|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  | | |  |
|  |  | | |  |
|  |  | | |  |
|  | (F000058) | | |  |
|  |  | | |  |
|  |  | | |  |
| Document Type | **Feature Implementation Specification (FIS)** | | |  |
| Template Version | **6.1a** | | |  |
| Document ID | **fis\_map-dome lights\_1** | | |  |
| Document Location |  | | |  |
| Document Owner | **jgajjar** | | |  |
| Document Version | **0.1** | | |  |
| Document Status | **Draft** | | |  |
| Date Issued | **yyyy/mm/dd** | | |  |
| Date Revised | **yyyy/mm/dd** | | |  |
| Document Classification | GIS1 Item Number: | **27.60/35** | |  |
| GIS2 Classification: | **Confidential** | |
|  | | | | |
|  | | | | |
| Document Approval | | | | |
| Person | Role | | Email Confirmation | Date |
| Herta Llusho | Feature Supervisor | | [hllusho@ford.com](mailto:hllusho@ford.com) |  |
|  |  | |  |  |

This document contains Ford Motor Company Confidential information. Disclosure of the information contained in any portion of this document is not permitted without the expressed, written consent of a duly authorized representative of Ford Motor Company, Dearborn, Michigan, U.S.A.

Copyright © 2021, Ford Motor Company

Printed Copies Are Uncontrolled

**Important Note**

You need to use the RE specification macros provided by the “RE\_SpecificationMacroTemplate.dotm” (refer to “Utilities” on [page “Specification Templates” in the RE Wiki](http://wiki.ford.com/display/RequirementsEngineering/Specification+templates)) to allow seamless VSEM import of the specification content. **Use only these RE specification macros to create requirements** in this specification. Refer to “[How to use the Specification Templates](http://wiki.ford.com/display/RequirementsEngineering/How+to+use+the+Specification+Templates?src=contextnavpagetreemode)” on how to enable and use the macros and the requirements templates in this specification.

# Contents

[Contents 3](#_Toc73700643)

[1 Introduction 6](#_Toc73700644)

[1.1 Document Purpose 6](#_Toc73700645)

[1.2 Document Scope 6](#_Toc73700646)

[1.3 Document Audience 6](#_Toc73700647)

[1.3.1 Stakeholder List 6](#_Toc73700648)

[1.4 Document Organization 7](#_Toc73700649)

[1.4.1 Document Context 7](#_Toc73700650)

[1.4.2 Document Structure 7](#_Toc73700651)

[1.5 Document Conventions 7](#_Toc73700652)

[1.5.1 Requirements Templates 7](#_Toc73700653)

[1.5.1.1 Identification of requirements 7](#_Toc73700654)

[1.5.1.2 Requirements Attributes 8](#_Toc73700655)

[1.6 References 8](#_Toc73700656)

[1.6.1 Ford Documents 8](#_Toc73700657)

[1.6.2 External Documents and Publications 9](#_Toc73700658)

[1.7 Glossary 9](#_Toc73700659)

[1.7.1 Definitions 9](#_Toc73700660)

[1.7.2 Abbreviations 9](#_Toc73700661)

[2 Feature Implementation Overview 10](#_Toc73700662)

[2.1 Description 10](#_Toc73700663)

[2.2 Input Requirements/Documents 10](#_Toc73700664)

[2.3 Lessons Learned 11](#_Toc73700665)

[2.4 Assumptions 11](#_Toc73700666)

[3 Feature Implementation Architecture 12](#_Toc73700667)

[3.1 Functional Architecture 12](#_Toc73700668)

[3.1.1 Description 12](#_Toc73700669)

[3.1.2 Function List 15](#_Toc73700670)

[3.1.3 Signal List 16](#_Toc73700671)

[3.2 Physical Architecture 16](#_Toc73700672)

[3.2.1 E/E Architecture 16](#_Toc73700673)

[3.2.1.1 E/E Architecture Variants 16](#_Toc73700674)

[3.2.1.1.1 E/E Architecture “Architecture Variant 1” 17](#_Toc73700675)

[3.2.1.1.2 **E/E Architecture “Architecture Variant 2**” The Electrical Architecture Variant 2 is called as Reductive Switch Variant. In this variant, All Lights switches, and Door Defeat switch is eliminated and converted to soft touch in SYNC/PDC. Individual control of lights can be done via soft touch in SYNC as well as using physical switch. 20](#_Toc73700676)

[3.2.1.2 E/E Components 21](#_Toc73700677)

[3.2.1.3 E/E Connections 22](#_Toc73700678)

[3.2.1.4 Signal List 25](#_Toc73700679)

[3.2.1.5 Software Component Architecture 28](#_Toc73700680)

