstraction shall define the CONSTP2VAR macro for the declaration and definition of constant pointers accessing variables.

Implements : DBASE03014

Table 18. Define CONSTP2VAR Description

Name	CONSTP2VAR
Initializer	ptrtype * const

2.8.1.12 Define FUNC

The compiler abstraction shall define the FUNC macro for the declaration and definition of functions, that ensures correct syntax of function declarations as required by a specific compiler.

Implements : DBASE03015

Table 19. Define FUNC Description

Name	FUNC

Table 19. Define FUNC Description*...continued*

Initializer	rettype

2.8.1.13 Define NULL_PTR

The compiler abstraction shall provide the NULL_PTR define with a void pointer to zero definition.

Implements : DBASE03009

Table 20. Define NULL_PTR Description

Name	NULL_PTR
Initializer	((void *)0)

2.8.1.14 Define P2CONST

The compiler abstraction shall define the P2CONST macro for the declaration and definition of pointers in RAM pointing to constants.

Implements : DBASE03017

Table 21. Define P2CONST Description

Name	P2CONST
Initializer	const ptrtype *

2.8.1.15 Define P2FUNC

The compiler abstraction shall define the P2FUNC macro for the type definition of pointers to functions.

Implements : DBASE03018

Table 22. Define P2FUNC Description

Name	P2FUNC
Initializer	rettype (*fctname)

2.8.1.16 Define P2VAR

The compiler abstraction shall define the P2VAR macro for the declaration and definition of pointers in RAM, pointing to variables.

Implements : DBASE03019

Table 23. Define P2VAR Description

Name	P2VAR
Initializer	ptrtype *

2.8.1.17 Define TYPEDEF

The memory class TYPEDEF shall be provided as empty definition. This memory class shall be used within type definitions, where no memory qualifier can be specified. This can be necessary for defining pointer types, with e.g. P2VAR, where the macros require two parameters. First parameter can be specified in the type definition (distance to the memory location referenced by the pointer), but the second one (memory allocation of the pointer itself) cannot be defined at this time. Hence memory class TYPEDEF shall be applied.

Implements : DBASE03011

Table 24. Define TYPEDEF Description

Name	TYPEDEF
Initializer	

2.8.1.18 Define VAR

The compiler abstraction shall define the VAR macro for the declaration and definition of variables.

Implements : DBASE03022

Table 25. Define VAR Description

Name	VAR
Initializer	vartype

2.8.1.19 Define ADC_CODE

ADC memory and pointer classes.

Implements : DBASE04001

Table 26. Define ADC_CODE Description

Name	ADC_CODE
Initializer	

2.8.1.20 Define ADC_CONST

ADC memory and pointer classes.

Implements : DBASE04001

Table 27. Define ADC_CONST Description

Name	ADC_CONST
Initializer	

2.8.1.21 Define ADC_APPL_DATA

ADC memory and pointer classes.

Implements : DBASE04001

Table 28. Define ADC_APPL_DATA Description

Name	ADC_APPL_DATA
Initializer	

2.8.1.22 Define ADC_APPL_CONST

ADC memory and pointer classes.

Implements : DBASE04001

Table 29. Define ADC_APPL_CONST Description

Name	ADC_APPL_CONST
Initializer	

2.8.1.23 Define ADC_APPL_CODE

ADC memory and pointer classes.

Implements : DBASE04001

Table 30. Define ADC_APPL_CODE Description

Name	ADC_APPL_CODE
Initializer	

2.8.1.24 Define ADC_CALLOUT_CODE

ADC memory and pointer classes.

Implements : DBASE04001

Table 31. Define ADC_CALLOUT_CODE Description

Name	ADC_CALLOUT_CODE
Initializer	

2.8.1.25 Define ADC_VAR_NOINIT

ADC memory and pointer classes.

Implements : DBASE04001

Table 32. Define ADC_VAR_NOINIT Description

Name	ADC_VAR_NOINIT
Initializer	

2.8.1.26 Define ADC_VAR_POWER_ON_INIT

ADC memory and pointer classes.

Implements : DBASE04001

Table 33. Define ADC_VAR_POWER_ON_INIT Description

Name	ADC_VAR_POWER_ON_INIT
Initializer	

2.8.1.27 Define ADC_VAR_FAST

ADC memory and pointer classes.

Implements : DBASE04001

Table 34. Define ADC_VAR_FAST Description

Name	ADC_VAR_FAST
Initializer	

2.8.1.28 Define ADC_VAR

ADC memory and pointer classes.

Implements : DBASE04001

Table 35. Define ADC_VAR Description

Name	ADC_VAR
Initializer	

2.8.1.29 Define CAN_CODE

CAN memory and pointer classes.

Implements : DBASE04001

Table 36. Define CAN_CODE Description

Name	CAN_CODE
Initializer	

2.8.1.30 Define CAN_CONST

CAN memory and pointer classes.

Implements : DBASE04001

Table 37. Define CAN_CONST Description

Name	CAN_CONST
Initializer	

2.8.1.31 Define CAN_APPL_DATA

CAN memory and pointer classes.

Implements : DBASE04001

Table 38. Define CAN_APPL_DATA Description

Name	CAN_APPL_DATA
Initializer	

2.8.1.32 Define CAN_APPL_CONST

CAN memory and pointer classes.

Implements : DBASE04001

Table 39. Define CAN_APPL_CONST Description

Name	CAN_APPL_CONST
Initializer	

2.8.1.33 Define CAN_APPL_CODE

CAN memory and pointer classes.

Implements : DBASE04001

Table 40. Define CAN_APPL_CODE Description

Name	CAN_APPL_CODE
Initializer	

2.8.1.34 Define CAN_CALLOUT_CODE

CAN memory and pointer classes.

Implements : DBASE04001

Table 41. Define CAN_CALLOUT_CODE Description

Name	CAN_CALLOUT_CODE

Table 41. Define CAN_CALLOUT_CODE Description*...continued*

Initializer	

2.8.1.35 Define CAN_VAR_NOINIT

CAN memory and pointer classes.

Implements : DBASE04001

Table 42. Define CAN_VAR_NOINIT Description

Name	CAN_VAR_NOINIT
Initializer	

2.8.1.36 Define CAN_VAR_POWER_ON_INIT

CAN memory and pointer classes.

Implements : DBASE04001

Table 43. Define CAN_VAR_POWER_ON_INIT Description

Name	CAN_VAR_POWER_ON_INIT
Initializer	

2.8.1.37 Define CAN_VAR_FAST

CAN memory and pointer classes.

Implements : DBASE04001

Table 44. Define CAN_VAR_FAST Description

Name	CAN_VAR_FAST
Initializer	

2.8.1.38 Define CAN_VAR

CAN memory and pointer classes.

Implements : DBASE04001

Table 45. Define CAN_VAR Description

Name	CAN_VAR
Initializer	

2.8.1.39 Define CRCU_CODE

CRCU memory and pointer classes.

Implements :

Table 46. Define CRCU_CODE Description

Name	CRCU_CODE
Initializer	

2.8.1.40 Define CRCU_CONST

CRCU memory and pointer classes.

Implements :

Table 47. Define CRCU_CONST Description

Name	CRCU_CONST
Initializer	

2.8.1.41 Define CRCU_APPL_DATA

CRCU memory and pointer classes.

Implements :

Table 48. Define CRCU_APPL_DATA Description

Name	CRCU_APPL_DATA
Initializer	

2.8.1.42 Define CRCU_APPL_CONST

CRCU memory and pointer classes.

Implements :

Table 49. Define CRCU_APPL_CONST Description

Name	CRCU_APPL_CONST
Initializer	

2.8.1.43 Define CRCU_APPL_CODE

CRCU memory and pointer classes.

Implements : DBASE04001

Table 50. Define CRCU_APPL_CODE Description

Name	CRCU_APPL_CODE
Initializer	

2.8.1.44 Define CRCU_CALLOUT_CODE

CRCU memory and pointer classes.

Implements :

Table 51. Define CRCU_CALLOUT_CODE Description

Name	CRCU_CALLOUT_CODE
Initializer	

2.8.1.45 Define CRCU_VAR_NOINIT

CRCU memory and pointer classes.

Implements :

Table 52. Define CRCU_VAR_NOINIT Description

Name	CRCU_VAR_NOINIT
Initializer	

2.8.1.46 Define CRCU_VAR_POWER_ON_INIT

CRCU memory and pointer classes.

Implements :

Table 53. Define CRCU_VAR_POWER_ON_INIT Description

Name	CRCU_VAR_POWER_ON_INIT
Initializer	

2.8.1.47 Define CRCU_VAR_FAST

CRCU memory and pointer classes.

Implements :

Table 54. Define CRCU_VAR_FAST Description

Name	CRCU_VAR_FAST
Initializer	

2.8.1.48 Define CRCU_VAR

CRCU memory and pointer classes.

Implements :

Table 55. Define CRCU_VAR Description

Name	CRCU_VAR
Initializer	

2.8.1.49 Define CANIF_CODE

CANIF memory and pointer classes.

Implements : DBASE04001

Table 56. Define CANIF_CODE Description

Name	CANIF_CODE
Initializer	

2.8.1.50 Define CANIF_CONST

CANIF memory and pointer classes.

Implements : DBASE04001

Table 57. Define CANIF_CONST Description

Name	CANIF_CONST
Initializer	

2.8.1.51 Define CANIF_APPL_DATA

CANIF memory and pointer classes.

Implements : DBASE04001

Table 58. Define CANIF_APPL_DATA Description

Name	CANIF_APPL_DATA
Initializer	

2.8.1.52 Define CANIF_APPL_CONST

CANIF memory and pointer classes.

Implements : DBASE04001

Table 59. Define CANIF_APPL_CONST Description

Name	CANIF_APPL_CONST
Initializer	

2.8.1.53 Define CANIF_APPL_CODE

CANIF memory and pointer classes.

Implements : DBASE04001

Table 60. Define CANIF_APPL_CODE Description

Name	CANIF_APPL_CODE
Initializer	

2.8.1.54 Define CANIF_CALLOUT_CODE

CANIF memory and pointer classes.

Implements : DBASE04001

Table 61. Define CANIF_CALLOUT_CODE Description

Name	CANIF_CALLOUT_CODE
Initializer	

2.8.1.55 Define CANIF_VAR_NOINIT

CANIF memory and pointer classes.

Implements : DBASE04001

Table 62. Define CANIF_VAR_NOINIT Description

Name	CANIF_VAR_NOINIT
Initializer	

2.8.1.56 Define CANIF_VAR_POWER_ON_INIT

CANIF memory and pointer classes.

Implements : DBASE04001

Table 63. Define CANIF_VAR_POWER_ON_INIT Description

Name	CANIF_VAR_POWER_ON_INIT
Initializer	

2.8.1.57 Define CANIF_VAR_FAST

CANIF memory and pointer classes.

Implements : DBASE04001

Table 64. Define CANIF_VAR_FAST Description

Name	CANIF_VAR_FAST

Table 64. Define CANIF_VAR_FAST Description*...continued*

Initializer	

2.8.1.58 Define CANIF_VAR

CANIF memory and pointer classes.

Implements : DBASE04001

Table 65. Define CANIF_VAR Description

Name	CANIF_VAR
Initializer	

2.8.1.59 Define DEM_CODE

DEM memory and pointer classes.

Implements : DBASE04001

Table 66. Define DEM_CODE Description

Name	DEM_CODE
Initializer	

2.8.1.60 Define DEM_CONST

DEM memory and pointer classes.

Implements : DBASE04001

Table 67. Define DEM_CONST Description

Name	DEM_CONST
Initializer	

2.8.1.61 Define DEM_APPL_DATA

DEM memory and pointer classes.

Implements : DBASE04001

Table 68. Define DEM_APPL_DATA Description

Name	DEM_APPL_DATA
Initializer	

2.8.1.62 Define DEM_APPL_CONST

DEM memory and pointer classes.

Implements : DBASE04001

Table 69. Define DEM_APPL_CONST Description

Name	DEM_APPL_CONST
Initializer	

2.8.1.63 Define DEM_APPL_CODE

DEM memory and pointer classes.

Implements : DBASE04001

Table 70. Define DEM_APPL_CODE Description

Name	DEM_APPL_CODE
Initializer	

2.8.1.64 Define DEM_CALLOUT_CODE

DEM memory and pointer classes.

Implements : DBASE04001

Table 71. Define DEM_CALLOUT_CODE Description

Name	DEM_CALLOUT_CODE
Initializer	

2.8.1.65 Define DEM_VAR_NOINIT

DEM memory and pointer classes.

Implements : DBASE04001

Table 72. Define DEM_VAR_NOINIT Description

Name	DEM_VAR_NOINIT
Initializer	

2.8.1.66 Define DEM_VAR_POWER_ON_INIT

DEM memory and pointer classes.

Implements : DBASE04001

Table 73. Define DEM_VAR_POWER_ON_INIT Description

Name	DEM_VAR_POWER_ON_INIT
Initializer	

2.8.1.67 Define DEM_VAR_FAST

DEM memory and pointer classes.

Implements : DBASE04001

Table 74. Define DEM_VAR_FAST Description

Name	DEM_VAR_FAST
Initializer	

2.8.1.68 Define DEM_VAR

DEM memory and pointer classes.

Implements : DBASE04001

Table 75. Define DEM_VAR Description

Name	DEM_VAR
Initializer	

2.8.1.69 Define DET_CODE

DET memory and pointer classes.

Implements : DBASE04001

Table 76. Define DET_CODE Description

Name	DET_CODE
Initializer	

2.8.1.70 Define DET_CONST

DET memory and pointer classes.

Implements : DBASE04001

Table 77. Define DET_CONST Description

Name	DET_CONST
Initializer	

2.8.1.71 Define DET_APPL_DATA

DET memory and pointer classes.

Implements : DBASE04001

Table 78. Define DET_APPL_DATA Description

Name	DET_APPL_DATA
Initializer	

2.8.1.72 Define DET_APPL_CONST

DET memory and pointer classes.

Implements : DBASE04001

Table 79. Define DET_APPL_CONST Description

Name	DET_APPL_CONST
Initializer	

2.8.1.73 Define DET_APPL_CODE

DET memory and pointer classes.

Implements : DBASE04001

Table 80. Define DET_APPL_CODE Description

Name	DET_APPL_CODE
Initializer	

2.8.1.74 Define DET_CALLOUT_CODE

DET memory and pointer classes.

Implements : DBASE04001

Table 81. Define DET_CALLOUT_CODE Description

Name	DET_CALLOUT_CODE
Initializer	

2.8.1.75 Define DET_VAR_NOINIT

DET memory and pointer classes.

Implements : DBASE04001

Table 82. Define DET_VAR_NOINIT Description

Name	DET_VAR_NOINIT
Initializer	

2.8.1.76 Define DET_VAR_POWER_ON_INIT

DET memory and pointer classes.

Implements : DBASE04001

Table 83. Define DET_VAR_POWER_ON_INIT Description

Name	DET_VAR_POWER_ON_INIT
Initializer	

2.8.1.77 Define DET_VAR_FAST

DET memory and pointer classes.

Implements : DBASE04001

Table 84. Define DET_VAR_FAST Description

Name	DET_VAR_FAST
Initializer	

2.8.1.78 Define DET_VAR

DET memory and pointer classes.

Implements : DBASE04001

Table 85. Define DET_VAR Description

Name	DET_VAR
Initializer	

2.8.1.79 Define DIO_CODE

DIO memory and pointer classes.

Implements : DBASE04001

Table 86. Define DIO_CODE Description

Name	DIO_CODE
Initializer	

2.8.1.80 Define DIO_CONST

DIO memory and pointer classes.

Implements : DBASE04001

Table 87. Define DIO_CONST Description

Name	DIO_CONST

Table 87. Define DIO_CONST Description*...continued*

Initializer	

2.8.1.81 Define DIO_APPL_DATA

DIO memory and pointer classes.

Implements : DBASE04001

Table 88. Define DIO_APPL_DATA Description

Name	DIO_APPL_DATA
Initializer	

2.8.1.82 Define DIO_APPL_CONST

DIO memory and pointer classes.

Implements : DBASE04001

Table 89. Define DIO_APPL_CONST Description

Name	DIO_APPL_CONST
Initializer	

2.8.1.83 Define DIO_APPL_CODE

DIO memory and pointer classes.

Implements : DBASE04001

Table 90. Define DIO_APPL_CODE Description

Name	DIO_APPL_CODE
Initializer	

2.8.1.84 Define DIO_CALLOUT_CODE

DIO memory and pointer classes.

Implements : DBASE04001

Table 91. Define DIO_CALLOUT_CODE Description

Name	DIO_CALLOUT_CODE
Initializer	

2.8.1.85 Define DIO_VAR_NOINIT

DIO memory and pointer classes.

Implements : DBASE04001

Table 92. Define DIO_VAR_NOINIT Description

Name	DIO_VAR_NOINIT
Initializer	

2.8.1.86 Define DIO_VAR_POWER_ON_INIT

DIO memory and pointer classes.

Implements : DBASE04001

Table 93. Define DIO_VAR_POWER_ON_INIT Description

Name	DIO_VAR_POWER_ON_INIT
Initializer	

2.8.1.87 Define DIO_VAR_FAST

DIO memory and pointer classes.

