|  |  |
| --- | --- |
| SFR  Standard function recovery | |
| **Summary** | This is the Software Detailed Design Document for the *DAIMLER MMA* Project. |

|  |  |  |
| --- | --- | --- |
| **Author** | **Review** | **Approval** |
| Title: M . Obada | Title1: M. Ianos  Title2: Name2 | Title1: D. Andris  Title2: Name2 |
|  |  |  |
| **Distribution** | | |
| Title1: A. Nerea  Title4: Name 4 | Title2: M.Obada  Title5: Name5 | Title3: Name3  Title6: Name6 |

# Table of content

1. General Information 3

1.1. Revision history \* 3

1.2. Purpose and Scope 3

1.3. Referenced documents 3

1.3.1. External documents 3

1.3.2. Internal Documents 3

1.4. Terminology and definitions 3

2. SW atomic architectural unit design 4

2.1. Overview 4

2.2. Traceability 4

2.3. Files structure 4

3. FEATURES 5

3.1. Services 5

3.1.1. Service Name 5

3.2. Types 6

3.2.1. Name Structure definition 6

3.3. Variables 6

3.4. Constants 6

4. EEPROM 6

5. Configuration 6

6. Compilation Options 6

# General Information

## Revision history \*

|  |  |  |  |
| --- | --- | --- | --- |
| **Revision** | **Date** | **Author(s)** | **Description/comment** |
| 001 | 28.03.2014 | S. Fléchelle | First review of document |
|  |  |  |  |
|  |  |  |  |

*\* Template history is found in the CM tool used for templates*

## Purpose and Scope

The review of this document is done thanks to …

The purpose of this document is to describe the design of the PP4G SFR Software Component.

## Referenced documents

### External documents

|  |  |  |
| --- | --- | --- |
| **Id** | **Title** | **Reference** |
|  |  |  |
|  |  |  |
|  |  |  |

### Internal Documents

|  |  |  |
| --- | --- | --- |
| **Id** | **Title** | **Reference** |
|  | SW architecture design interface description | E1355904 in PTC |
|  | ECU parameters specification | DOORS/PP4G/DES/  TF-J |
|  |  |  |
|  |  |  |

## Terminology and definitions

|  |  |
| --- | --- |
| **Terminology** | **Meaning** |
| AAU | Atomic architectural unit |
| MBD | Model Based Design |
| SFR | Standard Function Recovery algorithm |
| SW | software |
|  |  |

# SW atomic architectural unit design

## Overview

The aim of the “SFR” component is to decide to trigger and to decide to abort the SFR cycles:

* The anti-patina (or startup) cycle,
* The hard release cycle,
* The smooth release cycle,
* The backup release cycle.

The design of the “SFR” component is handled by the MBD technology, with the support of tools such as MATLAB environment and a code generator tool such as Target Link for instance.

## Traceability

The traceability matrix is built from the system specification until the architecture document.

Then, refer to [Doc1] the get the traceability against the functional, design and safety requirements, related to this unit.

## Files structure

Below is the description of the files structure defined for this unit.

**SFR\_StandardFunctionRecovery.h**

**SFR\_StandardFunctionRecovery.c**

**SFR\_StandardFunctionRecovery.c**

It will gather:

* The main function of SFR algorithm, called SFR\_runStandardFunctionRecovery (), which is the exported function in charge of providing:
  + the triggering decision of all available SFR cycles
  + the abortion decision of all available SFR cycles
  + the pre-conditions information of all available SFR cycles
  + the pre-activation information of all available SFR cycles
  + the triggering source information of all available SFR cycles.
* All the internal functions which provide the steps to decide to trigger and to abort all the SFR cycles.

**SFR\_StandardFunctionRecovery.h**

It will gather the definition of all exported constants and functions. Only the definition of the unique exported function is available: the SFR\_runStandardFunctionRecovery () runnable.

# FEATURES

The purpose of this chapter is to only describe the internal implementation of the component.

For the description of the external implementation, please refer to [Doc1] (to get the list of services, types, variables and constants exported by this unit).

Actually, the description of the internal implementation is not necessary for this SW unit since its complexity is extremely low. Therefore, the current chapter shall not be treated. Then, refer to C-code implementation directly.

## Services

### Service Name

The list and the description of services are available in [Doc1].

Refer to this document for more details.

|  |  |  |
| --- | --- | --- |
| **Object** | | |
|  | | |
| **Prototype** | | |
|  | | |
| **Input Parameters** | | |
| Name | Type | Description |
|  |  |  |
| **Output Parameters :** | | |
| Name | Type | Description |
|  |  |  |
| **Return value** | | |
| Type | Description | |
|  |  | |
| **Dynamic aspect** | | |
| Who (callers) | Description | |
|  |  | |
| **Static aspect** | | |
|  | | |
| **Constraints** | | |
|  | | |

/

## Types

### Name Structure definition

The list and the description of types are available in [Doc1].

Refer to this document for more details.

|  |  |  |
| --- | --- | --- |
| **Field Name** | **Field Type** | **Field description** |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

## Variables

The list and the description of variables are available in [Doc1].

Refer to this document for more details.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Name** | **Object** | **Type** | **Unit** | **Range** | **Dynamic Aspect** | **Constraints** |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |

## Constants

The list and the description of constants are available in [Doc1].

Refer to this document for more details.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Name** | **Object** | **Type** | **Value** | **Unit** | **Constraints** |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

# EEPROM

The EEPROM parameters are all specified in [Doc2].

Refer to this document for more details.

# Configuration

# Compilation Options