[3.2.1.6 Description 28](#_Toc73700681)

[3.3 Function Deployment 28](#_Toc73700682)

[3.3.1 Deployment Variants 28](#_Toc73700683)

[3.3.1.1 Deployment “Variant 1” 29](#_Toc73700684)

[3.3.2 Function Allocation 29](#_Toc73700685)

[4 Feature Implementation Modeling 31](#_Toc73700686)

[4.1 Component Interaction Diagrams 31](#_Toc73700687)

[4.1.1 Scenario: “System Startup / Shutdown” 31](#_Toc73700688)

[4.1.2 Scenario: “Normal Operation” 31](#_Toc73700689)

[4.2 Component Interface Behavior Diagrams 31](#_Toc73700690)

[5 Feature Implementation Requirements 32](#_Toc73700691)

[5.1 Functional Safety 32](#_Toc73700692)

[5.1.1 ASIL Decomposition of Technical Safety Requirements 32](#_Toc73700693)

[5.2 Requirements on Components 33](#_Toc73700694)

[5.2.1 Body Control Module 33](#_Toc73700695)

[5.2.1.1 Technology Function “MyLogicalFunctionA\_Component1” 33](#_Toc73700696)

[5.2.1.1.1 Function Interfaces 33](#_Toc73700697)

[5.2.1.1.1.1 Inputs 34](#_Toc73700698)

[5.2.1.1.1.2 Outputs 34](#_Toc73700699)

[5.2.1.1.1.3 Parameters 35](#_Toc73700700)

[5.2.1.1.1.4 Interface Requirements 35](#_Toc73700701)

[5.2.1.1.2 Function Requirements 36](#_Toc73700702)

[5.2.1.1.2.1 Component Specific Requirements 36](#_Toc73700703)

[5.2.1.2 (Technology) Function “MyLogicalFunctionB\_Component1” 36](#_Toc73700704)

[5.2.2 Phoenix Domain Controller (PDC) 37](#_Toc73700705)

[5.2.2.1.1.1 Inputs 37](#_Toc73700706)

[5.2.2.1.1.2 Outputs 37](#_Toc73700707)

[5.2.2.1.1.3 Parameters 37](#_Toc73700708)

[5.2.2.1.1.4 Interface Requirements 37](#_Toc73700709)

[5.2.2.1.2 Function Requirements 38](#_Toc73700710)

[5.2.2.1.2.1 Component Specific Requirements 38](#_Toc73700711)

[5.2.2.2 (Technology) Function “MyLogicalFunctionB\_Component1” 38](#_Toc73700712)

[5.2.3 Overhead Zone Module 39](#_Toc73700713)

[5.2.3.1.1.1 Inputs 39](#_Toc73700714)

[5.2.3.1.1.2 Outputs 39](#_Toc73700715)

[5.2.3.1.1.3 Parameters 39](#_Toc73700716)

[5.2.3.1.1.4 Interface Requirements 40](#_Toc73700717)

[5.2.3.1.2 Function Requirements 40](#_Toc73700718)

[5.2.3.1.2.1 Component Specific Requirements 41](#_Toc73700719)

[5.2.3.2 (Technology) Function “MyLogicalFunctionB\_Component1” 41](#_Toc73700720)

[5.3 Requirements on Connections 41](#_Toc73700721)

[5.3.1 Networks 41](#_Toc73700722)

[5.3.1.1 “CAN Bus xxx” 41](#_Toc73700723)

[5.3.1.1.1 Protocol Requirements 41](#_Toc73700724)

[5.3.1.1.2 Electrical Requirements 41](#_Toc73700725)

[5.3.1.2 “LIN Bus xxx” 41](#_Toc73700726)

[5.3.1.2.1 Protocol Requirements 41](#_Toc73700727)

[5.3.1.2.1.1 Schedule Table 41](#_Toc73700728)

[5.3.1.2.2 Electrical Requirements 42](#_Toc73700729)

[5.3.1.3 “Ethernet xxx” 42](#_Toc73700730)

[5.3.2 HW I/Os 42](#_Toc73700731)

[5.3.2.1 “HW I/O xxx” 42](#_Toc73700732)

[5.4 Requirements on Development Process 42](#_Toc73700733)

[6 Open Concerns 43](#_Toc73700734)

[7 Revision History 44](#_Toc73700735)

[8 Appendix 45](#_Toc73700737)

[8.1 Data Dictionary 45](#_Toc73700738)

[8.1.1 Logical Signals 45](#_Toc73700739)

[8.1.2 Logical Parameters 45](#_Toc73700740)

[8.1.3 Technical Signals 45](#_Toc73700741)

[8.1.3.1 GSDB Signals 45](#_Toc73700742)