Implements : DBASE04001

Table 94. Define DIO_VAR_FAST Description

Name	DIO_VAR_FAST
Initializer	

2.8.1.88 Define DIO_VAR

DIO memory and pointer classes.

Implements : DBASE04001

Table 95. Define DIO_VAR Description

Name	DIO_VAR
Initializer	

2.8.1.89 Define ETH_CODE

ETH memory and pointer classes.

Implements : DBASE04001

Table 96. Define ETH_CODE Description

Name	ETH_CODE
Initializer	

2.8.1.90 Define ETH_CONST

ETH memory and pointer classes.

Implements : DBASE04001

Table 97. Define ETH_CONST Description

Name	ETH_CONST
Initializer	

2.8.1.91 Define ETH_APPL_DATA

ETH memory and pointer classes.

Implements : DBASE04001

Table 98. Define ETH_APPL_DATA Description

Name	ETH_APPL_DATA
Initializer	

2.8.1.92 Define ETH_APPL_CONST

ETH memory and pointer classes.

Implements : DBASE04001

Table 99. Define ETH_APPL_CONST Description

Name	ETH_APPL_CONST
Initializer	

2.8.1.93 Define ETH_APPL_CODE

ETH memory and pointer classes.

Implements : DBASE04001

Table 100. Define ETH_APPL_CODE Description

Name	ETH_APPL_CODE
Initializer	

2.8.1.94 Define ETH_CALLOUT_CODE

ETH memory and pointer classes.

Implements : DBASE04001

Table 101. Define ETH_CALLOUT_CODE Description

Name	ETH_CALLOUT_CODE
Initializer	

2.8.1.95 Define ETH_VAR_NOINIT

ETH memory and pointer classes.

Implements : DBASE04001

Table 102. Define ETH_VAR_NOINIT Description

Name	ETH_VAR_NOINIT
Initializer	

2.8.1.96 Define ETH_VAR_POWER_ON_INIT

ETH memory and pointer classes.

Implements : DBASE04001

Table 103. Define ETH_VAR_POWER_ON_INIT Description

Name	ETH_VAR_POWER_ON_INIT
Initializer	

2.8.1.97 Define ETH_VAR_FAST

ETH memory and pointer classes.

Implements : DBASE04001

Table 104. Define ETH_VAR_FAST Description

Name	ETH_VAR_FAST
Initializer	

2.8.1.98 Define ETH_VAR

ETH memory and pointer classes.

Implements : DBASE04001

Table 105. Define ETH_VAR Description

Name	ETH_VAR
Initializer	

2.8.1.99 Define ETH_AR_RELEASE_MAJOR_VERSION_ETHGENERALTYPES

Violates : MISRA rule 1.4

Table 106. Define ETH_AR_RELEASE_MAJOR_VERSION_ETHGENERALTYPES Description

Name	ETH_AR_RELEASE_MAJOR_VERSION_ETHGENERALTYPES
Initializer	4

2.8.1.100 Define ETH_AR_RELEASE_MINOR_VERSION_ETHGENERALTYPES

Violates : MISRA rule 1.4

Table 107. Define ETH_AR_RELEASE_MINOR_VERSION_ETHGENERALTYPES Description

Name	ETH_AR_RELEASE_MINOR_VERSION_ETHGENERALTYPES
Initializer	2

2.8.1.101 Define ETH_AR_RELEASE_REVISION_VERSION_ETHGENERALTYPES

Violates : MISRA rule 1.4

Table 108. Define ETH_AR_RELEASE_REVISION_VERSION_ETHGENERALTYPES Description

Name	ETH_AR_RELEASE_REVISION_VERSION_ETHGENERALTYPES
Initializer	2

2.8.1.102 Define ETH_MODULE_ID_ETHGENERALTYPES

Table 109. Define ETH_MODULE_ID_ETHGENERALTYPES Description

Name	ETH_MODULE_ID_ETHGENERALTYPES
Initializer	0

2.8.1.103 Define ETH_SW_MAJOR_VERSION_ETHGENERALTYPES

Violates : MISRA rule 1.4

Table 110. Define ETH_SW_MAJOR_VERSION_ETHGENERALTYPES Description

Name	ETH_SW_MAJOR_VERSION_ETHGENERALTYPES
Initializer	1

2.8.1.104 Define ETH_SW_MINOR_VERSION_ETHGENERALTYPES

Violates : MISRA rule 1.4

Table 111. Define ETH_SW_MINOR_VERSION_ETHGENERALTYPES Description

Name	ETH_SW_MINOR_VERSION_ETHGENERALTYPES
Initializer	0

2.8.1.105 Define ETH_SW_PATCH_VERSION_ETHGENERALTYPES

Violates : MISRA rule 1.4

Table 112. Define ETH_SW_PATCH_VERSION_ETHGENERALTYPES Description

Name	ETH_SW_PATCH_VERSION_ETHGENERALTYPES
Initializer	5

2.8.1.106 Define ETH_VENDOR_ID_ETHGENERALTYPES

Table 113. Define ETH_VENDOR_ID_ETHGENERALTYPES Description

Name	ETH_VENDOR_ID_ETHGENERALTYPES
Initializer	43

2.8.1.107 Define FEE_CODE

FEE memory and pointer classes.

Implements : DBASE04001

Table 114. Define FEE_CODE Description

Name	FEE_CODE
Initializer	

2.8.1.108 Define FEE_CONST

FEE memory and pointer classes.

Implements : DBASE04001

Table 115. Define FEE_CONST Description

Name	FEE_CONST
Initializer	

2.8.1.109 Define FEE_APPL_DATA

FEE memory and pointer classes.

Implements : DBASE04001

Table 116. Define FEE_APPL_DATA Description

Name	FEE_APPL_DATA
Initializer	

2.8.1.110 Define FEE_APPL_CONST

FEE memory and pointer classes.

Implements : DBASE04001

Table 117. Define FEE_APPL_CONST Description

Name	FEE_APPL_CONST
Initializer	

2.8.1.111 Define FEE_APPL_CODE

FEE memory and pointer classes.

Implements : DBASE04001

Table 118. Define FEE_APPL_CODE Description

Name	FEE_APPL_CODE
Initializer	

2.8.1.112 Define FEE_CALLOUT_CODE

FEE memory and pointer classes.

Implements : DBASE04001

Table 119. Define FEE_CALLOUT_CODE Description

Name	FEE_CALLOUT_CODE
Initializer	

2.8.1.113 Define FEE_VAR_NOINIT

FEE memory and pointer classes.

Implements : DBASE04001

Table 120. Define FEE_VAR_NOINIT Description

Name	FEE_VAR_NOINIT
Initializer	

2.8.1.114 Define FEE_VAR_POWER_ON_INIT

FEE memory and pointer classes.

Implements : DBASE04001

Table 121. Define FEE_VAR_POWER_ON_INIT Description

Name	FEE_VAR_POWER_ON_INIT
Initializer	

2.8.1.115 Define FEE_VAR_FAST

FEE memory and pointer classes.

Implements : DBASE04001

Table 122. Define FEE_VAR_FAST Description

Name	FEE_VAR_FAST
Initializer	

2.8.1.116 Define FEE_VAR

FEE memory and pointer classes.

Implements : DBASE04001

Table 123. Define FEE_VAR Description

Name	FEE_VAR

Table 123. Define FEE_VAR Description*...continued*

Initializer	

2.8.1.117 Define FLS_CODE

FLS memory and pointer classes.

Implements : DBASE04001

Table 124. Define FLS_CODE Description

Name	FLS_CODE
Initializer	

2.8.1.118 Define FLS_CONST

FLS memory and pointer classes.

Implements : DBASE04001

Table 125. Define FLS_CONST Description

Name	FLS_CONST
Initializer	

2.8.1.119 Define FLS_APPL_DATA

FLS memory and pointer classes.

Implements : DBASE04001

Table 126. Define FLS_APPL_DATA Description

Name	FLS_APPL_DATA
Initializer	

2.8.1.120 Define FLS_APPL_CONST

FLS memory and pointer classes.

Implements : DBASE04001

Table 127. Define FLS_APPL_CONST Description

Name	FLS_APPL_CONST
Initializer	

2.8.1.121 Define FLS_APPL_CODE

FLS memory and pointer classes.

Implements : DBASE04001

Table 128. Define FLS_APPL_CODE Description

Name	FLS_APPL_CODE
Initializer	

2.8.1.122 Define FLS_CALLOUT_CODE

FLS memory and pointer classes.

Implements : DBASE04001

Table 129. Define FLS_CALLOUT_CODE Description

Name	FLS_CALLOUT_CODE
Initializer	

2.8.1.123 Define FLS_VAR_NOINIT

FLS memory and pointer classes.

Implements : DBASE04001

Table 130. Define FLS_VAR_NOINIT Description

Name	FLS_VAR_NOINIT
Initializer	

2.8.1.124 Define FLS_VAR_POWER_ON_INIT

FLS memory and pointer classes.

Implements : DBASE04001

Table 131. Define FLS_VAR_POWER_ON_INIT Description

Name	FLS_VAR_POWER_ON_INIT
Initializer	

2.8.1.125 Define FLS_VAR_FAST

FLS memory and pointer classes.

Implements : DBASE04001

Table 132. Define FLS_VAR_FAST Description

Name	FLS_VAR_FAST
Initializer	

2.8.1.126 Define FLS_VAR

FLS memory and pointer classes.

Implements : DBASE04001

Table 133. Define FLS_VAR Description

Name	FLS_VAR
Initializer	

2.8.1.127 Define FR_APPL_CODE

FlexRay memory and pointer classes.

Implements : DBASE04001

Table 134. Define FR_APPL_CODE Description

Name	FR_APPL_CODE
Initializer	

2.8.1.128 Define FR_APPL_CONST

FlexRay memory and pointer classes.

Implements : DBASE04001

Table 135. Define FR_APPL_CONST Description

Name	FR_APPL_CONST
Initializer	

2.8.1.129 Define FR_APPL_DATA

FlexRay memory and pointer classes.

Implements : DBASE04001

Table 136. Define FR_APPL_DATA Description

Name	FR_APPL_DATA
Initializer	

2.8.1.130 Define FR_CALLOUT_CODE

FlexRay memory and pointer classes.

Implements : DBASE04001

Table 137. Define FR_CALLOUT_CODE Description

Name	FR_CALLOUT_CODE
Initializer	

2.8.1.131 Define FR_CIDX_GCOLDSTARTATTEMPTS

Table 138. Define FR_CIDX_GCOLDSTARTATTEMPTS Description

Name	FR_CIDX_GCOLDSTARTATTEMPTS
Initializer	17U

2.8.1.132 Define FR_CIDX_GCYCLECOUNTMAX

Table 139. Define FR_CIDX_GCYCLECOUNTMAX Description

Name	FR_CIDX_GCYCLECOUNTMAX
Initializer	18U

2.8.1.133 Define FR_CIDX_GDACTIONPOINTOFFSET

Table 140. Define FR_CIDX_GDACTIONPOINTOFFSET Description

Name	FR_CIDX_GDACTIONPOINTOFFSET
Initializer	25U

2.8.1.134 Define FR_CIDX_GDBIT

Table 141. Define FR_CIDX_GDBIT Description

Name	FR_CIDX_GDBIT
Initializer	26U

2.8.1.135 Define FR_CIDX_GDCASRXLOWMAX

Table 142. Define FR_CIDX_GDCASRXLOWMAX Description

Name	FR_CIDX_GDCASRXLOWMAX
Initializer	27U

2.8.1.136 Define FR_CIDX_GDCYCLE

Macros which can be passed into Fr_ReadCCConfig as parameter Fr_ConfigParamIdx.

Details :

Each macro (index) uniquely identifies a configuration parameter which value can be read out of the controllers configuration using Fr_ReadCCConfig.

