Naming Rule

Nr. :   01

Title: Naming Convention

Contents

1. Introduction

The main goals of these rules are to:
- Standardise the way of naming so the "language" is the same for all the developers
- Minimise the adaptation period for new readers : the software is quickly understandable
- Clarify the writing / reading of the source file
- Clarify the architecture of the software

2. Macros and type definitions

The macros and type definitions will be written in capital letters. If possible try to minimize exported macros and constants for visibility.

Macros must use a prefix "CNF_" at the beginning if the macros will be used for configuration propouses.

To define the type of a variable, we used « typedef » to bypass the C compiler default size definition.

The following tables list the type used:

Type	Definition	Size	Range
u8	unsigned char	1	0..255
u16	unsigned short	2	0..65535
u32	unsigned long	4	0.. 4294967295
s8	signed char	1	-127.. 127
s16	short	2	-32767.. 32767
s32	long int	4	-2147483647.. 2147483647

Naming rules for variables

Variables must start with a prefix of their layer according with Autosar following with their Module name and then a short description of their function.

All variables will be written with an initial lower case letter

<AutosarLayer_AutosarModule_VariableName>

For example:
A new variable will be created for the Microcontroller Abstraction Layer in the module of Microcontroller Drivers.

mcal_mcu_<variable_description>

Naming rules for functions task

Function's names will be created using the Autosar's Module following with an underscore and a short description of the function. The first letter of every word must be upper case and the rest of the word lower case.

---

return <AutosarModule>_<Description>(params)

For example:
A new function is created for the Microcontroller Driver module of Autosar, the the function could be defined as the following:

return **M**cu_**V**ariable**N**ame(params)