[8.1.3.2 HW I/Os 45](#_Toc73700743)

[8.1.3.3 Diagnostic Interfaces 46](#_Toc73700744)

[8.1.3.3.1 DTCs 46](#_Toc73700745)

[8.1.3.3.2 DIDs 46](#_Toc73700746)

[8.1.4 Technical Parameters 46](#_Toc73700747)

[8.1.5 Mappings 46](#_Toc73700748)

[8.1.6 Technical Interfaces 46](#_Toc73700749)

[8.1.6.1 AIS Interfaces 46](#_Toc73700750)

[8.1.6.1.1 Publisher Interfaces 47](#_Toc73700751)

[8.1.6.1.2 Subscriber Interfaces 47](#_Toc73700752)

[8.1.6.2 AUTOSAR Ports 47](#_Toc73700753)

[8.1.7 Messages/APIs 47](#_Toc73700754)

[8.1.7.1 CAN Bus “<Bus Name>” 47](#_Toc73700755)

[8.1.7.2 LIN Bus “<Bus Name>” 47](#_Toc73700756)

[8.1.7.3 AUTOSAR Interfaces 47](#_Toc73700757)

[8.1.7.4 SOA Service Contracts 47](#_Toc73700758)

[8.1.8 Encoding Types 48](#_Toc73700759)

**List of Figures**

[Figure 3‑1: Functional Architecture 13](#_Toc35000653)

[Figure 3‑2 E/E Architecture (SysML Style) 18](#_Toc35000654)

[Figure 3‑3 E/E Architecture (Network Topology Style) 19](#_Toc35000655)

[Figure 3‑4: AUTOSAR compliant SW Component Architecture 28](#_Toc35000656)

**List of Tables**

[Table 1‑1: Electrical Architecture(s) referenced in this document 6](#_Toc73700379)

[Table 2 List of Stakeholders 6](#_Toc73700380)

[Table 1‑3: Ford internal Documents 8](#_Toc73700381)

[Table 1‑4: External documents and publications 9](#_Toc73700382)

[Table 1‑5: Definitions used in this document 9](#_Toc73700383)

[Table 1‑6: Abbreviations used in this document. 9](#_Toc73700384)

[Table 7: Input Requirements/Documents 11](#_Toc73700385)

[Table 3‑1: List of Functions 15](#_Toc73700386)

[Table 3‑1.3: List of Logical Signal 16](#_Toc73700387)

[Table 3‑2: Electrical Components 22](#_Toc73700388)

[Table 3‑3: E/E Connections 24](#_Toc73700389)

[Table 3‑4: Sample Deployment Diagram 29](#_Toc73700390)

[Table 3‑5: Function Allocation Table (Basic) 30](#_Toc73700391)

[Table 5‑1: ASIL Decomposition Table 33](#_Toc73700392)

[Table 5‑2: Input Signal mappings of Function “MyLogicalFunctionA\_Component1” 34](#_Toc73700393)

[Table 5‑3: Output Signal mappings of Function “MyLogicalFunctionA\_Component1”A 35](#_Toc73700394)

[Table 5‑4: Parameter mappings of Function “MyLogicalFunctionA\_Component1” 35](#_Toc73700395)

[Table 5‑5: Component Specific Requirements 36](#_Toc73700396)

[Table 5‑6: Inherited Requirements 36](#_Toc73700397)

[Table 5‑2: Input Signal mappings of Function “MyLogicalFunctionA\_Component1” 37](#_Toc73700398)

[Table 5‑3: Output Signal mappings of Function “MyLogicalFunctionA\_Component1”A 37](#_Toc73700399)

[Table 5‑4: Parameter mappings of Function “MyLogicalFunctionA\_Component1” 37](#_Toc73700400)

[Table 5‑5: Component Specific Requirements 38](#_Toc73700401)

[Table 5‑6: Inherited Requirements 38](#_Toc73700402)

[Table 5‑2: Input Signal mappings of Function “MyLogicalFunctionA\_Component1” 39](#_Toc73700403)

[Table 5‑3: Output Signal mappings of Function “MyLogicalFunctionA\_Component1”A 39](#_Toc73700404)

[Table 5‑4: Parameter mappings of Function “MyLogicalFunctionA\_Component1” 40](#_Toc73700405)

[Table 5‑5: Component Specific Requirements 41](#_Toc73700406)

[Table 5‑6: Inherited Requirements 41](#_Toc73700407)

[Table 6‑1: Open Concerns 43](#_Toc73700408)

# Introduction

## Document Purpose

The Feature Implementation Specification (FIS) specifies the deployment of the logical functions of a feature to an electrical architecture. The FIS specifies all interactions between the ECUs of the electrical architecture required for the feature including the technical signals and the interfaces. It also gives interface and integration requirements, which are specific to the feature for the electrical architecture.