Covers FR657

Implements : DFR32010

Table 143. Define FR_CIDX_GDCYCLE Description

Name	FR_CIDX_GDCYCLE
Initializer	0U

2.8.1.137 Define FR_CIDX_GDDYNAMICSLOTIDLEPHASE

Table 144. Define FR_CIDX_GDDYNAMICSLOTIDLEPHASE Description

Name	FR_CIDX_GDDYNAMICSLOTIDLEPHASE
Initializer	28U

2.8.1.138 Define FR_CIDX_GDIGNOREAFTERTX

Table 145. Define FR_CIDX_GDIGNOREAFTERTX Description

Name	FR_CIDX_GDIGNOREAFTERTX
Initializer	54U

2.8.1.139 Define FR_CIDX_GDMACROTICK

Table 146. Define FR_CIDX_GDMACROTICK Description

Name	FR_CIDX_GDMACROTICK
Initializer	4U

2.8.1.140 Define FR_CIDX_GDMINISLOT

Table 147. Define FR_CIDX_GDMINISLOT Description

Name	FR_CIDX_GDMINISLOT
Initializer	30U

2.8.1.141 Define FR_CIDX_GDMINISLOTACTIONPOINTOFFSET

Table 148. Define FR_CIDX_GDMINISLOTACTIONPOINTOFFSET Description

Name	FR_CIDX_GDMINISLOTACTIONPOINTOFFSET
Initializer	29U

2.8.1.142 Define FR_CIDX_GDNIT

Table 149. Define FR_CIDX_GDNIT Description

Name	FR_CIDX_GDNIT
Initializer	7U

2.8.1.143 Define FR_CIDX_GDSAMPLECLOCKPERIOD

Table 150. Define FR_CIDX_GDSAMPLECLOCKPERIOD Description

Name	FR_CIDX_GDSAMPLECLOCKPERIOD
Initializer	31U

2.8.1.144 Define FR_CIDX_GDSTATICSLOT

Table 151. Define FR_CIDX_GDSTATICSLOT Description

Name	FR_CIDX_GDSTATICSLOT
Initializer	8U

2.8.1.145 Define FR_CIDX_GDSYMBOLWINDOW

Table 152. Define FR_CIDX_GDSYMBOLWINDOW Description

Name	FR_CIDX_GDSYMBOLWINDOW
Initializer	32U

2.8.1.146 Define FR_CIDX_GDSYMBOLWINDOWACTIONPOINTOFFSET

Table 153. Define FR_CIDX_GDSYMBOLWINDOWACTIONPOINTOFFSET Description

Name	FR_CIDX_GDSYMBOLWINDOWACTIONPOINTOFFSET
Initializer	33U

2.8.1.147 Define FR_CIDX_GDTSSTRANSMITTER

Table 154. Define FR_CIDX_GDTSSTRANSMITTER Description

Name	FR_CIDX_GDTSSTRANSMITTER
Initializer	34U

2.8.1.148 Define FR_CIDX_GDWAKEUPRXIDLE

Table 155. Define FR_CIDX_GDWAKEUPRXIDLE Description

Name	FR_CIDX_GDWAKEUPRXIDLE
Initializer	35U

2.8.1.149 Define FR_CIDX_GDWAKEUPRXLOW

Table 156. Define FR_CIDX_GDWAKEUPRXLOW Description

Name	FR_CIDX_GDWAKEUPRXLOW
Initializer	36U

2.8.1.150 Define FR_CIDX_GDWAKEUPRXWINDOW

Table 157. Define FR_CIDX_GDWAKEUPRXWINDOW Description

Name	FR_CIDX_GDWAKEUPRXWINDOW
Initializer	9U

2.8.1.151 Define FR_CIDX_GDWAKEUPTXACTIVE

Table 158. Define FR_CIDX_GDWAKEUPTXACTIVE Description

Name	FR_CIDX_GDWAKEUPTXACTIVE
Initializer	37U

2.8.1.152 Define FR_CIDX_GDWAKEUPTXIDLE

Table 159. Define FR_CIDX_GDWAKEUPTXIDLE Description

Name	FR_CIDX_GDWAKEUPTXIDLE
Initializer	38U

2.8.1.153 Define FR_CIDX_GLISTENNOISE

Table 160. Define FR_CIDX_GLISTENNOISE Description

Name	FR_CIDX_GLISTENNOISE
Initializer	19U

2.8.1.154 Define FR_CIDX_GMACROPERCYCLE

Table 161. Define FR_CIDX_GMACROPERCYCLE Description

Name	FR_CIDX_GMACROPERCYCLE
Initializer	3U

2.8.1.155 Define FR_CIDX_GMAXWITHOUTCLOCKCORRECTFATAL

Table 162. Define FR_CIDX_GMAXWITHOUTCLOCKCORRECTFATAL Description

Name	FR_CIDX_GMAXWITHOUTCLOCKCORRECTFATAL
Initializer	20U

2.8.1.156 Define FR_CIDX_GMAXWITHOUTCLOCKCORRECTPASSIVE

Table 163. Define FR_CIDX_GMAXWITHOUTCLOCKCORRECTPASSIVE Description

Name	FR_CIDX_GMAXWITHOUTCLOCKCORRECTPASSIVE
Initializer	21U

2.8.1.157 Define FR_CIDX_GNETWORKMANAGEMENTVECTORLENGTH

Table 164. Define FR_CIDX_GNETWORKMANAGEMENTVECTORLENGTH Description

Name	FR_CIDX_GNETWORKMANAGEMENTVECTORLENGTH
Initializer	22U

2.8.1.158 Define FR_CIDX_GNUMBEROFMINISLOTS

Table 165. Define FR_CIDX_GNUMBEROFMINISLOTS Description

Name	FR_CIDX_GNUMBEROFMINISLOTS
Initializer	5U

2.8.1.159 Define FR_CIDX_GNUMBEROFSTATICSLOTS

Table 166. Define FR_CIDX_GNUMBEROFSTATICSLOTS Description

Name	FR_CIDX_GNUMBEROFSTATICSLOTS
Initializer	6U

2.8.1.160 Define FR_CIDX_GPAYLOADLENGTHSTATIC

Table 167. Define FR_CIDX_GPAYLOADLENGTHSTATIC Description

Name	FR_CIDX_GPAYLOADLENGTHSTATIC
Initializer	23U

2.8.1.161 Define FR_CIDX_GSYNCFRAMEIDCOUNTMAX

Table 168. Define FR_CIDX_GSYNCFRAMEIDCOUNTMAX Description

Name	FR_CIDX_GSYNCFRAMEIDCOUNTMAX
Initializer	24U

2.8.1.162 Define FR_CIDX_PALLOWHALTDUETOCLOCK

Table 169. Define FR_CIDX_PALLOWHALTDUETOCLOCK Description

Name	FR_CIDX_PALLOWHALTDUETOCLOCK
Initializer	55U

2.8.1.163 Define FR_CIDX_PALLOWPASSIVETOACTIVE

Table 170. Define FR_CIDX_PALLOWPASSIVETOACTIVE Description

Name	FR_CIDX_PALLOWPASSIVETOACTIVE
Initializer	39U

2.8.1.164 Define FR_CIDX_PCHANNELS

Table 171. Define FR_CIDX_PCHANNELS Description

Name	FR_CIDX_PCHANNELS
Initializer	40U

2.8.1.165 Define FR_CIDX_PCLUSTERDRIFTDAMPING

Table 172. Define FR_CIDX_PCLUSTERDRIFTDAMPING Description

Name	FR_CIDX_PCLUSTERDRIFTDAMPING
Initializer	41U

2.8.1.166 Define FR_CIDX_PDACCEPTEDSTARTUPRANGE

Table 173. Define FR_CIDX_PDACCEPTEDSTARTUPRANGE Description

Name	FR_CIDX_PDACCEPTEDSTARTUPRANGE
Initializer	16U

2.8.1.167 Define FR_CIDX_PDECODINGCORRECTION

Table 174. Define FR_CIDX_PDECODINGCORRECTION Description

Name	FR_CIDX_PDECODINGCORRECTION
Initializer	42U

2.8.1.168 Define FR_CIDX_PDELAYCOMPENSATIONA

Table 175. Define FR_CIDX_PDELAYCOMPENSATIONA Description

Name	FR_CIDX_PDELAYCOMPENSATIONA
Initializer	43U

2.8.1.169 Define FR_CIDX_PDELAYCOMPENSATIONB

Table 176. Define FR_CIDX_PDELAYCOMPENSATIONB Description

Name	FR_CIDX_PDELAYCOMPENSATIONB
Initializer	44U

2.8.1.170 Define FR_CIDX_PDLISTENTIMEOUT

Table 177. Define FR_CIDX_PDLISTENTIMEOUT Description

Name	FR_CIDX_PDLISTENTIMEOUT
Initializer	2U

2.8.1.171 Define FR_CIDX_PDMICROTICK

Table 178. Define FR_CIDX_PDMICROTICK Description

Name	FR_CIDX_PDMICROTICK
Initializer	53U

2.8.1.172 Define FR_CIDX_PEXTERNALSYNC

Table 179. Define FR_CIDX_PEXTERNALSYNC Description

Name	FR_CIDX_PEXTERNALSYNC
Initializer	56U

2.8.1.173 Define FR_CIDX_PFALLBACKINTERNAL

Table 180. Define FR_CIDX_PFALLBACKINTERNAL Description

Name	FR_CIDX_PFALLBACKINTERNAL
Initializer	57U

2.8.1.174 Define FR_CIDX_PKEYSLOTID

Table 181. Define FR_CIDX_PKEYSLOTID Description

Name	FR_CIDX_PKEYSLOTID
Initializer	10U

2.8.1.175 Define FR_CIDX_PKEYSLOTONLYENABLED

Table 182. Define FR_CIDX_PKEYSLOTONLYENABLED Description

Name	FR_CIDX_PKEYSLOTONLYENABLED
Initializer	58U

2.8.1.176 Define FR_CIDX_PKEYSLOTUSEDFORSTARTUP

Table 183. Define FR_CIDX_PKEYSLOTUSEDFORSTARTUP Description

Name	FR_CIDX_PKEYSLOTUSEDFORSTARTUP
Initializer	59U

2.8.1.177 Define FR_CIDX_PKEYSLOTUSEDFORSYNC

Table 184. Define FR_CIDX_PKEYSLOTUSEDFORSYNC Description

Name	FR_CIDX_PKEYSLOTUSEDFORSYNC
Initializer	60U

2.8.1.178 Define FR_CIDX_PLATESTTX

Table 185. Define FR_CIDX_PLATESTTX Description

Name	FR_CIDX_PLATESTTX
Initializer	11U

2.8.1.179 Define FR_CIDX_PMACROINITIALOFFSETA

Table 186. Define FR_CIDX_PMACROINITIALOFFSETA Description

Name	FR_CIDX_PMACROINITIALOFFSETA
Initializer	45U

2.8.1.180 Define FR_CIDX_PMACROINITIALOFFSETB

Table 187. Define FR_CIDX_PMACROINITIALOFFSETB Description

Name	FR_CIDX_PMACROINITIALOFFSETB
Initializer	46U

2.8.1.181 Define FR_CIDX_PMICROINITIALOFFSETA

Table 188. Define FR_CIDX_PMICROINITIALOFFSETA Description

Name	FR_CIDX_PMICROINITIALOFFSETA
Initializer	47U

2.8.1.182 Define FR_CIDX_PMICROINITIALOFFSETB

Table 189. Define FR_CIDX_PMICROINITIALOFFSETB Description

Name	FR_CIDX_PMICROINITIALOFFSETB
Initializer	48U

2.8.1.183 Define FR_CIDX_PMICROPERCYCLE

Table 190. Define FR_CIDX_PMICROPERCYCLE Description

Name	FR_CIDX_PMICROPERCYCLE
Initializer	1U

2.8.1.184 Define FR_CIDX_PNMVECTOREARLYUPDATE

Table 191. Define FR_CIDX_PNMVECTOREARLYUPDATE Description

Name	FR_CIDX_PNMVECTOREARLYUPDATE
Initializer	61U

2.8.1.185 Define FR_CIDX_POFFSETCORRECTIONOUT

Table 192. Define FR_CIDX_POFFSETCORRECTIONOUT Description

Name	FR_CIDX_POFFSETCORRECTIONOUT
Initializer	12U

2.8.1.186 Define FR_CIDX_POFFSETCORRECTIONSTART

Table 193. Define FR_CIDX_POFFSETCORRECTIONSTART Description

Name	FR_CIDX_POFFSETCORRECTIONSTART
Initializer	13U

2.8.1.187 Define FR_CIDX_PPAYLOADLENGTHDYNMAX

Table 194. Define FR_CIDX_PPAYLOADLENGTHDYNMAX Description

Name	FR_CIDX_PPAYLOADLENGTHDYNMAX
Initializer	49U

2.8.1.188 Define FR_CIDX_PRATECORRECTIONOUT

Table 195. Define FR_CIDX_PRATECORRECTIONOUT Description

Name	FR_CIDX_PRATECORRECTIONOUT
Initializer	14U

2.8.1.189 Define FR_CIDX_PSAMPLESPERMICROTICK

Table 196. Define FR_CIDX_PSAMPLESPERMICROTICK Description

Name	FR_CIDX_PSAMPLESPERMICROTICK
Initializer	50U

2.8.1.190 Define FR_CIDX_PSECONDKEYSLOTID

Table 197. Define FR_CIDX_PSECONDKEYSLOTID Description

Name	FR_CIDX_PSECONDKEYSLOTID
Initializer	15U

2.8.1.191 Define FR_CIDX_PTWOKEYSLOTMODE

Table 198. Define FR_CIDX_PTWOKEYSLOTMODE Description

Name	FR_CIDX_PTWOKEYSLOTMODE
Initializer	62U

2.8.1.192 Define FR_CIDX_PWAKEUPCHANNEL

Table 199. Define FR_CIDX_PWAKEUPCHANNEL Description

Name	FR_CIDX_PWAKEUPCHANNEL
Initializer	51U

2.8.1.193 Define FR_CIDX_PWAKEUPPATTERN

Table 200. Define FR_CIDX_PWAKEUPPATTERN Description

Name	FR_CIDX_PWAKEUPPATTERN
Initializer	52U

2.8.1.194 Define FR_CODE

FlexRay memory and pointer classes.

Implements : DBASE04001

Table 201. Define FR_CODE Description

Name	FR_CODE
Initializer	

2.8.1.195 Define FR_CONST

FlexRay memory and pointer classes.

Implements : DBASE04001

Table 202. Define FR_CONST Description

Name	FR_CONST
Initializer	

2.8.1.196 Define FR_SLOTMODE_SINGLE

This macro is used for backward compatibility with Autosar 3.0 definition of Fr_SlotModeType Covers FR599.

Implements : DFR32011

Table 203. Define FR_SLOTMODE_SINGLE Description

Name	FR_SLOTMODE_SINGLE
Initializer	FR_SLOTMODE_KEYSLOT

2.8.1.197 Define FR_VAR

FlexRay memory and pointer classes.

Implements : DBASE04001

Table 204. Define FR_VAR Description

Name	FR_VAR
Initializer	

2.8.1.198 Define FR_VAR_FAST

FlexRay memory and pointer classes.

Implements : DBASE04001

Table 205. Define FR_VAR_FAST Description

Name	FR_VAR_FAST
Initializer	

2.8.1.199 Define FR_VAR_NOINIT

FlexRay memory and pointer classes.

Implements : DBASE04001

Table 206. Define FR_VAR_NOINIT Description

Name	FR_VAR_NOINIT
Initializer	

2.8.1.200 Define FR_VAR_POWER_ON_INIT

FlexRay memory and pointer classes.

Implements : DBASE04001

Table 207. Define FR_VAR_POWER_ON_INIT Description

Name	FR_VAR_POWER_ON_INIT
Initializer	

2.8.1.201 Define GPT_CODE

GPT memory and pointer classes.

Implements : DBASE04001

Table 208. Define GPT_CODE Description

Name	GPT_CODE

Table 208. Define GPT_CODE Description*...continued*

Initializer	

2.8.1.202 Define GPT_CONST

GPT memory and pointer classes.

Implements : DBASE04001

Table 209. Define GPT_CONST Description

Name	GPT_CONST
Initializer	

2.8.1.203 Define GPT_APPL_DATA

GPT memory and pointer classes.

Implements : DBASE04001

Table 210. Define GPT_APPL_DATA Description

Name	GPT_APPL_DATA
Initializer	

2.8.1.204 Define GPT_APPL_CONST

GPT memory and pointer classes.

Implements : DBASE04001

Table 211. Define GPT_APPL_CONST Description

Name	GPT_APPL_CONST
Initializer	

2.8.1.205 Define GPT_APPL_CODE

GPT memory and pointer classes.

Implements : DBASE04001

Table 212. Define GPT_APPL_CODE Description

Name	GPT_APPL_CODE
Initializer	

2.8.1.206 Define GPT_CALLOUT_CODE

GPT memory and pointer classes.

Implements : DBASE04001

Table 213. Define GPT_CALLOUT_CODE Description

Name	GPT_CALLOUT_CODE
Initializer	

2.8.1.207 Define GPT_VAR_NOINIT

GPT memory and pointer classes.

Implements : DBASE04001

Table 214. Define GPT_VAR_NOINIT Description

Name	GPT_VAR_NOINIT
Initializer	

2.8.1.208 Define GPT_VAR_POWER_ON_INIT

GPT memory and pointer classes.

Implements : DBASE04001

Table 215. Define GPT_VAR_POWER_ON_INIT Description

Name	GPT_VAR_POWER_ON_INIT
Initializer	

2.8.1.209 Define GPT_VAR_FAST

GPT memory and pointer classes.

Implements : DBASE04001

Table 216. Define GPT_VAR_FAST Description

Name	GPT_VAR_FAST
Initializer	

2.8.1.210 Define GPT_VAR

GPT memory and pointer classes.

Implements : DBASE04001

Table 217. Define GPT_VAR Description

Name	GPT_VAR
Initializer	

2.8.1.211 Define ICU_CODE

ICU memory and pointer classes.

Implements : DBASE04001

Table 218. Define ICU_CODE Description

Name	ICU_CODE
Initializer	

2.8.1.212 Define ICU_CONST

ICU memory and pointer classes.

Implements : DBASE04001

Table 219. Define ICU_CONST Description

Name	ICU_CONST
Initializer	

2.8.1.213 Define ICU_APPL_DATA

ICU memory and pointer classes.

Implements : DBASE04001

Table 220. Define ICU_APPL_DATA Description

Name	ICU_APPL_DATA
Initializer	

2.8.1.214 Define ICU_APPL_CONST

ICU memory and pointer classes.

Implements : DBASE04001

Table 221. Define ICU_APPL_CONST Description

Name	ICU_APPL_CONST
Initializer	

2.8.1.215 Define ICU_APPL_CODE

ICU memory and pointer classes.

Implements : DBASE04001

Table 222. Define ICU_APPL_CODE Description

Name	ICU_APPL_CODE
Initializer	

2.8.1.216 Define ICU_CALLOUT_CODE

ICU memory and pointer classes.

Implements : DBASE04001

Table 223. Define ICU_CALLOUT_CODE Description

Name	ICU_CALLOUT_CODE
Initializer	

2.8.1.217 Define ICU_VAR_NOINIT

ICU memory and pointer classes.

Implements : DBASE04001

Table 224. Define ICU_VAR_NOINIT Description

Name	ICU_VAR_NOINIT
Initializer	

2.8.1.218 Define ICU_VAR_POWER_ON_INIT

ICU memory and pointer classes.

Implements : DBASE04001

Table 225. Define ICU_VAR_POWER_ON_INIT Description

Name	ICU_VAR_POWER_ON_INIT
Initializer	

2.8.1.219 Define ICU_VAR_FAST

ICU memory and pointer classes.

Implements : DBASE04001

Table 226. Define ICU_VAR_FAST Description

Name	ICU_VAR_FAST
Initializer	

2.8.1.220 Define ICU_VAR

ICU memory and pointer classes.

Implements : DBASE04001

Table 227. Define ICU_VAR Description

Name	ICU_VAR
Initializer	

2.8.1.221 Define LIN_CODE

LIN memory and pointer classes.

Implements : DBASE04001

Table 228. Define LIN_CODE Description

Name	LIN_CODE
Initializer	

2.8.1.222 Define LIN_CONST

LIN memory and pointer classes.

Implements : DBASE04001

Table 229. Define LIN_CONST Description

Name	LIN_CONST
Initializer	

2.8.1.223 Define LIN_APPL_DATA

LIN memory and pointer classes.

Implements : DBASE04001

Table 230. Define LIN_APPL_DATA Description

Name	LIN_APPL_DATA
Initializer	

2.8.1.224 Define LIN_APPL_CONST

LIN memory and pointer classes.

Implements : DBASE04001

Table 231. Define LIN_APPL_CONST Description

Name	LIN_APPL_CONST

Table 231. Define LIN_APPL_CONST Description*...continued*

Initializer	

2.8.1.225 Define LIN_APPL_CODE

LIN memory and pointer classes.

Implements : DBASE04001

Table 232. Define LIN_APPL_CODE Description

Name	LIN_APPL_CODE
Initializer	

2.8.1.226 Define LIN_CALLOUT_CODE

LIN memory and pointer classes.

Implements : DBASE04001

Table 233. Define LIN_CALLOUT_CODE Description

Name	LIN_CALLOUT_CODE
Initializer	

2.8.1.227 Define LIN_VAR_NOINIT

LIN memory and pointer classes.

Implements : DBASE04001

Table 234. Define LIN_VAR_NOINIT Description

Name	LIN_VAR_NOINIT
Initializer	

2.8.1.228 Define LIN_VAR_POWER_ON_INIT

LIN memory and pointer classes.

Implements : DBASE04001

Table 235. Define LIN_VAR_POWER_ON_INIT Description

Name	LIN_VAR_POWER_ON_INIT
Initializer	

2.8.1.229 Define LIN_VAR_FAST

LIN memory and pointer classes.

Implements : DBASE04001

Table 236. Define LIN_VAR_FAST Description

Name	LIN_VAR_FAST
Initializer	

2.8.1.230 Define LIN_VAR

LIN memory and pointer classes.

Implements : DBASE04001

Table 237. Define LIN_VAR Description

Name	LIN_VAR
Initializer	

2.8.1.231 Define MCEM_CODE

MCEM memory and pointer classes.