To get more information about the concept of feature, function and component level abstraction refer to the [Ford RE Wiki](http://wiki.ford.com/display/RequirementsEngineering/Requirements+Engineering+for+SW+Enabled+Features).

## Document Scope

***#Hint:*** *The FIS can be used to document multiple deployment variants (refer to chapters “Deployment Variants” and “E/E Architecture Variants”). It is however recommended (except for small features) to have a separate FIS for each variant, because managing multiple variants in the same document easily gets complex and cumbersome.*

***#Functional Safety:*** *For Functional Safety specify only one deployment variant per FIS.*

This FIS describes the deployment of the feature <Overhead Lighting (Map/Dome Lights)> to the following electrical architecture(s):

| **Electrical Architecture Name** | **Owner** | **Reference** |
| --- | --- | --- |
| FNV3 |  | <Add VSEM Link> |
|  |  |  |

Table 1‑1: Electrical Architecture(s) referenced in this document

## Document Audience

The FIS is authored by <Jayneel Gajjar / Feature Owner>. All Stakeholders, i.e., all people who have a valid interest in the feature implementation should read and, if possible, review the FIS. It needs to be guaranteed, that all stakeholders have access to the currently valid version of the FIS.

***#Hint:*** *The FIS template has the IP Classification “Proprietary” by default. IP Classification “Confidential” might be required in some cases, e.g. by Ford Functional Safety.*

***#Macro:***[*Add Ins -> Edit Document Properties macro*](http://wiki.ford.com/display/RequirementsEngineering/Editing+Specification+Properties)

### Stakeholder List

For the latest list of the function stakeholders and their roles & responsibilities refer to <Put VSEM Link here>.

***#Hint:*** *Refer to* [*Ford RE Wiki – Stakeholder List*](http://wiki.ford.com/display/RequirementsEngineering/Stakeholder+Analysis) *on how to create a stakeholder list. The stakeholder list should be stored in VSEM in the pseudo folder “General Data Artifacts” of the corresponding feature / function / component.*

Table List of Stakeholders

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | |  |  |  |  |  |
| **Name** | **CDSID/phone** | **Description of Stake** | **Contact date** | **Elicitation response** | **Review worksheet** | **Review meeting** |
| Herta Llusho | hllusho | Core Feature Supervisor |  |  |  |  |
| Ken Cunningham | kcunni16 | Body Functional Safety |  |  |  |  |
| Steven Antilla | santilla | Interior Lighting Supervisor |  |  |  |  |
| Dave Bergen | dbergen5 | Interior Lighting D&R |  |  |  |  |
| Elizabeth Wickey | ewickey | Interior Lighting Electrical |  |  |  |  |
| Jason Emrich | jemrich | Optics Engineer |  |  |  |  |
| Bill Crafts | wcrafts1 | OZM D&R |  |  |  |  |
| James Baker | jbake286 | OZM D&R |  |  |  |  |
| Mark Dewitt | Mdewitt9 | BCM D&R |  |  |  |  |
| John Barrs | jbarrs | BCM Software |  |  |  |  |
| Brinda Ganesan | fbrinda | CIED Engineer |  |  |  |  |
| Demetrius Gault | Djohn840 | Sync Wireframe Engineer |  |  |  |  |
| Trupti Masurkar | Tmasurka | SYNC Product Engineer |  |  |  |  |
| Neelima Majjiga | Nmajjiga | SYNC Engineer |  |  |  |  |

## Document Organization

### Document Context

Refer to the [Specification Structure page](http://wiki.ford.com/display/RequirementsEngineering/Specification+templates) in the [Ford RE Wiki](http://wiki.ford.com/display/RequirementsEngineering/Requirements+Engineering+for+SW+Enabled+Features) to understand how the FIS relates to other Ford Requirements Documents and Specifications.

### Document Structure

The structure of this document is explained below:

**Section 1** – Introduction – Giving an explanation how to use this document including responsibilities and the scope of the document. Additionally, it contains the revision history and a list of unsettled but known issues that have to be consolidated in future versions. It explains the terminology and gives a clarification of the definitions, concepts and abbreviations used in the document.

**Section 2** – Feature Implementation Description – Giving an overview of the platform and listing assumptions, constraints or dependencies

**Section 3** – Feature Implementation Architecture – Describing 3 Architecture Views:

* Functional Architecture – Showing the logical architecture of functions
* Physical Architecture – Showing the physical architecture (first of all the E/E Architecture), which the Logical Functions get allocated to.
* Software Architecture – Showing the software architecture relevant for the feature (for features with in-house development only)
* Function Deployment – Presenting the allocation of logical functions and signals to the electrical and other components

**Section 4** – Deployment Specific Modeling –Modeling techniques providing additional detail on e.g. interface behavior

**Section 5** – Deployment Specific Requirements – Deployment specific requirements for ECUs, Network Communication, and Process

**Section 6** – List of Open Concerns

**Section 7** – Revision History

**Section 8** – Appendix - Presenting additional data mainly in a tabular form, e.g., a data dictionary

## Document Conventions

### Requirements Templates

Refer to “[How to use the Specification Templates](http://wiki.ford.com/display/RequirementsEngineering/How+to+use+the+Specification+Templates?src=contextnavpagetreemode)” on how to use the specification templates and the VBA macros to create/edit the requirements in the specifications.

The VBA macro enable the import of the specification to VSEM (refer to ["How to import specifications into VSEM as separate requirements"](http://wiki.ford.com/pages/viewpage.action?pageId=104991616&src=contextnavpagetreemode)).

#### Identification of requirements

The unique requirement ID given in the headline of any requirement follows the requirement throughout the development process. The requirement ID format follows a well-defined syntax.

All identifiers in an FIS shall be composed of 4 parts:

* A leading prefix, which indicates the type of requirement (R=Requirement, UC=Use Case, SC=Scenario, …)
* A prefix, which indicates the abstraction level (F=Feature, FNC=Function, CMP = component).
* Followed by a name, indicating the scope, which the requirement belongs to (e.g. feature or function name )
* Ending with the actual requirement number

*Example:*

*R\_CMP\_LockArbitrator\_00004* This is the fourth requirement on component level for the function Lock Arbitrator.

#### Requirements Attributes

Additionally attributes can be added to each requirement. This helps to classify requirements. A [list of available attributes](http://wiki.ford.com/display/RequirementsEngineering/Requirements+Attributes?src=contextnavpagetreemode) is given in the RE Wiki.

## References

### Ford Documents

The list of all Ford internal documents, which are directly related.

| **Reference** | **Title** | **Doc. ID** | **Revision** | **Document Location** |
| --- | --- | --- | --- | --- |
|  | Internal Lighting Specification | **RQT-170200-020520** |  | <https://www.fede.ford.com/awc/#/teamcenter.search.search?searchCriteria=ID:RQT-170200-020520&filter=SE4_Req_StandardRevision.se4_RequirementState%3DReleased~~Categorization.category%3DRequirements~~WorkspaceObject.object_type%3DSE4_RequirementRevision> |
|  | Feature Document: Map/Dome Light | **VDOC085753** |  |  |
|  | Functional Specification: Map/Dome Light | **VDOC089586** |  |  |
|  | BCM Functional Specification | **FS-MU5T-14B476-AAB002** |  |  |

Table 1‑3: Ford internal Documents

### External Documents and Publications

The list of external documents could include books, reports and online sources.

***#Hint:*** *You may refer to* [*IEEE Citation Reference*](http://www.ieee.org/documents/ieeecitationref.pdf) *on how to format a reference.*

| **Reference** | **Document / Publication** |
| --- | --- |
| [bbb] |  |
|  |  |

Table 1‑4: External documents and publications

## Glossary

### Definitions

| **Definition** | **Description** |
| --- | --- |
|  |  |
|  |  |
|  |  |

Table 1‑5: Definitions used in this document

### Abbreviations

| **Abbr.** | **Stands for** | **Description** |
| --- | --- | --- |
| FS | Functional Safety |  |
| E/E | Electrical and Electronics |  |
|  |  |  |
|  |  |  |

Table 1‑6: Abbreviations used in this document.

# Feature Implementation Overview

## Description

**#Hint:** Give a short overview on what E/E systems / platforms the feature is implemented and what special considerations must be taken into account for E/E systems / platforms.

Overhead Lights (Map/Dome Light) illuminate the interior of the vehicle. This feature can either be manually activated in global mode i.e. all lights on/off or local mode i.e. individual lights on/off based on the customer input. The lights can be in DD Enable or DD Disable Mode. These lights undergo Courtesy and Embrace Ramp On/OFF in DD Disable Mode.

There are two variants for activation:

1. **Physical Switch:** Customer input via physical switch located in OHC for Global Control and Individual switches for Map light in OHC along with second and third row based on the vehicle segment. Moreover, Door Defeat Mode switch is also available on OHC.
2. **Reductive Switch Design:** Customer input to Global On/Off and Door Defeat using SYNC screen.Dual Input availability for Map lights or local activation through sync as well as through physical switch.

Legacy Map/Dome Light have Smart Master Smart Slave configuration. The light intensity is not configurable.