Implements :

Table 238. Define MCEM_CODE Description

Name	MCEM_CODE
Initializer	

2.8.1.232 Define MCEM_CONST

MCEM memory and pointer classes.

Implements :

Table 239. Define MCEM_CONST Description

Name	MCEM_CONST
Initializer	

2.8.1.233 Define MCEM_APPL_DATA

MCEM memory and pointer classes.

Implements :

Table 240. Define MCEM_APPL_DATA Description

Name	MCEM_APPL_DATA
Initializer	

2.8.1.234 Define MCEM_APPL_CONST

MCEM memory and pointer classes.

Implements :

Table 241. Define MCEM_APPL_CONST Description

Name	MCEM_APPL_CONST
Initializer	

2.8.1.235 Define MCEM_APPL_CODE

MCEM memory and pointer classes.

Implements :

Table 242. Define MCEM_APPL_CODE Description

Name	MCEM_APPL_CODE
Initializer	

2.8.1.236 Define MCEM_CALLOUT_CODE

MCEM memory and pointer classes.

Implements :

Table 243. Define MCEM_CALLOUT_CODE Description

Name	MCEM_CALLOUT_CODE
Initializer	

2.8.1.237 Define MCEM_VAR_NOINIT

MCEM memory and pointer classes.

Implements :

Table 244. Define MCEM_VAR_NOINIT Description

Name	MCEM_VAR_NOINIT
Initializer	

2.8.1.238 Define MCEM_VAR_POWER_ON_INIT

MCEM memory and pointer classes.

Implements :

Table 245. Define MCEM_VAR_POWER_ON_INIT Description

Name	MCEM_VAR_POWER_ON_INIT
Initializer	

2.8.1.239 Define MCEM_VAR_FAST

MCEM memory and pointer classes.

Implements :

Table 246. Define MCEM_VAR_FAST Description

Name	MCEM_VAR_FAST
Initializer	

2.8.1.240 Define MCEM_VAR

MCEM memory and pointer classes.

Implements :

Table 247. Define MCEM_VAR Description

Name	MCEM_VAR
Initializer	

2.8.1.241 Define MCL_CODE

MCL memory and pointer classes.

Implements :

Table 248. Define MCL_CODE Description

Name	MCL_CODE
Initializer	

2.8.1.242 Define MCL_CONST

MCL memory and pointer classes.

Implements :

Table 249. Define MCL_CONST Description

Name	MCL_CONST
Initializer	

2.8.1.243 Define MCL_APPL_DATA

MCL memory and pointer classes.

Implements :

Table 250. Define MCL_APPL_DATA Description

Name	MCL_APPL_DATA
Initializer	

2.8.1.244 Define MCL_APPL_CONST

MCL memory and pointer classes.

Implements :

Table 251. Define MCL_APPL_CONST Description

Name	MCL_APPL_CONST
Initializer	

2.8.1.245 Define MCL_APPL_CODE

MCL memory and pointer classes.

Implements :

Table 252. Define MCL_APPL_CODE Description

Name	MCL_APPL_CODE
Initializer	

2.8.1.246 Define MCL_CALLOUT_CODE

MCL memory and pointer classes.

Implements :

Table 253. Define MCL_CALLOUT_CODE Description

Name	MCL_CALLOUT_CODE
Initializer	

2.8.1.247 Define MCL_VAR_NOINIT

MCL memory and pointer classes.

Implements :

Table 254. Define MCL_VAR_NOINIT Description

Name	MCL_VAR_NOINIT

Table 254. Define MCL_VAR_NOINIT Description*...continued*

Initializer	

2.8.1.248 Define MCL_VAR_POWER_ON_INIT

MCL memory and pointer classes.

Implements :

Table 255. Define MCL_VAR_POWER_ON_INIT Description

Name	MCL_VAR_POWER_ON_INIT
Initializer	

2.8.1.249 Define MCL_VAR_FAST

MCL memory and pointer classes.

Implements :

Table 256. Define MCL_VAR_FAST Description

Name	MCL_VAR_FAST
Initializer	

2.8.1.250 Define MCL_VAR

MCL memory and pointer classes.

Implements :

Table 257. Define MCL_VAR Description

Name	MCL_VAR
Initializer	

2.8.1.251 Define MCU_CODE

MCU memory and pointer classes.

Implements : DBASE04001

Table 258. Define MCU_CODE Description

Name	MCU_CODE
Initializer	

2.8.1.252 Define MCU_CONST

MCU memory and pointer classes.

Implements : DBASE04001

Table 259. Define MCU_CONST Description

Name	MCU_CONST
Initializer	

2.8.1.253 Define MCU_APPL_DATA

MCU memory and pointer classes.

Implements : DBASE04001

Table 260. Define MCU_APPL_DATA Description

Name	MCU_APPL_DATA
Initializer	

2.8.1.254 Define MCU_APPL_CONST

MCU memory and pointer classes.

Implements : DBASE04001

Table 261. Define MCU_APPL_CONST Description

Name	MCU_APPL_CONST
Initializer	

2.8.1.255 Define MCU_APPL_CODE

MCU memory and pointer classes.

Implements : DBASE04001

Table 262. Define MCU_APPL_CODE Description

Name	MCU_APPL_CODE
Initializer	

2.8.1.256 Define MCU_CALLOUT_CODE

MCU memory and pointer classes.

Implements : DBASE04001

Table 263. Define MCU_CALLOUT_CODE Description

Name	MCU_CALLOUT_CODE
Initializer	

2.8.1.257 Define MCU_VAR_NOINIT

MCU memory and pointer classes.

Implements : DBASE04001

Table 264. Define MCU_VAR_NOINIT Description

Name	MCU_VAR_NOINIT
Initializer	

2.8.1.258 Define MCU_VAR_POWER_ON_INIT

MCU memory and pointer classes.

Implements : DBASE04001

Table 265. Define MCU_VAR_POWER_ON_INIT Description

Name	MCU_VAR_POWER_ON_INIT
Initializer	

2.8.1.259 Define MCU_VAR_FAST

MCU memory and pointer classes.

Implements : DBASE04001

Table 266. Define MCU_VAR_FAST Description

Name	MCU_VAR_FAST
Initializer	

2.8.1.260 Define MCU_VAR

MCU memory and pointer classes.

Implements : DBASE04001

Table 267. Define MCU_VAR Description

Name	MCU_VAR
Initializer	

2.8.1.261 Define PORT_CODE

PORT memory and pointer classes.

Implements : DBASE04001

Table 268. Define PORT_CODE Description

Name	PORT_CODE
Initializer	

2.8.1.262 Define PORT_CONST

PORT memory and pointer classes.

Implements : DBASE04001

Table 269. Define PORT_CONST Description

Name	PORT_CONST
Initializer	

2.8.1.263 Define PORT_APPL_DATA

PORT memory and pointer classes.

Implements : DBASE04001

Table 270. Define PORT_APPL_DATA Description

Name	PORT_APPL_DATA
Initializer	

2.8.1.264 Define PORT_APPL_CONST

PORT memory and pointer classes.

Implements : DBASE04001

Table 271. Define PORT_APPL_CONST Description

Name	PORT_APPL_CONST
Initializer	

2.8.1.265 Define PORT_APPL_CODE

PORT memory and pointer classes.

Implements : DBASE04001

Table 272. Define PORT_APPL_CODE Description

Name	PORT_APPL_CODE
Initializer	

2.8.1.266 Define PORT_CALLOUT_CODE

PORT memory and pointer classes.

Implements : DBASE04001

Table 273. Define PORT_CALLOUT_CODE Description

Name	PORT_CALLOUT_CODE
Initializer	

2.8.1.267 Define PORT_VAR_NOINIT

PORT memory and pointer classes.

Implements : DBASE04001

Table 274. Define PORT_VAR_NOINIT Description

Name	PORT_VAR_NOINIT
Initializer	

2.8.1.268 Define PORT_VAR_POWER_ON_INIT

PORT memory and pointer classes.

Implements : DBASE04001

Table 275. Define PORT_VAR_POWER_ON_INIT Description

Name	PORT_VAR_POWER_ON_INIT
Initializer	

2.8.1.269 Define PORT_VAR_FAST

PORT memory and pointer classes.

Implements : DBASE04001

Table 276. Define PORT_VAR_FAST Description

Name	PORT_VAR_FAST
Initializer	

2.8.1.270 Define PORT_VAR

PORT memory and pointer classes.

Implements : DBASE04001

Table 277. Define PORT_VAR Description

Name	PORT_VAR

Table 277. Define PORT_VAR Description*...continued*

Initializer	

2.8.1.271 Define PWM_CODE

PWM memory and pointer classes.

Implements : DBASE04001

Table 278. Define PWM_CODE Description

Name	PWM_CODE
Initializer	

2.8.1.272 Define PWM_CONST

PWM memory and pointer classes.

Implements : DBASE04001

Table 279. Define PWM_CONST Description

Name	PWM_CONST
Initializer	

2.8.1.273 Define PWM_APPL_DATA

PWM memory and pointer classes.

Implements : DBASE04001

Table 280. Define PWM_APPL_DATA Description

Name	PWM_APPL_DATA
Initializer	

2.8.1.274 Define PWM_APPL_CONST

PWM memory and pointer classes.

Implements : DBASE04001

Table 281. Define PWM_APPL_CONST Description

Name	PWM_APPL_CONST
Initializer	

2.8.1.275 Define PWM_APPL_CODE

PWM memory and pointer classes.

Implements : DBASE04001

Table 282. Define PWM_APPL_CODE Description

Name	PWM_APPL_CODE
Initializer	

2.8.1.276 Define PWM_CALLOUT_CODE

PWM memory and pointer classes.

Implements : DBASE04001

Table 283. Define PWM_CALLOUT_CODE Description

Name	PWM_CALLOUT_CODE
Initializer	

2.8.1.277 Define PWM_VAR_NOINIT

PWM memory and pointer classes.

Implements : DBASE04001

Table 284. Define PWM_VAR_NOINIT Description

Name	PWM_VAR_NOINIT
Initializer	

2.8.1.278 Define PWM_VAR_POWER_ON_INIT

PWM memory and pointer classes.

Implements : DBASE04001

Table 285. Define PWM_VAR_POWER_ON_INIT Description

Name	PWM_VAR_POWER_ON_INIT
Initializer	

2.8.1.279 Define PWM_VAR_FAST

PWM memory and pointer classes.

Implements : DBASE04001

Table 286. Define PWM_VAR_FAST Description

Name	PWM_VAR_FAST
Initializer	

2.8.1.280 Define PWM_VAR

PWM memory and pointer classes.

Implements : DBASE04001

Table 287. Define PWM_VAR Description

Name	PWM_VAR
Initializer	

2.8.1.281 Define RAMTST_CODE

RamTST memory and pointer classes.

Implements : DBASE04001

Table 288. Define RAMTST_CODE Description

Name	RAMTST_CODE
Initializer	

2.8.1.282 Define RAMTST_CONST

RamTST memory and pointer classes.

Implements : DBASE04001

Table 289. Define RAMTST_CONST Description

Name	RAMTST_CONST
Initializer	

2.8.1.283 Define RAMTST_APPL_DATA

RamTST memory and pointer classes.

Implements : DBASE04001

Table 290. Define RAMTST_APPL_DATA Description

Name	RAMTST_APPL_DATA
Initializer	

2.8.1.284 Define RAMTST_APPL_CONST

RamTST memory and pointer classes.

Implements : DBASE04001

Table 291. Define RAMTST_APPL_CONST Description

Name	RAMTST_APPL_CONST
Initializer	

2.8.1.285 Define RAMTST_APPL_CODE

RamTST memory and pointer classes.

Implements : DBASE04001

Table 292. Define RAMTST_APPL_CODE Description

Name	RAMTST_APPL_CODE
Initializer	

2.8.1.286 Define RAMTST_CALLOUT_CODE

RamTST memory and pointer classes.

Implements : DBASE04001

Table 293. Define RAMTST_CALLOUT_CODE Description

Name	RAMTST_CALLOUT_CODE
Initializer	

2.8.1.287 Define RAMTST_VAR_NOINIT

RamTST memory and pointer classes.

Implements : DBASE04001

Table 294. Define RAMTST_VAR_NOINIT Description

Name	RAMTST_VAR_NOINIT
Initializer	

2.8.1.288 Define RAMTST_VAR_POWER_ON_INIT

RamTST memory and pointer classes.

Implements : DBASE04001

Table 295. Define RAMTST_VAR_POWER_ON_INIT Description

Name	RAMTST_VAR_POWER_ON_INIT
Initializer	

2.8.1.289 Define RAMTST_VAR_FAST

RamTST memory and pointer classes.

Implements : DBASE04001

Table 296. Define RAMTST_VAR_FAST Description

Name	RAMTST_VAR_FAST
Initializer	

2.8.1.290 Define RAMTST_VAR

RamTST memory and pointer classes.

Implements : DBASE04001

Table 297. Define RAMTST_VAR Description

Name	RAMTST_VAR
Initializer	

2.8.1.291 Define SCHM_CODE

SchM memory and pointer classes.

Implements : DBASE04001

Table 298. Define SCHM_CODE Description

Name	SCHM_CODE
Initializer	

2.8.1.292 Define SCHM_CONST

SchM memory and pointer classes.

Implements : DBASE04001

Table 299. Define SCHM_CONST Description

Name	SCHM_CONST
Initializer	

2.8.1.293 Define SCHM_APPL_DATA

SchM memory and pointer classes.

Implements : DBASE04001

Table 300. Define SCHM_APPL_DATA Description

Name	SCHM_APPL_DATA

All information provided in this document is subject to legal disclaimers.

Table 300. Define SCHM_APPL_DATA Description*...continued*

Initializer	

2.8.1.294 Define SCHM_APPL_CONST

SchM memory and pointer classes.

Implements : DBASE04001

Table 301. Define SCHM_APPL_CONST Description

Name	SCHM_APPL_CONST
Initializer	

2.8.1.295 Define SCHM_APPL_CODE

SchM memory and pointer classes.

Implements : DBASE04001

Table 302. Define SCHM_APPL_CODE Description

Name	SCHM_APPL_CODE
Initializer	

2.8.1.296 Define SCHM_CALLOUT_CODE

SchM memory and pointer classes.

Implements : DBASE04001

Table 303. Define SCHM_CALLOUT_CODE Description

Name	SCHM_CALLOUT_CODE
Initializer	

2.8.1.297 Define SCHM_VAR_NOINIT

SchM memory and pointer classes.

Implements : DBASE04001

Table 304. Define SCHM_VAR_NOINIT Description

Name	SCHM_VAR_NOINIT
Initializer	

2.8.1.298 Define SCHM_VAR_POWER_ON_INIT

SchM memory and pointer classes.

Implements : DBASE04001

Table 305. Define SCHM_VAR_POWER_ON_INIT Description

Name	SCHM_VAR_POWER_ON_INIT
Initializer	

2.8.1.299 Define SCHM_VAR_FAST

SchM memory and pointer classes.

Implements : DBASE04001

Table 306. Define SCHM_VAR_FAST Description

Name	SCHM_VAR_FAST
Initializer	

2.8.1.300 Define SCHM_VAR

SchM memory and pointer classes.

Implements : DBASE04001

Table 307. Define SCHM_VAR Description

Name	SCHM_VAR
Initializer	

2.8.1.301 Define SPI_CODE

SPI memory and pointer classes.

Implements : DBASE04001

Table 308. Define SPI_CODE Description

Name	SPI_CODE
Initializer	

2.8.1.302 Define SPI_CONST

SPI memory and pointer classes.

Implements : DBASE04001

Table 309. Define SPI_CONST Description

Name	SPI_CONST
Initializer	

2.8.1.303 Define SPI_APPL_DATA

SPI memory and pointer classes.

Implements : DBASE04001

Table 310. Define SPI_APPL_DATA Description

Name	SPI_APPL_DATA
Initializer	

2.8.1.304 Define SPI_APPL_CONST

SPI memory and pointer classes.

Implements : DBASE04001

Table 311. Define SPI_APPL_CONST Description

Name	SPI_APPL_CONST
Initializer	

2.8.1.305 Define SPI_APPL_CODE

SPI memory and pointer classes.

Implements : DBASE04001

Table 312. Define SPI_APPL_CODE Description

Name	SPI_APPL_CODE
Initializer	

2.8.1.306 Define SPI_CALLOUT_CODE

SPI memory and pointer classes.

Implements : DBASE04001

Table 313. Define SPI_CALLOUT_CODE Description

Name	SPI_CALLOUT_CODE
Initializer	

2.8.1.307 Define SPI_VAR_NOINIT

SPI memory and pointer classes.

Implements : DBASE04001

Table 314. Define SPI_VAR_NOINIT Description

Name	SPI_VAR_NOINIT
Initializer	

2.8.1.308 Define SPI_VAR_POWER_ON_INIT

SPI memory and pointer classes.

Implements : DBASE04001

Table 315. Define SPI_VAR_POWER_ON_INIT Description

Name	SPI_VAR_POWER_ON_INIT
Initializer	

2.8.1.309 Define SPI_VAR_FAST

SPI memory and pointer classes.