## Input Requirements/Documents

**#Hint:** *The table below helps the feature owner to collect relevant input* (requirements, documents, mails, models, …) *while writing the spec. When finalizing the spec, the feature owner should check, if all inputs have been properly considered by derived/outgoing requirements* in chapter “Feature Implementation Requirements” *or the architectural elements.*

*Note:* It is not required to list each input requirement individually in this table, referencing the input document is enough (if relevant document section is indicated).

|  |  |  |  |
| --- | --- | --- | --- |
| **Reference**  (Reference as listed in ch. “References”) | **Section/Requirement** | **Description** | **Derived Requirement**  (optional – reference to requirement in ch. “Feature Implementation Requirements”) |
| **Feature/Function Requirements** | | | |
| Feature Document: Map/Dome Light | **VDOC085753** |  |  |
| Functional Specification: Map/Dome Light | **VDOC089586** |  |  |
| **Ford Engineering Standards** | | | |
| Internal Lighting Specification | **RQT-170200-020520** |  |  |
|  |  |  |  |
| **Legal Regulations** | | | |
| FMVSS 101 | **REG-130101-003390** |  |  |
|  |  |  |  |
| **Industry Standards** | | | |
|  | <Example: some ISO/IEEE/SAE or other standard> |  |  |
|  |  |  |  |
| **Other Sources** | | | |
|  | <Example: some stakeholder document> |  |  |
|  |  |  |  |

Table 7: Input Requirements/Documents

## Lessons Learned

**#Classification**: Optional

**#Hint:** Additional information and lessons learned from previous development or related features. A typical source for Lessons Learned is the FMA Quality History.

## Assumptions

**#Classification**: Optional

**#Hint:** A list of assumptions concerning the effects/dependencies of the feature’s deployment as well as (e.g. known limitations). During the feature development most of those assumptions are typically either converted into actual requirements or discarded at some point – such that this chapter ideally remains mostly empty.

# Feature Implementation Architecture

## Functional Architecture

**#Hint**: This section depicts the Functional Architecture, i.e., the decomposition into Logical Functions. This architectural step is needed to find the right functional partitioning for the function level.

### Description

**#Hint**: Provide some informal description of the characteristics of the chosen Functional Architecture. Also give some graphical representation of the Functional Architecture. Either SysML activity diagrams or [Data Flow Diagrams](http://wiki.ford.com/display/RequirementsEngineering/Data+Flow+Diagram?src=contextnavpagetreemode) could be used to depict such a Functional Architecture.

**#Link:** [*SysML - Activity Diagrams*](https://pd3.spt.ford.com/sites/SystemsEngineering/SEC/sysml-teamsite/SysML%20Wiki/Activity%20Diagram%20Basics.aspx) or [*RE Wiki - Data Flow Diagrams*](http://wiki.ford.com/display/RequirementsEngineering/Data+Flow+Diagram?src=contextnavpagetreemodehttp://wiki.ford.com/display/RequirementsEngineering/Data+Flow+Diagram?src=contextnavpagetreemode)

Figure 3‑1: Functional Architecture Variant 1: LIN



Figure Functional Architecture Variant 2: RSD

### Function List

The following functions from the [Global Feature & Function List](https://www.vsemweb.ford.com:443/tc/launchapp?-attach=true&-s=226TCSession&-o=ZmZNi0JHx3NrTDAAAAAAAAAAAAA) are referenced in this Feature Implementation Specification:

|  |  |  |
| --- | --- | --- |
| **Function ID** | **Function Name** | **Function Description** |
| F1 | Control Local On/Off | Local activation/deactivation of Map light through HMI via Sync Screen and Physical switch. |
| F2 | Control Global On/Off | Activation/deactivation of all Map/Dome light through HMI via Sync Screen or Physical Switch based on the variant. |
| F3 | Control Door Defeat | DD Enable/Disable of Map/Dome light through HMI via Sync Screen or Physical Switch based on the variant. |
| F4 | Control Cargo Defeat | Cargo Defeat Enable or Disable via Physical switch. |
| F5 | Provide Door Ajar Status (***Enclosures***) | Communicates Door/ Liftgate Ajar or Closed Status. |
| F6 | Provide Ignition Status (**Electronics Engine Controls**) | Communicates Ignition Arbitrator status (Off, Acc, Run, Start) |
| F7 | Provide Vehicle (***Brake Controls***) | Communicates Vehicle speed data in kph |
| F8 | Provide Vehicle Lock/Unlock (***Power Locking/Unlocking***) | Communicates whether vehicle is locked or unlocked. |
| F9 | Battery Saver (***Electrical Power Management***) | Deactivate lights to prevent battery drain if user forgets to turn Off lights and ignition is in Off state. |
| F10 | Provide Perimeter Alarm Courtesy Request (***Contents Protection***) | Perimeter Alarm status signals light to Turn On /OFF. |
| F11 | Provide Post-Crash Alert (***Restraint Electronics***) | Provides post-crash alert signals to turn lights On. |
| F12 | Provide silent mode or police dark car status (***Police Features***) | Silent Mode or Police Dark car mode blocks courtesy and welcome/farewell input to lamps. Lamps can turn On only when requested. |
| F13 | Provide Vehicle Availability Status (Variant 1) | Provide update bit for vehicle speed available. |
| F14 | Provide Park Status (Variant 1) | Provide vehicle park status |
| F15 | Provide Day Night Status (Variant 1) | Provide day and night status |
| F16 | Provide Dimming Level  (Variant 1) | Provide different dimming level for daytime and night time for backlighting. |
| F17 | Arbitration | Arbitration commands filters the signal and generates welcome/farewell output. Moreover, it also routes remaining signals to Control Light function. |
| F18 | Control Lights | Conducts the final arbitration of all the input signal received from arbitration function. Generates Global, Local, Door Defeat and Ramp On/OFF command. |
| F19 | Map/Dome Lights | Provide Light Output. |