Implements : DBASE04001

Table 316. Define SPI_VAR_FAST Description

Name	SPI_VAR_FAST
Initializer	

2.8.1.310 Define SPI_VAR

SPI memory and pointer classes.

Implements : DBASE04001

Table 317. Define SPI_VAR Description

Name	SPI_VAR
Initializer	

2.8.1.311 Define WDG_CODE

WDG memory and pointer classes.

Implements : DBASE04001

Table 318. Define WDG_CODE Description

Name	WDG_CODE
Initializer	

2.8.1.312 Define WDG_CONST

WDG memory and pointer classes.

Implements : DBASE04001

Table 319. Define WDG_CONST Description

Name	WDG_CONST
Initializer	

2.8.1.313 Define WDG_APPL_DATA

WDG memory and pointer classes.

Implements : DBASE04001

Table 320. Define WDG_APPL_DATA Description

Name	WDG_APPL_DATA
Initializer	

2.8.1.314 Define WDG_APPL_CONST

WDG memory and pointer classes.

Implements : DBASE04001

Table 321. Define WDG_APPL_CONST Description

Name	WDG_APPL_CONST
Initializer	

2.8.1.315 Define WDG_APPL_CODE

WDG memory and pointer classes.

Implements : DBASE04001

Table 322. Define WDG_APPL_CODE Description

Name	WDG_APPL_CODE
Initializer	

2.8.1.316 Define WDG_CALLOUT_CODE

WDG memory and pointer classes.

Implements : DBASE04001

Table 323. Define WDG_CALLOUT_CODE Description

Name	WDG_CALLOUT_CODE

Table 323. Define WDG_CALLOUT_CODE Description*...continued*

Initializer	

2.8.1.317 Define WDG_VAR_NOINIT

WDG memory and pointer classes.

Implements : DBASE04001

Table 324. Define WDG_VAR_NOINIT Description

Name	WDG_VAR_NOINIT
Initializer	

2.8.1.318 Define WDG_VAR_POWER_ON_INIT

WDG memory and pointer classes.

Implements : DBASE04001

Table 325. Define WDG_VAR_POWER_ON_INIT Description

Name	WDG_VAR_POWER_ON_INIT
Initializer	

2.8.1.319 Define WDG_VAR_FAST

WDG memory and pointer classes.

Implements : DBASE04001

Table 326. Define WDG_VAR_FAST Description

Name	WDG_VAR_FAST
Initializer	

2.8.1.320 Define WDG_VAR

WDG memory and pointer classes.

Implements : DBASE04001

Table 327. Define WDG_VAR Description

Name	WDG_VAR
Initializer	

2.8.1.321 Define WDGIF_CODE

WDGIF memory and pointer classes.

Implements : DBASE04001

Table 328. Define WDGIF_CODE Description

Name	WDGIF_CODE
Initializer	

2.8.1.322 Define WDGIF_CONST

WDGIF memory and pointer classes.

Implements : DBASE04001

Table 329. Define WDGIF_CONST Description

Name	WDGIF_CONST
Initializer	

2.8.1.323 Define WDGIF_APPL_DATA

WDGIF memory and pointer classes.

Implements : DBASE04001

Table 330. Define WDGIF_APPL_DATA Description

Name	WDGIF_APPL_DATA
Initializer	

2.8.1.324 Define WDGIF_APPL_CONST

WDGIF memory and pointer classes.

Implements : DBASE04001

Table 331. Define WDGIF_APPL_CONST Description

Name	WDGIF_APPL_CONST
Initializer	

2.8.1.325 Define WDGIF_APPL_CODE

WDGIF memory and pointer classes.

Implements : DBASE04001

Table 332. Define WDGIF_APPL_CODE Description

Name	WDGIF_APPL_CODE
Initializer	

2.8.1.326 Define WDGIF_CALLOUT_CODE

WDGIF memory and pointer classes.

Implements : DBASE04001

Table 333. Define WDGIF_CALLOUT_CODE Description

Name	WDGIF_CALLOUT_CODE
Initializer	

2.8.1.327 Define WDGIF_VAR_NOINIT

WDGIF memory and pointer classes.

Implements : DBASE04001

Table 334. Define WDGIF_VAR_NOINIT Description

Name	WDGIF_VAR_NOINIT
Initializer	

2.8.1.328 Define WDGIF_VAR_POWER_ON_INIT

WDGIF memory and pointer classes.

Implements : DBASE04001

Table 335. Define WDGIF_VAR_POWER_ON_INIT Description

Name	WDGIF_VAR_POWER_ON_INIT
Initializer	

2.8.1.329 Define WDGIF_VAR_FAST

WDGIF memory and pointer classes.

Implements : DBASE04001

Table 336. Define WDGIF_VAR_FAST Description

Name	WDGIF_VAR_FAST
Initializer	

2.8.1.330 Define WDGIF_VAR

WDGIF memory and pointer classes.

Implements : DBASE04001

Table 337. Define WDGIF_VAR Description

Name	WDGIF_VAR
Initializer	

2.8.1.331 Define AUTOSAR_COMSTACKDATA

Define for ComStack Data.

Implements : DBASE04001

Table 338. Define AUTOSAR_COMSTACKDATA Description

Name	AUTOSAR_COMSTACKDATA
Initializer	

2.8.1.332 Define BUSTRCV_E_ERROR

Bus transceiver detected an unclassified error.

Details :

General return codes for BusTrcvErrorType

Implements : DBASE02012

Table 339. Define BUSTRCV_E_ERROR Description

Name	BUSTRCV_E_ERROR
Initializer	0x01

2.8.1.333 Define BUSTRCV_OK

There is no bus transceiver error seen or transceiver does not support the detection of bus errors.

Details :

General return codes for BusTrcvErrorType

Implements : DBASE02012

Table 340. Define BUSTRCV_OK Description

Name	BUSTRCV_OK
Initializer	0x00

2.8.1.334 Define COMSTACKTYPE_AR_RELEASE_MAJOR_VERSION

Table 341. Define COMSTACKTYPE_AR_RELEASE_MAJOR_VERSION Description

Name	COMSTACKTYPE_AR_RELEASE_MAJOR_VERSION
Initializer	4

2.8.1.335 Define COMSTACKTYPE_AR_RELEASE_MINOR_VERSION

Table 342. Define COMSTACKTYPE_AR_RELEASE_MINOR_VERSION Description

Name	COMSTACKTYPE_AR_RELEASE_MINOR_VERSION
Initializer	2

2.8.1.336 Define COMSTACKTYPE_AR_RELEASE_REVISION_VERSION

Table 343. Define COMSTACKTYPE_AR_RELEASE_REVISION_VERSION Description

Name	COMSTACKTYPE_AR_RELEASE_REVISION_VERSION
Initializer	2

2.8.1.337 Define COMSTACKTYPE_SW_MAJOR_VERSION

Table 344. Define COMSTACKTYPE_SW_MAJOR_VERSION Description

Name	COMSTACKTYPE_SW_MAJOR_VERSION
Initializer	1

2.8.1.338 Define COMSTACKTYPE_SW_MINOR_VERSION

Table 345. Define COMSTACKTYPE_SW_MINOR_VERSION Description

Name	COMSTACKTYPE_SW_MINOR_VERSION
Initializer	0

2.8.1.339 Define COMSTACKTYPE_SW_PATCH_VERSION

Table 346. Define COMSTACKTYPE_SW_PATCH_VERSION Description

Name	COMSTACKTYPE_SW_PATCH_VERSION
Initializer	5

2.8.1.340 Define COMSTACKTYPE_VENDOR_ID

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

Implements : DBASE02013

Table 347. Define COMSTACKTYPE_VENDOR_ID Description

Name	COMSTACKTYPE_VENDOR_ID
Initializer	43

2.8.1.341 Define NTFRSLT_E_ABORT

Flow control (FC) N_PDU with FlowStatus = OVFLW received.

Details :

General return codes for NotifResultType

Implements : DBASE02011

Table 348. Define NTFRSLT_E_ABORT Description

Name	NTFRSLT_E_ABORT
Initializer	0x09

2.8.1.342 Define NTFRSLT_E_CANCELATION_NOT_OK

Request cancellation has not been executed Due to an internal error the requested cancelation has not been executed. This will happen e.g. if the to be canceled transmission has been executed already.

Details :

General return codes for NotifResultType

Implements : DBASE02011

Table 349. Define NTFRSLT_E_CANCELATION_NOT_OK Description

Name	NTFRSLT_E_CANCELATION_NOT_OK
Initializer	0x0C

2.8.1.343 Define NTFRSLT_E_CANCELATION_OK

Requested cancellation has been executed.

Details :

General return codes for NotifResultType

Implements : DBASE02011

Table 350. Define NTFRSLT_E_CANCELATION_OK Description

Name	NTFRSLT_E_CANCELATION_OK
Initializer	0x0B

2.8.1.344 Define NTFRSLT_E_INVALID_FS

Invalid or unknown FlowStatus value has been received.

Details :

General return codes for NotifResultType

Implements : DBASE02011

Table 351. Define NTFRSLT_E_INVALID_FS Description

Name	NTFRSLT_E_INVALID_FS
Initializer	0x06

2.8.1.345 Define NTFRSLT_E_NO_BUFFER

Indicates an abort of a transmission.

Details :

General return codes for NotifResultType

Implements : DBASE02011

Table 352. Define NTFRSLT_E_NO_BUFFER Description

Name	NTFRSLT_E_NO_BUFFER
Initializer	0x0A

2.8.1.346 Define NTFRSLT_E_NOT_OK

Message not successfully received or sent out.

Details :

General return codes for NotifResultType

Implements : DBASE02011

Table 353. Define NTFRSLT_E_NOT_OK Description

Name	NTFRSLT_E_NOT_OK
Initializer	0x01

2.8.1.347 Define NTFRSLT_E_PARAMETER_NOT_OK

The request for the change of the parameter did not complete successfully.

Details :

General return codes for NotifResultType

Implements : DBASE02011

Table 354. Define NTFRSLT_E_PARAMETER_NOT_OK Description

Name	NTFRSLT_E_PARAMETER_NOT_OK
Initializer	0x0E

2.8.1.348 Define NTFRSLT_E_RX_ON

The parameter change request not executed successfully due to an ongoing reception.

Details :

General return codes for NotifResultType

Implements : DBASE02011

Table 355. Define NTFRSLT_E_RX_ON Description

Name	NTFRSLT_E_RX_ON
Initializer	0x0F

2.8.1.349 Define NTFRSLT_E_TIMEOUT_A

Timer N_Ar/N_As has passed its time-out value N_Asmax/N_Armax.

Details :

General return codes for NotifResultType

Implements : DBASE02011

Table 356. Define NTFRSLT_E_TIMEOUT_A Description

Name	NTFRSLT_E_TIMEOUT_A
Initializer	0x02

2.8.1.350 Define NTFRSLT_E_TIMEOUT_BS

Timer N_Bs has passed its time-out value N_Bsmax.

Details :

General return codes for NotifResultType

Implements : DBASE02011

Table 357. Define NTFRSLT_E_TIMEOUT_BS Description

Name	NTFRSLT_E_TIMEOUT_BS
Initializer	0x03

2.8.1.351 Define NTFRSLT_E_TIMEOUT_CR

Timer N_Cr has passed its time-out value N_Crmax.

Details :

General return codes for NotifResultType

Implements : DBASE02011

Table 358. Define NTFRSLT_E_TIMEOUT_CR Description

Name	NTFRSLT_E_TIMEOUT_CR
Initializer	0x04

2.8.1.352 Define NTFRSLT_E_UNEXP_PDU

Unexpected protocol data unit received.

Details :

General return codes for NotifResultType

Implements : DBASE02011

Table 359. Define NTFRSLT_E_UNEXP_PDU Description

Name	NTFRSLT_E_UNEXP_PDU
Initializer	0x07

2.8.1.353 Define NTFRSLT_E_VALUE_NOT_OK

The parameter change request not executed successfully due to a wrong value.

Details :

General return codes for NotifResultType

Implements : DBASE02011

Table 360. Define NTFRSLT_E_VALUE_NOT_OK Description

Name	NTFRSLT_E_VALUE_NOT_OK
Initializer	0x10

2.8.1.354 Define NTFRSLT_E_WFT_OVRN

Flow control WAIT frame that exceeds the maximum counter N_WFTmax received.

Details :

General return codes for NotifResultType

Implements : DBASE02011

Table 361. Define NTFRSLT_E_WFT_OVRN Description

Name	NTFRSLT_E_WFT_OVRN
Initializer	0x08

2.8.1.355 Define NTFRSLT_E_WRONG_SN

Unexpected sequence number (PCI.SN) value received.

Details :

General return codes for NotifResultType

Implements : DBASE02011

Table 362. Define NTFRSLT_E_WRONG_SN Description

Name	NTFRSLT_E_WRONG_SN
Initializer	0x05

2.8.1.356 Define NTFRSLT_OK

Action has been successfully finished.

Details :

General return codes for NotifResultType

Implements : DBASE02011

Table 363. Define NTFRSLT_OK Description

Name	NTFRSLT_OK
Initializer	0x00

2.8.1.357 Define NTFRSLT_PARAMETER_OK

The parameter change request has been successfully executed.

Details :

General return codes for NotifResultType

Implements : DBASE02011

Table 364. Define NTFRSLT_PARAMETER_OK Description

Name	NTFRSLT_PARAMETER_OK
Initializer	0x0D

2.8.1.358 Define CONSTP2FUNC

The compiler abstraction for const pointer to function.

Implements : DBASE05031

Table 365. Define CONSTP2FUNC Description

Name	CONSTP2FUNC
Initializer	rettype (* const fctname)

2.8.1.359 Define EXIT_INTERRUPT

Compiler abstraction for returning from an ISR if no OS is present.

Implements : DBASE05006

Table 366. Define EXIT_INTERRUPT Description

Name	EXIT_INTERRUPT
Initializer	SuspendAllInterrupts(); *((volatileuint32*)((uint32)INTC_BASEADDR + (uint32)INTC_EOIR_OFFSET)) = 0U

2.8.1.360 Define ISR

Compiler abstraction for creating an interrupt handler if no OS is present.

Implements : DBASE05016

Table 367. Define ISR Description

Name	ISR
Initializer	INTERRUPT_FUNC void IsrName(void)

2.8.1.361 Define MCAL_AR_RELEASE_MAJOR_VERSION

Table 368. Define MCAL_AR_RELEASE_MAJOR_VERSION Description

Name	MCAL_AR_RELEASE_MAJOR_VERSION
Initializer	4

2.8.1.362 Define MCAL_AR_RELEASE_MINOR_VERSION

Table 369. Define MCAL_AR_RELEASE_MINOR_VERSION Description

Name	MCAL_AR_RELEASE_MINOR_VERSION
Initializer	2

2.8.1.363 Define MCAL_AR_RELEASE_REVISION_VERSION

Table 370. Define MCAL_AR_RELEASE_REVISION_VERSION Description

Name	MCAL_AR_RELEASE_REVISION_VERSION
Initializer	2

2.8.1.364 Define MCAL_MODULE_ID

Table 371. Define MCAL_MODULE_ID Description

Name	MCAL_MODULE_ID
Initializer	0

2.8.1.365 Define MCAL_SW_MAJOR_VERSION

Table 372. Define MCAL_SW_MAJOR_VERSION Description

Name	MCAL_SW_MAJOR_VERSION
Initializer	1

2.8.1.366 Define MCAL_SW_MINOR_VERSION

Table 373. Define MCAL_SW_MINOR_VERSION Description

Name	MCAL_SW_MINOR_VERSION

Table 373. Define MCAL_SW_MINOR_VERSION Description*...continued*

Initializer	0

2.8.1.367 Define MCAL_SW_PATCH_VERSION

Table 374. Define MCAL_SW_PATCH_VERSION Description

Name	MCAL_SW_PATCH_VERSION
Initializer	5

2.8.1.368 Define MCAL_VENDOR_ID

Table 375. Define MCAL_VENDOR_ID Description

Name	MCAL_VENDOR_ID
Initializer	43

2.8.1.369 Define P2P2CONST

The compiler abstraction for pointer to pointer to constant.

Implements : DBASE05026

Table 376. Define P2P2CONST Description

Name	P2P2CONST
Initializer	const ptrtype **

2.8.1.370 Define P2P2VAR

The compiler abstraction for pointer to pointer to variable.

Implements : DBASE05025

Table 377. Define P2P2VAR Description

Name	P2P2VAR
Initializer	ptrtype **

2.8.1.371 Define ResumeAllInterrupts

Compiler abstraction for re-enabling all interrupts if no OS is present.

Implements : DBASE05020

Table 378. Define ResumeAllInterrupts Description

Name	ResumeAllInterrupts
Initializer	ASM_KEYWORD(" wrteei 1")

2.8.1.372 Define STATIC

The compiler abstraction shall provide the STATIC define for abstraction of compiler keyword static. Keep here for backward compatibility. It has been removed from ASR4.0.