Table 3‑1: List of Functions

### Signal List

*#Hint: Refer to the Data Dictionary - Logical Signals.*

|  |  |
| --- | --- |
| **Signal Name** | **Description** |
| Global On/Off | Global/All Lights signal from user will either turn ON or turn OFF all Overhead Lights. |
| Local On/Off | Local Signal from user will turn ON or turn OFF individual Overhead Light. |
| Door Defeat Enable/Disable | Door Defeat signal from the user will either enable or disable Door Defeat. |
| Cargo Defeat  Enable/Disable | Cargo Defeat signal from the user will either enable or disable Cargo Defeat. |
| Door Ajar Status | Provide cabin/cargo/liftgate door ajar status |
| Ignition Status | Provide vehicle ignition status |
| Lock/Unlock | Provide vehicle lock and unlock status |
| Vehicle Speed | Provide vehicle speed value in KPH. |
| Vehicle Speed Available | Provide status check whether vehicle speed is available or unavailable. |
| Battery Saver | Provide command for turn Off lights if the user forgets to turn them OFF while exiting the vehicle and ignition is OFF. |
| Perimeter Alarm Courtesy Request | Triggers Overhead light to turn ON if the Perimeter Alarm gets activated. |
| Crash Courtesy Request | Triggers Overhead light to turn On post-crash. |
| Silent Mode | Silent mode or Dark Car Mode restricts lights to turn on during courtesy or w/f. It only allows lights to turn on/off when requested by user. |
| Park Status | Provide status whether vehicle is in park mode or not |
| Day Night Status | Provide Day and Night status. |
| Dimming Level | Provide different dimming level for day and night-time for backlighting. |

Table 3‑2.3: List of Logical Signal

## Physical Architecture

### E/E Architecture

#### E/E Architecture Variants

**#Classification:** Mandatory – State “No Variants defined”, if not used.

**#Hint:** If different variants of the E/E architecture are specified in this section, list those variants in the table below.

Variants can be expressed based on Variant Options. Typical Variant Options (think of them as Logical Parameters) driven by architecture are e.g. “Network Topology” (e.g. FNV2 or AV) or “Powertrain Type” (e.g. “Electrical Vehicle” vs. “conventional powertrain”. “Conventional Powertrain” might be further split into “ECM+TCM” vs. “PCM”). The optional column “Variant condition” allows to express the dependency of a Variant based on Variant Options/Logical Parameters.

If requirements or certain architectural elements (signals, interfaces, components) are not applicable for all variants, those requirements/elements should state explicitly, which E/E architecture variant they apply to.

**#Link:** [RE Wiki – Variant Management](http://wiki.ford.com/display/RequirementsEngineering/Variant+Management).

|  |  |  |
| --- | --- | --- |
| E/E Architecture Variant Name | Variant Description | Variant Condition (optional) |
| **Variant 1** Physical Switch | Physical switch for activation/deactivation All lights, Door Defeat and Individual Lights. |  |
| **Variant 2** Reductive Switch Design | Soft touch button in SYNC for activation/deactivation of All lights and Door Defeat. Dual activation for individual lights via Sync as well as physical switch. |  |

##### E/E Architecture “Architecture Variant 1”

**#Classification:** Mandatory

**#Hint:** Place a diagram of high-level E/E architecture here. Optionally the allocated functions could be shown in the diagram. Either use some SysML like diagram (refer to Figure 3‑3 E/E Architecture (SysML Style)) or enhanced the network topology, which Netcom generates from its master Excel sheet diagrams (refer to Figure 3‑4 E/E Architecture (Network Topology Style)) according to the needs of the feature.