Implements : DBASE05030

Table 379. Define STATIC Description

Name	STATIC
Initializer	static

2.8.1.373 Define SuspendAllInterrupts

Compiler abstraction for disabling all interrupts if no OS is present.

Implements : DBASE05021

Table 380. Define SuspendAllInterrupts Description

Name	SuspendAllInterrupts
Initializer	ASM_KEYWORD(" wrteei 0")

2.8.1.374 Define MEMMAP_VENDOR_ID

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

Implements : DBASE02002

Table 381. Define MEMMAP_VENDOR_ID Description

Name	MEMMAP_VENDOR_ID
Initializer	43

2.8.1.375 Define MEMMAP_AR_RELEASE_MAJOR_VERSION

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

Implements : DBASE02002

Table 382. Define MEMMAP_AR_RELEASE_MAJOR_VERSION Description

Name	MEMMAP_AR_RELEASE_MAJOR_VERSION
Initializer	4

2.8.1.376 Define MEMMAP_AR_RELEASE_MINOR_VERSION

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

Implements : DBASE02002

Table 383. Define MEMMAP_AR_RELEASE_MINOR_VERSION Description

Name	MEMMAP_AR_RELEASE_MINOR_VERSION
Initializer	2

2.8.1.377 Define MEMMAP_AR_RELEASE_REVISION_VERSION

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

Implements : DBASE02002

Table 384. Define MEMMAP_AR_RELEASE_REVISION_VERSION Description

Name	MEMMAP_AR_RELEASE_REVISION_VERSION
Initializer	2

2.8.1.378 Define MEMMAP_SW_MAJOR_VERSION

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

Implements : DBASE02002

Table 385. Define MEMMAP_SW_MAJOR_VERSION Description

Name	MEMMAP_SW_MAJOR_VERSION
Initializer	1

2.8.1.379 Define MEMMAP_SW_MINOR_VERSION

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

Implements : DBASE02002

Table 386. Define MEMMAP_SW_MINOR_VERSION Description

Name	MEMMAP_SW_MINOR_VERSION
Initializer	0

2.8.1.380 Define MEMMAP_SW_PATCH_VERSION

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

Implements : DBASE02002

Table 387. Define MEMMAP_SW_PATCH_VERSION Description

Name	MEMMAP_SW_PATCH_VERSION
Initializer	5

2.8.1.381 Define MEMMAP_ERROR

Symbol used for checking correctness of the includes.

Implements : DBASE02001

Table 388. Define MEMMAP_ERROR Description

Name	MEMMAP_ERROR
Initializer	

2.8.1.382 Define CPU_BIT_ORDER

Bit order on register level.

Implements : DBASE08017

Table 389. Define CPU_BIT_ORDER Description

Name	CPU_BIT_ORDER
Initializer	(MSB_FIRST)

2.8.1.383 Define CPU_BYTE_ORDER

The byte order on memory level shall be indicated in the platform types header file using the symbol CPU_BYTE_ORDER.

Implements : DBASE08018

Table 390. Define CPU_BYTE_ORDER Description

Name	CPU_BYTE_ORDER
Initializer	(HIGH_BYTE_FIRST)

2.8.1.384 Define CPU_TYPE

Processor type.

Implements : DBASE08019

Table 391. Define CPU_TYPE Description

Name	CPU_TYPE
Initializer	(CPU_TYPE_32)

2.8.1.385 Define CPU_TYPE_16

16bit Type Processor

Implements : DBASE08020

Table 392. Define CPU_TYPE_16 Description

Name	CPU_TYPE_16
Initializer	16

2.8.1.386 Define CPU_TYPE_32

32bit Type Processor

Implements : DBASE08021

Table 393. Define CPU_TYPE_32 Description

Name	CPU_TYPE_32
Initializer	32

2.8.1.387 Define CPU_TYPE_8

8bit Type Processor

Implements : DBASE08022

Table 394. Define CPU_TYPE_8 Description

Name	CPU_TYPE_8
Initializer	8

2.8.1.388 Define FALSE

Boolean false value.

Implements : DBASE08023

Table 395. Define FALSE Description

Name	FALSE
Initializer	0

2.8.1.389 Define HIGH_BYTE_FIRST

HIGH_BYTE_FIRST Processor.

Implements : DBASE08024

Table 396. Define HIGH_BYTE_FIRST Description

Name	HIGH_BYTE_FIRST

Table 396. Define HIGH_BYTE_FIRST Description*...continued*

Initializer	0

2.8.1.390 Define LOW_BYTE_FIRST

LOW_BYTE_FIRST Processor.

Implements : DBASE08025

Table 397. Define LOW_BYTE_FIRST Description

Name	LOW_BYTE_FIRST
Initializer	1

2.8.1.391 Define LSB_FIRST

LSB First Processor.

Implements : DBASE08026

Table 398. Define LSB_FIRST Description

Name	LSB_FIRST
Initializer	1

2.8.1.392 Define MSB_FIRST

MSB First Processor.

Implements : DBASE08027

Table 399. Define MSB_FIRST Description

Name	MSB_FIRST
Initializer	0

2.8.1.393 Define PLATFORM_AR_RELEASE_MAJOR_VERSION

Table 400. Define PLATFORM_AR_RELEASE_MAJOR_VERSION Description

Name	PLATFORM_AR_RELEASE_MAJOR_VERSION
Initializer	4

2.8.1.394 Define PLATFORM_AR_RELEASE_MINOR_VERSION

Table 401. Define PLATFORM_AR_RELEASE_MINOR_VERSION Description

Name	PLATFORM_AR_RELEASE_MINOR_VERSION
Initializer	2

2.8.1.395 Define PLATFORM_AR_RELEASE_REVISION_VERSION

Table 402. Define PLATFORM_AR_RELEASE_REVISION_VERSION Description

Name	PLATFORM_AR_RELEASE_REVISION_VERSION
Initializer	2

2.8.1.396 Define PLATFORM_SW_MAJOR_VERSION

Table 403. Define PLATFORM_SW_MAJOR_VERSION Description

Name	PLATFORM_SW_MAJOR_VERSION
Initializer	1

2.8.1.397 Define PLATFORM_SW_MINOR_VERSION

Table 404. Define PLATFORM_SW_MINOR_VERSION Description

Name	PLATFORM_SW_MINOR_VERSION
Initializer	0

2.8.1.398 Define PLATFORM_SW_PATCH_VERSION

Table 405. Define PLATFORM_SW_PATCH_VERSION Description

Name	PLATFORM_SW_PATCH_VERSION
Initializer	5

2.8.1.399 Define PLATFORM_VENDOR_ID

Table 406. Define PLATFORM_VENDOR_ID Description

Name	PLATFORM_VENDOR_ID
Initializer	43

2.8.1.400 Define TRUE

Boolean true value.

Implements : DBASE08035

Table 407. Define TRUE Description

Name	TRUE
Initializer	1

2.8.1.401 Define E_NOT_OK

Return code for failure/error.

Implements : DBASE12005

Table 408. Define E_NOT_OK Description

Name	E_NOT_OK
Initializer	0x01

2.8.1.402 Define E_OK

Success return code.

Implements : DBASE12004

Table 409. Define E_OK Description

Name	E_OK
Initializer	0x00

2.8.1.403 Define STATUSTYPEDEFINED

Because E_OK is already defined within OSEK, the symbol E_OK has to be shared. To avoid name clashes and redefinition problems, the symbols have to be defined in the following way (approved within implementation).

Table 410. Define STATUSTYPEDEFINED Description

Name	STATUSTYPEDEFINED
Initializer	

2.8.1.404 Define STD_ACTIVE

Logical state active.

Implements : DBASE12008

Table 411. Define STD_ACTIVE Description

Name	STD_ACTIVE
Initializer	0x01

2.8.1.405 Define STD_HIGH

Physical state 5V or 3.3V.

Implements : DBASE12006

Table 412. Define STD_HIGH Description

Name	STD_HIGH
Initializer	0x01

2.8.1.406 Define STD_IDLE

Logical state idle.

Implements : DBASE12009

Table 413. Define STD_IDLE Description

Name	STD_IDLE
Initializer	0x00

2.8.1.407 Define STD_LOW

Physical state 0V.

Implements : DBASE12007

Table 414. Define STD_LOW Description

Name	STD_LOW
Initializer	0x00

2.8.1.408 Define STD_OFF

OFF state.

Implements : DBASE12011

Table 415. Define STD_OFF Description

Name	STD_OFF
Initializer	0x00

2.8.1.409 Define STD_ON

ON State.

Implements : DBASE12010

Table 416. Define STD_ON Description

Name	STD_ON

All information provided in this document is subject to legal disclaimers.

Table 416. Define STD_ON Description*...continued*

Initializer	0x01

2.8.1.410 Define STD_TYPES_AR_RELEASE_MAJOR_VERSION

Table 417. Define STD_TYPES_AR_RELEASE_MAJOR_VERSION Description

Name	STD_TYPES_AR_RELEASE_MAJOR_VERSION
Initializer	4

2.8.1.411 Define STD_TYPES_AR_RELEASE_MINOR_VERSION

Table 418. Define STD_TYPES_AR_RELEASE_MINOR_VERSION Description

Name	STD_TYPES_AR_RELEASE_MINOR_VERSION
Initializer	2

2.8.1.412 Define STD_TYPES_AR_RELEASE_REVISION_VERSION

Table 419. Define STD_TYPES_AR_RELEASE_REVISION_VERSION Description

Name	STD_TYPES_AR_RELEASE_REVISION_VERSION
Initializer	2

2.8.1.413 Define STD_TYPES_SW_MAJOR_VERSION

Table 420. Define STD_TYPES_SW_MAJOR_VERSION Description

Name	STD_TYPES_SW_MAJOR_VERSION
Initializer	1

2.8.1.414 Define STD_TYPES_SW_MINOR_VERSION

Table 421. Define STD_TYPES_SW_MINOR_VERSION Description

Name	STD_TYPES_SW_MINOR_VERSION
Initializer	0

2.8.1.415 Define STD_TYPES_SW_PATCH_VERSION

Table 422. Define STD_TYPES_SW_PATCH_VERSION Description

Name	STD_TYPES_SW_PATCH_VERSION
Initializer	5

2.8.1.416 Define STD_TYPES_VENDOR_ID

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

Implements : DBASE12012, DBASE12013, DBASE12014, DBASE12015, DBASE12016, DBASE12017, DBASE12018

Table 423. Define STD_TYPES_VENDOR_ID Description

Name	STD_TYPES_VENDOR_ID
Initializer	43

2.8.2 Enum Reference

Enumeration of all constants supported by the driver are as per AUTOSAR BASE Driver software specification Version 4.2 Rev0002 .

2.8.2.1 Enumeration Can_ReturnType

Can_ReturnType.

Details :

CAN Return Types from Functions.

Implements : DCAN02414

Table 424. Enumeration Can_ReturnType Values

Name	Initializer	Description
CAN_OK	0U	Operation was ok executed.
CAN_NOT_OK		Operation was not ok executed.
CAN_BUSY		Operation was rejected because of busy state.

2.8.2.2 Enumeration Can_StateTransitionType

CAN State Modes of operation.

Details :

State transitions that are used by the function CAN_SetControllerMode().

Implements : DCAN02415

Table 425. Enumeration Can_StateTransitionType Values

Name	Initializer	Description
CAN_T_STOP	0U	CANIF_CS_STARTED -> CANIF_CS_STOPPED.
CAN_T_START		CANIF_CS_STOPPED -> CANIF_CS_STARTED.
CAN_T_SLEEP		CANIF_CS_STOPPED -> CANIF_CS_SLEEP.
CAN_T_WAKEUP		CANIF_CS_SLEEP -> CANIF_CS_STOPPED.

2.8.2.3 Enumeration CanIf_ControllerModeType

CanIf_ControllerModeType.

Details :

Operating modes of the CAN Controller and CAN Driver

Table 426. Enumeration CanIf_ControllerModeType Values

Name	Initializer	Description
CANIF_CS_UNINIT	0U	UNINIT mode.
CANIF_CS_SLEEP		SLEEP mode.
CANIF_CS_STARTED		STARTED mode.
CANIF_CS_STOPPED		STOPPED mode.

2.8.2.4 Enumeration Eth_FilterActionType

Action type for PHY address filtering.

Details :

The Enumeration type describes the action to be taken for the MAC address given in *PhysAddrPtr

Table 427. Enumeration Eth_FilterActionType Values

Name	Initializer	Description
ETH_ADD_TO_FILTER	0	Add address to the filter.
ETH_REMOVE_FROM_FILTER		Remove address.

2.8.2.5 Enumeration Eth_ModeType

The Ethernet controller mode.

Details :

This type is used to store the information whether the Ethernet controller is stopped or running.

Table 428. Enumeration Eth_ModeType Values

Name	Initializer	Description
ETH_MODE_DOWN	0	Controller is shut down.
ETH_MODE_ACTIVE		Controller is active.

2.8.2.6 Enumeration Eth_ReturnType

The Ethernet specific return type.

Details :

This return type informs about the function success/failure status.

Table 429. Enumeration Eth_ReturnType Values

Name	Initializer	Description
ETH_OK	0	Success.
ETH_E_NOT_OK		General failure.
ETH_E_NO_ACCESS		Ethernet hardware access failure.

2.8.2.7 Enumeration Eth_RxStatusType

The Ethernet reception status.

Details :

This status is returned by the`Eth_Receive()`function to indicate whether any frame has been received and if yes, whether there is any frame still waiting in the queue (for another`Eth_Receive()`call).

Table 430. Enumeration Eth_RxStatusType Values

Name	Initializer	Description
ETH_RECEIVED	0	A frame has been received and there are no more frames in the queue.
ETH_NOT_RECEIVED		No frames received.
ETH_RECEIVED_MORE_DATA_AVAILABLE		A frame received and at least another one in the queue detected.
ETH_RECEIVED_FRAMES_LOST		Ethernet frame has been received, some frames got lost.

2.8.2.8 Enumeration Eth_StateType

The Ethernet driver state.

Details :

A variable of this type holds the state of the Ethernet driver module. The driver is at the ETH_STATE_UNINIT at the beginning until the Eth_Init() function is called. The state remains equal to the ETH_STATE_INIT until the Eth_ControllerInit() function is called. Then the state is ETH_STATE_ACTIVE.

Table 431. Enumeration Eth_StateType Values

Name	Initializer	Description
ETH_STATE_UNINIT	0	The driver has not been initialized yet.
ETH_STATE_INIT		The driver has not been configured but the controller has not been initialized yet.
ETH_STATE_ACTIVE		The driver was initialized and the controller was configured.

2.8.2.9 Enumeration Fr_ChannelType

Details :

This type is used to select the channel.

Implements : DFR32001

Table 432. Enumeration Fr_ChannelType Values

Name	Initializer	Description
FR_CHANNEL_A	0x01U	Channel A only
FR_CHANNEL_B	0x02U	Channel B only
FR_CHANNEL_AB	0x03U	Both A and B channels

2.8.2.10 Enumeration Fr_ErrorModeType

Variables of this type are used for storage of FlexRay controller error mode.

Implements : DFR32009

Table 433. Enumeration Fr_ErrorModeType Values

Name	Initializer	Description
FR_ERRORMODE_ACTIVE	0U	Controller is synchronized to FlexRay cluster. Synchronization has not been lost more than gMaxWithoutClockCorrectionPassive cycles
FR_ERRORMODE_PASSIVE	1U	Controller has lost synchronization for less than gMaxWithoutClockCorrectionFatal cycles
FR_ERRORMODE_COMM_HALT	2U	FlexRay controller was stoped due to loss of synchronization.

2.8.2.11 Enumeration Fr_POCStateType

Details :

Variables of this type are used to store the POC:state of the controller.

Implements : DFR32007

Table 434. Enumeration Fr_POCStateType Values

Name	Initializer	Description
FR_POCSTATE_CONFIG	0U	POC:Config - controller can be configured
FR_POCSTATE_DEFAULT_CONFIG		POC:Default Config - controller is initialized and needs to be configured.
FR_POCSTATE_HALT		POC:Halt - controller is stopped.
FR_POCSTATE_NORMAL_ACTIVE		POC:Normal Active - controller is connected to the FlexRay cluster
FR_POCSTATE_NORMAL_PASSIVE		POC:Normal Passive - controller has problems with synchronization to the cluster.
FR_POCSTATE_READY		POC:Ready - controller is configured and ready to communicate
FR_POCSTATE_STARTUP		POC:Startup - controller is starting up the cluster or integrating to the cluster.
FR_POCSTATE_WAKEUP		POC:Wakeup - controller initiate Wakeup procedure

2.8.2.12 Enumeration Fr_RxLPduStatusType

Transmit resource status is stored to variable of this type.

Implements : DFR32003

Table 435. Enumeration Fr_RxLPduStatusType Values

Name	Initializer	Description
FR_RECEIVED	0U	LSdu has been received
FR_NOT_RECEIVED		LSdu has not been received.
FR_RECEIVED_MORE_DATA_AVAILABLE		LSdu has been received. More instances of this LPdu are available (FIFO usage)

2.8.2.13 Enumeration Fr_SlotModeType

This type is used to store the slot mode of the controller.

Details :

Covers FR506

Implements : DFR32008

Table 436. Enumeration Fr_SlotModeType Values

Name	Initializer	Description
FR_SLOTMODE_KEYSLOT	0U	FlexRay controller is transmitting only in single slot (key slot)
FR_SLOTMODE_ALL_PENDING		After end of the cycle controller will change slot mode to FR_ SLOTMODE_ALL
FR_SLOTMODE_ALL		FlexRay controller transmits in all configured slots

2.8.2.14 Enumeration Fr_StartupStateType

Details :

Variable of this type is used to query the FlexRay controller Startup state.

Implements : DFR32004

Table 437. Enumeration Fr_StartupStateType Values

Name	Initializer	Description
FR_STARTUP_UNDEFINED	0U	Unknown state - controller has not been in the Startup state since Default Configstate.
FR_STARTUP_COLDSTART_ LISTEN		Controller listens before it initiates CAS transmission.
FR_STARTUP_INTEGRATION_ COLDSTART_CHECK		Node is synchronizing to leading coldstarter.
FR_STARTUP_COLDSTART_JOIN		Controller has joined the leading coldstarter in transmission of startup frames
FR_STARTUP_COLDSTART_ COLLISION_RESOLUTION		Leading coldstarter transmits startup frames.
FR_STARTUP_COLDSTART_ CONSISTENCY_CHECK		Leading coldstarter checks if another node transmits startup frames
FR_STARTUP_INTEGRATION_ LISTEN		Integrating node (non-coldstarter) listens.
FR_STARTUP_INITIALIZE_ SCHEDULE		Schedule is being initialized
FR_STARTUP_INTEGRATION_ CONSISTENCY_CHECK		Controller looks for 4 pairs of startup frames to integrate into cluster.
FR_STARTUP_COLDSTART_GAP		Pause between two attempts of coldstart.

2.8.2.15 Enumeration Fr_TxLPduStatusType

Transmit resource status is stored to variable of this type.

Implements : DFR32005

Table 438. Enumeration Fr_TxLPduStatusType Values

Name	Initializer	Description
FR_TRANSMITTED	0U	No LPdu transmission is pending, LSdu has been transmitted
FR_NOT_TRANSMITTED		An LPdu transmission is pending, LSdu has not been transmitted.
FR_TRANSMITTED_CONFLICT		A transmission conflict has occurred.

2.8.2.16 Enumeration Fr_WakeupStatusType

Details :

Variable of this type is used to query the FlexRay controller Wakeup status.

Implements : DFR32006

Table 439. Enumeration Fr_WakeupStatusType Values

Name	Initializer	Description
FR_WAKEUP_UNDEFINED	0U	Unknown state - the controller has not been in POC:Wakeup state since it was in POC:Default Config.
FR_WAKEUP_RECEIVED_HEADER		Frame header was received during initial listen phase.
FR_WAKEUP_RECEIVED_WUP		Valid WUP was received during initial listen phase.
FR_WAKEUP_COLLISION_HEADER		Collision with frame header during WUP transmission.
FR_WAKEUP_COLLISION_WUP		Collision with another wakeup during WUP transmission.
FR_WAKEUP_COLLISION_UNKNOWN		Unknown collision during WUP transmission.
FR_WAKEUP_TRANSMITTED		WUP has been successfully transmitted.

2.8.2.17 Enumeration BufReq_ReturnType

Variables of this type are used to store the result of a buffer request.

Implements : DBASE02009

Table 440. Enumeration BufReq_ReturnType Values

Name	Initializer	Description
BUFREQ_OK	0	Buffer request accomplished successful.
BUFREQ_E_NOT_OK	1	Buffer request not successful. Buffer cannot be accessed.
BUFREQ_E_BUSY	2	Temporarily no buffer available. It's up the requestor to retry request for a certain time.
BUFREQ_E_OVFL	3	No Buffer of the required length can be provided.

2.8.2.18 Enumeration TpDataStateType

Variables of this type shall be used to store the state of TP buffer.

Implements : DBASE02010

Table 441. Enumeration TpDataStateType Values

Name	Initializer	Description
TP_DATACONF	0	Indicates that all data, that have been copied so far, are c confirmed and can be removed from the TP buffer.
TP_DATARETRY	1	Indicates that this API call shall copy already copied data in order to recover from an error.
TP_CONFPENDING	2	Indicates that the previously copied data must remain in the TP.
TP_NORETRY	3	Indicate that the copied transmit data can be removed from the buffer after it has been copied.

2.8.2.19 Enumeration TPParameterType

Specify the parameter to which the value has to be changed (BS or STmin)

Implements : DBASE02008

Table 442. Enumeration TPParameterType Values

Name	Initializer	Description
TP_STMIN	0	Separation Time.
TP_BS	1	Block Size.
TP_BC	2	Band width control parameter used in FlexRay transport protocol module.

2.8.2.20 Enumeration Lin_FrameCsModelType

Checksum models for the LIN Frame.

Details :

This type is used to specify the Checksum model to be used for the LIN Frame.

Implements : DLIN05031

Table 443. Enumeration Lin_FrameCsModelType Values

Name	Initializer	Description
LIN_ENHANCED_CS		Enhanced checksum model.
LIN_CLASSIC_CS		Classic checksum model.

2.8.2.21 Enumeration Lin_FrameResponseType

Frame response types.

Details :

This type is used to specify whether the frame processor is required to transmit the response part of the LIN frame.

Implements : DLIN05034

Table 444. Enumeration Lin_FrameResponseType Values

Name	Initializer	Description
LIN_MASTER_RESPONSE		Response is generated from this (master) node.
LIN_SLAVE_RESPONSE		Response is generated from a remote slave node.
LIN_SLAVE_TO_SLAVE		Response is generated from one slave to another slave. For the master the response will be anonymous, it does not have to receive the response.

2.8.2.22 Enumeration Lin_StatusType

LIN Frame and Channel states operation.

Details :

LIN operation states for a LIN channel or frame, as returned by the API service Lin_GetStatus(). part of the LIN frame.

Implements : DLIN05036

Table 445. Enumeration Lin_StatusType Values

Name	Initializer	Description
LIN_NOT_OK	0	Development or production error occurred.
LIN_TX_OK		Successful transmission.
LIN_TX_BUSY		Ongoing transmission (Header or Response).
LIN_TX_HEADER_ERROR		Erroneous header transmission such as:.
LIN_TX_ERROR		Erroneous transmission such as:.
LIN_RX_OK		Reception of correct response.
LIN_RX_BUSY		Ongoing reception: at least one response byte has been received, but the checksum byte has not been received.
LIN_RX_ERROR		Erroneous reception such as:.
LIN_RX_NO_RESPONSE		No response byte has been received so far. This is a mess !! Frame status is mixed with channel status but i kept it here only because of LIN168.

Table 445. Enumeration Lin_StatusType Values...*continued*

Name	Initializer	Description
LIN_OPERATIONAL		Normal operation;.
LIN_CH_SLEEP		Sleep mode operation;.

2.8.3 Function Reference

Functions of all functions supported by the driver are as per AUTOSAR BASE Driver software specification Version 4.2 Rev0002 .

2.8.4 Structs Reference

Data structures supported by the driver are as per AUTOSAR BASE Driver software specification Version 4.2 Rev0002 .

2.8.4.1 Structure Can_PduType

`Can_PduType.`

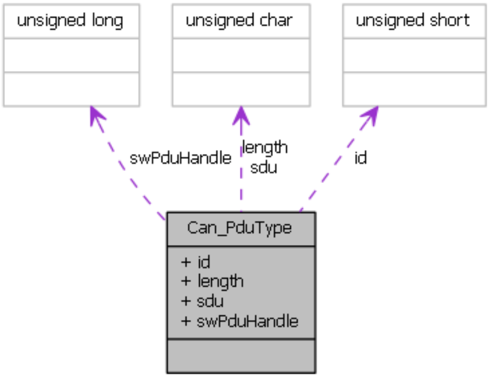


Figure 1. Struct Can_PduType

Details:

Type used to provide ID, DLC, SDU from CAN interface to CAN driver. HTH/HRH = ID +DLC+SDU.

Implements: DCAN02417

Declaration:

```
typedef struct
                      {
                        Can_IdType id,
                                uint8 length,
                                uint8* sdu,
                                PduIdType swPduHandle
                      } Can_PduType;
```

Table 446. Structure Can_PduType member description

Member	Description
id	CAN L-PDU = Data Link Layer Protocol Data Unit. Consists of Identifier, DLC and Data(SDU) It is uint32 for CAN_EXTENDEDID=STD_ON, else is uint16.

Table 446. Structure Can_PduType member description*...continued*

Member	Description
length	DLC = Data Length Code (part of L-PDU that describes the SDU length).
sdu	CAN L-SDU = Link Layer Service Data Unit. Data that is transported inside the L-PDU.
swPduHandle	The L-PDU Handle = defined and placed inside the CanIf module layer. Each handle represents an L-PDU, which is a constant structure with information for Tx/Rx processing.

2.8.4.2 Structure Fr_POCStatusType

Variables of this type are used to query the flexRay controller status.

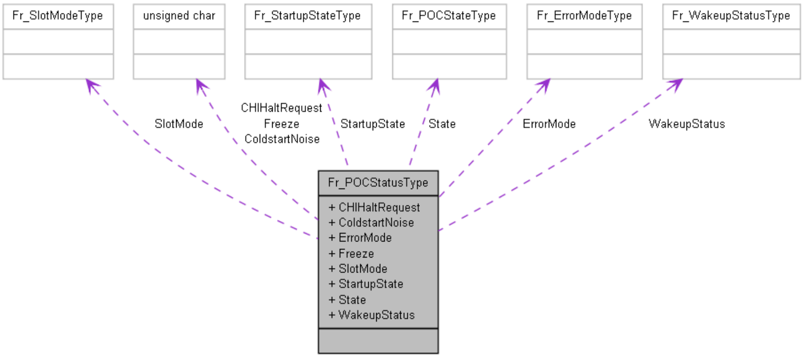


Figure 2. Struct Fr_POCStatusType

Implements : DFR32002

Declaration:

```
typedef struct
                     {
                       boolean CHIHaltRequest,
                               boolean ColdstartNoise,
                               Fr_ErrorModeType ErrorMode,
                               boolean Freeze,
                               Fr_SlotModeType SlotMode,
                               Fr_StartupStateType
 StartupState,
                               Fr_POCStateType State,
                               Fr_WakeupStatusType WakeupStatus
                     } Fr_POCStatusType;
```

Table 447. Structure Fr_POCStatusType member description

Member	Description
CHIHaltRequest	TRUE means that noise detected on bus during startup
ColdstartNoise	TRUE means that there is pending halt request
ErrorMode	TRUE means that internal error causing transition to the POC:Halt state or FREEZE command occurred
Freeze	Contains FlexRay controller slot mode
SlotMode	Contains FlexRay controller wakeup status
StartupState	Contains FlexRay controller error mode
State	Contains FlexRay controller startup state

Table 447. Structure Fr_POCStatusType member description*...continued*

Member	Description
WakeupStatus	Contains FlexRay controller POC state

2.8.4.3 Structure Lin_PduType

The LIN identifier (0..0x3F) with its parity bits.

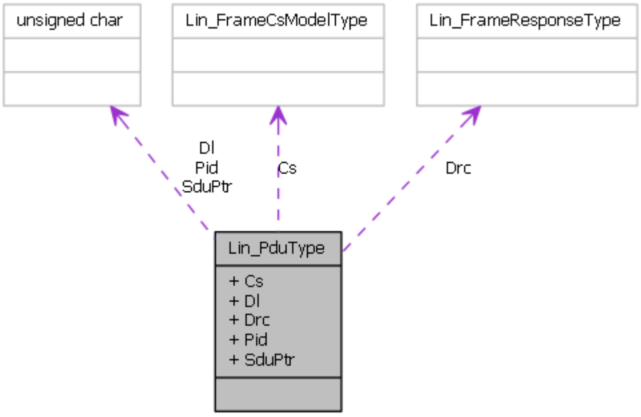


Figure 3. Struct Lin_PduType

Details:

This Type is used to provide PID, checksum model, data length and SDU pointer from the LIN Interface to the LIN driver.

Implements: DLIN05035

Declaration:

```
typedef struct
                          {
                            Lin_FrameCsModelType Cs,
                                      Lin_FrameDlType Dl,
                                      Lin_FrameResponseType Drc,
                                      Lin_FramePidType Pid,
                                      uint8* SduPtr
                          } Lin_PduType;
```

Table 448. Structure Lin_PduType member description

Member	Description
Cs	Checksum model type.
Dl	Data length.
Drc	Response type.
Pid	LIN frame identifier.
SduPtr	Pointer to Sdu.

2.8.4.4 Structure Mcal_DemErrorType

Typedef for DEM error management implemented by MCAL drivers.

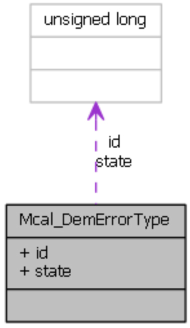


Figure 4. Struct Mcal_DemErrorType

Implements : DBASE05032

Declaration:

```
typedef struct
                          {
                            uint32 id,
                                      uint32 state
                          } Mcal_DemErrorType;
```

Table 449. Structure Mcal_DemErrorType member description

Member	Description
id	enabling/disabling the DEM error: Active=STD_ON/ Inactive=STD_OFF
state	ID of DEM error (0 if STD_OFF)

2.8.4.5 Structure PduInfoType

Variables of this type are used to store the basic information about a PDU of any type, namely a pointer variable pointing to it's SDU (payload), and the corresponding length of the SDU in bytes.

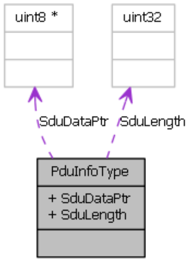


Figure 5. Struct PduInfoType

Implements : DBASE02006

Declaration:

```
typedef struct
                          {
                            uint8* SduDataPtr,
                                      PduLengthType SduLength
                          } PduInfoType;
```

Table 450. Structure PduInfoType member description

Member	Description
SduDataPtr	pointer to the SDU (i.e. payload data) of the PDU
SduLength	length of the SDU in bytes

2.8.4.6 Structure RetryInfoType

Variables of this type shall be used to store the information about Tp buffer handling.

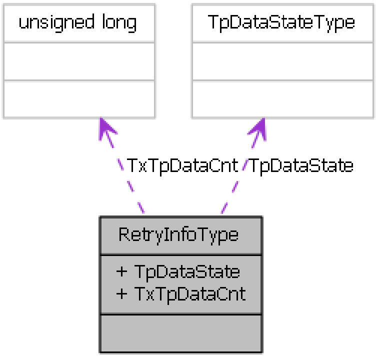


Figure 6. Struct RetryInfoType

Implements : DBASE02007

Declaration:

```
typedef struct
                        {
                          TpDataStateType TpDataState,
                                   PduLengthType TxTpDataCnt
                        } RetryInfoType;
```

Table 451. Structure RetryInfoType member description

Member	Description
TpDataState	The enum type to be used to store the state of Tp buffer
TxTpDataCnt	length of the SDU in bytes

2.8.4.7 Structure Std_VersionInfoType

This type shall be used to request the version of a BSW module using the "ModuleName"_GetVersionInfo() function.

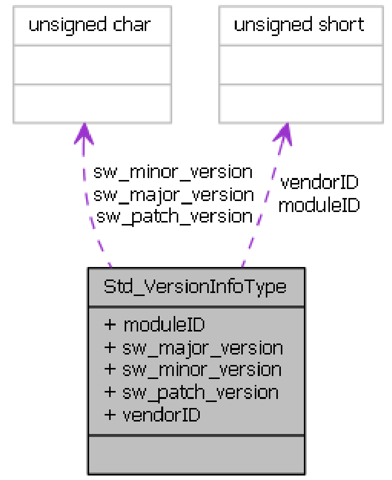


Figure 7. Struct Std_VersionInfoType

Implements : DBASE12003

Declaration:

```
typedef struct
                         {
                           uint16 moduleID,
                                     uint8 sw_major_version,
                                     uint8 sw_minor_version,
                                     uint8 sw_patch_version,
                                     uint16 vendorID
                         } Std_VersionInfoType;
```

Table 452. Structure Std_VersionInfoType member description

Member	Description
moduleID	0
sw_major_version	1
sw_minor_version	0
sw_patch_version	2
vendorID	43

2.8.5 Types Reference

Types supported by the driver are as per AUTOSAR BASE Driver software specification Version 4.2 Rev0002 .

2.8.5.1 Typedef Can_IdType

Can_IdType.

Details :

Type for storing the Identifier Length Type: Normal /Extended.

- used by "Can_MessageBufferConfigObjectType" structure. The driver does not distinguish between Extended and Mixed transmission modes. Extended transmission mode of operation behaves the same as Mixed mode.

Implements : DCAN02420

Type: `uint16`

2.8.5.2 Typedef Can_HwHandleType

Can_HwHandleType.

Details :

Represents the hardware object handles of a CAN hardware unit. For CAN hardware units with more than 255 HW objects use extended range.

- used by "Can_Write" function. The driver does not distinguish between Extended and Mixed transmission modes. Extended transmission mode of operation behaves the same as Mixed mode.

Implements : DCAN02421

Type: `uint16`

2.8.5.3 Typedef Eth_DataType

Type used to pass transmit/receive data to/from the driver.

Details :

This type was defined as 8 bit wide unsigned integer because this definition is available on all CPU types.

Type: `uint8`

2.8.5.4 Typedef Eth_FrameType

Frame type.

Details :

This type is used to pass the value of type/length field in the Ethernet frame header. It is 16 bits long unsigned integer.

- Values less than or equal to 1500 represent the length.
- Values grater than 1500 represent the type (i.e. 0x800 = IP).

Type: `uint16`

2.8.5.5 Typedef PduIdType

This type serve as a unique identifier of a PDU within a software module. Allowed ranges: uint8 .. uint16.

Implements : DBASE02002

Type: `uint32`

2.8.5.6 Typedef PduLengthType

This type serve as length information of a PDU in bytes. Allowed ranges: uint8 .. uint32.

Implements : DBASE02002

Type: `uint32`

2.8.5.7 Typedef BusTrcvErrorType

Variables of this type are used to return the bus status evaluated by a transceiver.

Implements : DBASE02005

Type: `uint8`

2.8.5.8 Typedef NetworkHandleType

Variables of the type NetworkHandleType are used to store the identifier of a communication channel.

Implements : DBASE02004

Type: `uint8`

2.8.5.9 Typedef NotifResultType

Variables of this type are used to store the result status of a notification (confirmation or indication).

Implements : DBASE02003

Type: `uint8`

2.8.5.10 Typedef Lin_FrameDlType

Data length of a LIN Frame.

Details :

This type is used to specify the number of SDU data bytes to copy.

Implements : DLIN05032

Type: `uint8`

2.8.5.11 Typedef Lin_FramePidType

The LIN identifier (0..0x3F) with its parity bits.

Details :

Represents all valid protected Identifier used by Lin_SendHeader().

Implements : DLIN05033

Type: `uint8`

2.8.5.12 Typedef boolean

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

Implements : DBASE08002

Type: unsigned char

2.8.5.13 Typedef float32

32bit long floating point data type

Implements : DBASE08015

Type: float

2.8.5.14 Typedef float64

64bit long floating point data type

Implements : DBASE08016

Type: double

2.8.5.15 Typedef sint16

Signed 16 bit integer with range of -32768 ..+32767 (0x8000..0x7FFF) - 15 bit + 1 sign bit.

Implements : DBASE08007

Type: signed short

2.8.5.16 Typedef sint16_least

Signed integer at least 16 bit long. Range - at least -32768 ..+32767. At least 15 bit + 1 bit sign.

Implements : DBASE08013

Type: signed long

2.8.5.17 Typedef sint32

Signed 32 bit integer with range of -2147483648.. +2147483647 (0x80000000..0x7FFFFFFF) - 31 bit + 1 sign bit.

Implements : DBASE08008

Type: signed long

2.8.5.18 Typedef sint32_least

Signed integer at least 32 bit long. Range - at least -2147483648.. +2147483647. At least 31 bit + 1 bit sign.

Implements : DBASE08014

Type: signed long

2.8.5.19 Typedef sint8

Signed 8 bit integer with range of -128 ..+127 (0x80..0x7F) - 7 bit + 1 sign bit.

Implements : DBASE08006

Type: signed char

2.8.5.20 Typedef sint8_least

Signed integer at least 8 bit long. Range - at least -128 ..+127. At least 7 bit + 1 bit sign.

Implements : DBASE08012

Type: signed long

2.8.5.21 Typedef uint16

Unsigned 16 bit integer with range of 0 ..+65535 (0x0000..0xFFFF) - 16 bit.

Implements : DBASE08004

Type: unsigned short

2.8.5.22 Typedef uint16_least

Unsigned integer at least 16 bit long. Range of at least 0 ..+65535 (0x0000..0xFFFF) - 16 bit.

Implements : DBASE08010

Type: unsigned long

2.8.5.23 Typedef uint32

Unsigned 32 bit integer with range of 0 ..+4294967295 (0x00000000..0xFFFFFFFF) - 32 bit.

Implements : DBASE08005

Type: unsigned long

2.8.5.24 Typedef uint32_least

Unsigned integer at least 32 bit long. Range of at least 0 ..+4294967295 (0x00000000..0xFFFFFFFF) - 32 bit.

Implements : DBASE08011

Type: unsigned long

2.8.5.25 Typedef uint8

Unsigned 8 bit integer with range of 0 ..+255 (0x00..0xFF) - 8 bit.

Implements : DBASE08003

Type: unsigned char

2.8.5.26 Typedef uint8_least

Unsigned integer at least 8 bit long. Range of at least 0 ..+255 (0x00..0xFF) - 8 bit.

Implements : DBASE08009

Type: unsigned long

2.8.5.27 Typedef StatusType

This type is defined for OSEK compliance.

Implements : DBASE12001

Type: unsigned char

2.8.5.28 Typedef Std_ReturnType

This type can be used as standard API return type which is shared between the RTE and the BSW modules.

Implements : DBASE12002

Type: `uint8`

2.9 Symbolic Names Disclaimer

All containers having the symbolic name tag set as true in the Autosar schema will generate defines like:

#define <Container_Short_Name> <Container_ID>

For this reason it is forbidden to duplicate the name of such containers across the MCAL configuration, or to use names that may trigger other compile issues (e.g. match existing #ifdefs arguments).

3 Tresos Configuration Plug-in

This chapter describes the Tresos configuration plug-in for the BASE Driver. The most of the parameters are described below.

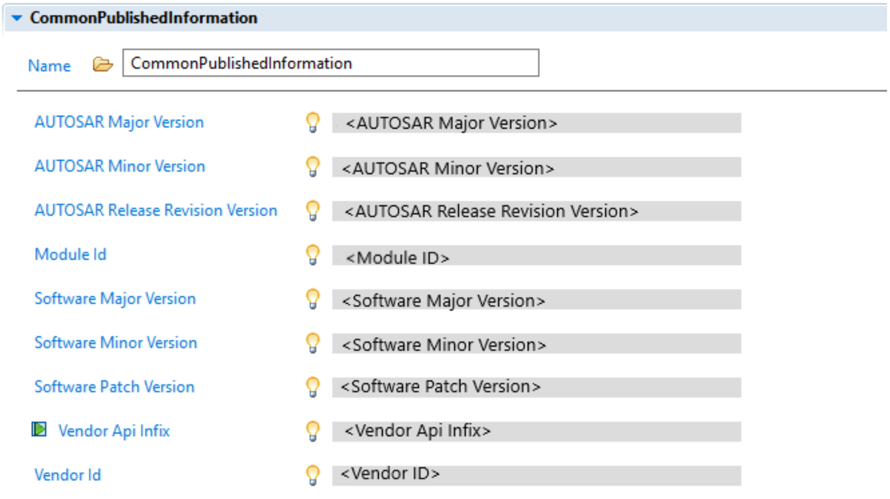
3.1 Configuration elements of Base

Included forms :

- Section 3.2 "Form CommonPublishedInformation"

3.2 Form CommonPublishedInformation

Common container, aggregated by all modules. It contains published information about vendor and versions.



CommonPublishedInformation

Name	CommonPublishedInformation
AUTOSAR Major Version	<AUTOSAR Major Version>
AUTOSAR Minor Version	<AUTOSAR Minor Version>
AUTOSAR Release Revision Version	<AUTOSAR Release Revision Version>
Module Id	<Module ID>
Software Major Version	<Software Major Version>
Software Minor Version	<Software Minor Version>
Software Patch Version	<Software Patch Version>
Vendor Api Infix	<Vendor Api Infix>
Vendor Id	<Vendor ID>

Figure 8. Tresos Plugin snapshot for CommonPublishedInformation form.

3.2.1 ArReleaseMajorVersion (CommonPublishedInformation)

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

Table 453. Attribute ArReleaseMajorVersion (CommonPublishedInformation) detailed description

Property	Value
Label	AUTOSAR Major Version
Type	INTEGER_LABEL
Origin	NXP
Symbolic Name	false
Default	4
Invalid	Range >=4 <=4

3.2.2 ArReleaseMinorVersion (CommonPublishedInformation)

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

Table 454. Attribute ArReleaseMinorVersion (CommonPublishedInformation) detailed description

Property	Value
Label	AUTOSAR Minor Version
Type	INTEGER_LABEL
Origin	NXP
Symbolic Name	false
Default	2
Invalid	Range >=2 <=2

3.2.3 ArReleaseRevisionVersion (CommonPublishedInformation)

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

Table 455. Attribute ArReleaseRevisionVersion (CommonPublishedInformation) detailed description

Property	Value
Label	AUTOSAR Release Revision Version
Type	INTEGER_LABEL
Origin	NXP
Symbolic Name	false
Default	2
Invalid	Range >=2 <=2

3.2.4 ModuleId (CommonPublishedInformation)

Module ID of this module from Module List.

Table 456. Attribute ModuleId (CommonPublishedInformation) detailed description

Property	Value
Label	Module Id
Type	INTEGER_LABEL
Origin	NXP
Symbolic Name	false
Default	0
Invalid	Range >=0 <=0

3.2.5 SwMajorVersion (CommonPublishedInformation)

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

Table 457. Attribute SwMajorVersion (CommonPublishedInformation) detailed description

Property	Value
Label	Software Major Version
Type	INTEGER_LABEL
Origin	NXP
Symbolic Name	false
Default	1
Invalid	Range >=1 <=1

3.2.6 SwMinorVersion (CommonPublishedInformation)

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

Table 458. Attribute SwMinorVersion (CommonPublishedInformation) detailed description

Property	Value
Label	Software Minor Version
Type	INTEGER_LABEL
Origin	NXP
Symbolic Name	false
Default	0
Invalid	Range >=0 <=0

3.2.7 SwPatchVersion (CommonPublishedInformation)

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

Table 459. Attribute SwPatchVersion (CommonPublishedInformation) detailed description

Property	Value
Label	Software Patch Version
Type	INTEGER_LABEL
Origin	NXP
Symbolic Name	false
Default	5
Invalid	Range >=5 <=5

3.2.8 VendorApiInfix (CommonPublishedInformation)

In driver modules which can be instantiated several times on a single ECU, BSW00347 requires that the name of APIs is extended by the VendorId and a vendor specific name. This parameter is used to specify the vendor specific name. In total, the implementation specific name is generated as follows: <ModuleName>_<VendorId>_<VendorApiInfix><Api name from SWS>. E.g. assuming that the VendorId of the implementor is 123 and the implementer chose a VendorApiInfix of "v11r456" a api name Can_Write defined in the SWS will translate to Can_123_v11r456Write. This parameter is mandatory for all modules with upper multiplicity > 1. It shall not be used for modules with upper multiplicity =1.

Table 460. Attribute VendorApiInfix (CommonPublishedInformation) detailed description

Property	Value
Label	Vendor Api Infix
Type	STRING_LABEL
Origin	NXP
Symbolic Name	false
Default	
Enable	false

3.2.9 VendorId (CommonPublishedInformation)

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

Table 461. Attribute VendorId (CommonPublishedInformation) detailed description

Property	Value
Label	Vendor Id
Type	INTEGER_LABEL
Origin	NXP
Symbolic Name	false
Default	43

Table 461. Attribute VendorId (CommonPublishedInformation) detailed description*...continued*

Property	Value
Invalid	Range >=43 <=43

4 Legal information

4.1 Definitions

Draft — A draft status on a document indicates that the content is still under internal review and subject to formal approval, which may result in modifications or additions. NXP Semiconductors does not give any representations or warranties as to the accuracy or completeness of information included in a draft version of a document and shall have no liability for the consequences of use of such information.

4.2 Disclaimers

Limited warranty and liability — Information in this document is believed to be accurate and reliable. However, NXP Semiconductors does not give any representations or warranties, expressed or implied, as to the accuracy or completeness of such information and shall have no liability for the consequences of use of such information. NXP Semiconductors takes no responsibility for the content in this document if provided by an information source outside of NXP Semiconductors.

In no event shall NXP Semiconductors be liable for any indirect, incidental, punitive, special or consequential damages (including - without limitation - lost profits, lost savings, business interruption, costs related to the removal or replacement of any products or rework charges) whether or not such damages are based on tort (including negligence), warranty, breach of contract or any other legal theory.

Notwithstanding any damages that customer might incur for any reason whatsoever, NXP Semiconductors' aggregate and cumulative liability towards customer for the products described herein shall be limited in accordance with the Terms and conditions of commercial sale of NXP Semiconductors.

Right to make changes — NXP Semiconductors reserves the right to make changes to information published in this document, including without limitation specifications and product descriptions, at any time and without notice. This document supersedes and replaces all information supplied prior to the publication hereof.

Suitability for use — NXP Semiconductors products are not designed, authorized or warranted to be suitable for use in life support, life-critical or safety-critical systems or equipment, nor in applications where failure or malfunction of an NXP Semiconductors product can reasonably be expected to result in personal injury, death or severe property or environmental damage. NXP Semiconductors and its suppliers accept no liability for inclusion and/or use of NXP Semiconductors products in such equipment or applications and therefore such inclusion and/or use is at the customer's own risk.

Applications — Applications that are described herein for any of these products are for illustrative purposes only. NXP Semiconductors makes no representation or warranty that such applications will be suitable for the specified use without further testing or modification.

Customers are responsible for the design and operation of their applications and products using NXP Semiconductors products, and NXP Semiconductors accepts no liability for any assistance with applications or customer product design. It is customer's sole responsibility to determine whether the NXP Semiconductors product is suitable and fit for the customer's applications and products planned, as well as for the planned application and use of customer's third party customer(s). Customers should provide appropriate design and operating safeguards to minimize the risks associated with their applications and products.

NXP Semiconductors does not accept any liability related to any default, damage, costs or problem which is based on any weakness or default in the customer's applications or products, or the application or use by customer's third party customer(s). Customer is responsible for doing all necessary testing for the customer's applications and products using NXP Semiconductors products in order to avoid a default of the applications and the products or of the application or use by customer's third party customer(s). NXP does not accept any liability in this respect.

Terms and conditions of commercial sale — NXP Semiconductors products are sold subject to the general terms and conditions of commercial sale, as published at http://www.nxp.com/profile/terms, unless otherwise agreed in a valid written individual agreement. In case an individual agreement is concluded only the terms and conditions of the respective agreement shall apply. NXP Semiconductors hereby expressly objects to applying the customer's general terms and conditions with regard to the purchase of NXP Semiconductors products by customer.

Export control — This document as well as the item(s) described herein may be subject to export control regulations. Export might require a prior authorization from competent authorities.

Suitability for use in non-automotive qualified products — Unless this data sheet expressly states that this specific NXP Semiconductors product is automotive qualified, the product is not suitable for automotive use. It is neither qualified nor tested in accordance with automotive testing or application requirements. NXP Semiconductors accepts no liability for inclusion and/or use of non-automotive qualified products in automotive equipment or applications.

In the event that customer uses the product for design-in and use in automotive applications to automotive specifications and standards, customer (a) shall use the product without NXP Semiconductors' warranty of the product for such automotive applications, use and specifications, and (b) whenever customer uses the product for automotive applications beyond NXP Semiconductors' specifications such use shall be solely at customer's own risk, and (c) customer fully indemnifies NXP Semiconductors for any liability, damages or failed product claims resulting from customer design and use of the product for automotive applications beyond NXP Semiconductors' standard warranty and NXP Semiconductors' product specifications.

Translations — A non-English (translated) version of a document is for reference only. The English version shall prevail in case of any discrepancy between the translated and English versions.

Security — Customer understands that all NXP products may be subject to unidentified vulnerabilities or may support established security standards or specifications with known limitations. Customer is responsible for the design and operation of its applications and products throughout their lifecycles to reduce the effect of these vulnerabilities on customer's applications and products. Customer's responsibility also extends to other open and/or proprietary technologies supported by NXP products for use in customer's applications. NXP accepts no liability for any vulnerability. Customer should regularly check security updates from NXP and follow up appropriately.

Customer shall select products with security features that best meet rules, regulations, and standards of the intended application and make the ultimate design decisions regarding its products and is solely responsible for compliance with all legal, regulatory, and security related requirements concerning its products, regardless of any information or support that may be provided by NXP.

NXP has a Product Security Incident Response Team (PSIRT) (reachable at PSIRT@nxp.com) that manages the investigation, reporting, and solution release to security vulnerabilities of NXP products.

4.3 Trademarks

Notice: All referenced brands, product names, service names, and trademarks are the property of their respective owners.

NXP — wordmark and logo are trademarks of NXP B.V.

Contents

All information provided in this document is subject to legal disclaimers.

Please be aware that important notices concerning this document and the product(s) described herein, have been included in section 'Legal information'.

For more information, please visit: http://www.nxp.com
For sales office addresses, please send an email to: salesaddresses@nxp.com

Date of release: 11 January 2022
Document identifier: AUTOSAR_MCAL_BASE_UM
Document number: UM2BASEASR4.2 Rev0002R1.0.5