This E/E Architecture variant 1 considers activation of Overhead Lights through physical switches located on Overhead Console. Under this architecture, the OHC receives the LIN communication from BCM and OHC controls the global activation, Door Defeat, Courtesy and Cargo Defeat via Serial Bus Communication to the lamp. Individual lamp activation takes place through physical switch as well.



Figure 3‑3 E/E Architecture (SysML Style)



Figure 3‑4 E/E Architecture (Network Topology Style) : Variant 1

##### **E/E Architecture “Architecture Variant 2**” The Electrical Architecture Variant 2 is called as Reductive Switch Variant. In this variant, All Lights switches, and Door Defeat switch is eliminated and converted to soft touch in SYNC/PDC. Individual control of lights can be done via soft touch in SYNC as well as using physical switch.



Figure E/E Architecture (Network Topology Style) : Variant 2

#### E/E Components

#Hint: Use component name/acronym as given in the [VSEM Global Core ECU & EE Devices Dictionary](https://www.vsemweb.ford.com/tc/launchapp?-attach=true&-s=226TCSession&-o=BleFgEP3x3NrTDAAAAAAAAAAAAA&servername=Production_Server) If not listed in that database, you may use the use PSF naming convention of the [EDAS signal database in VSEM](https://www.vsemweb.ford.com/tc/launchapp?-attach=true&-s=226TCSession&-o=gPXpSoIbx3NrTDAAAAAAAAAAAAA&servername=Production_Server).  
You may directly link to the corresponding VSEM entry. Refer to the examples below”.

#Links: [*PSF Translate*](https://pd3.spt.ford.com/sites/EESEC3P/PSF_Translate/SitePages/Home.aspx)r (a little utility to search for an EDAS component name in PSF notation)

|  |  |
| --- | --- |
| Component Name | **Description** |
| [BCM](https://www.vsemweb.ford.com/tc/launchapp?-attach=true&-s=226TCSession&-o=ZHZRgzTmx3NrTDAAAAAAAAAAAAA&servername=Production_Server) | Body Control Module |
| [ABS](https://www.vsemweb.ford.com/tc/launchapp?-attach=true&-s=226TCSession&-o=lkUh05pex3NrTDAAAAAAAAAAAAA&servername=Production_Server) | Anti-Lock Braking System |
| [PDC](https://www.vsemweb.ford.com/tc/launchapp?-attach=true&-s=226TCSession&-o=TITtpNBux3NrTDAAAAAAAAAAAAA&servername=Production_Server) | Accessory Protocol Interface Module |
| [ECG](https://www.vsemweb.ford.com/tc/launchapp?-attach=true&-s=226TCSession&-o=D3Sp5v3vx3NrTDAAAAAAAAAAAAA&servername=Production_Server) | Gateway Module |
| [PDB](https://www.vsemweb.ford.com/tc/launchapp?-attach=true&-s=226TCSession&-o=odWNN0U6x3NrTDAAAAAAAAAAAAA&servername=Production_Server) | Power Distribution Box |

Table 3‑3: Electrical Components

#### E/E Connections

#Hint: Lists the E/E connections relevant for the feature and - for network connections - which *Messages* from the *Data Dictionary* are allocated to them. The ‘Connection Type’ is derived from the [*GDT/EDAS Signal Classification*](https://pd3.spt.ford.com/sites/fede/vsem-spls/Shared%20Documents/13-gdt/training/ppt/Signal_Classifications_v6.ppt?web=1). The ‘Protocol’ selection list might not be complete, yet. Add your protocol definition, if needed.

#Links: - [*GDT/EDAS Signal Classification*](https://pd3.spt.ford.com/sites/fede/vsem-spls/Shared%20Documents/13-gdt/training/ppt/Signal_Classifications_v6.ppt?web=1) (as reference for ‘Connection Type’ below)

* [*PSF Translate*](https://pd3.spt.ford.com/sites/EESEC3P/PSF_Translate/SitePages/Home.aspx)r (a little utility to search for an EDAS signal name in PSF notation)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Connection Name | **Connection Type** | **Protocol**  Only if ‘Connection Type’ is “Network”/” RF-Digital” | **Description** | **Allocated Messages**  Only if ‘Connection Type’ is “Network”/” RF-Digital” | **Connected Nodes** |
| All Lights Request  (Variant 2) | Network | CAN (High Speed) | Request All Lights On/Off from SYNC. | OverheadLamps\_All\_Rq | <PDC to ECG> |
| Door Defeat Request  (Variant 2) | Network | CAN (High Speed) | Request Door Defeat Enable/Disable from SYNC. | DrDefeat\_D\_Rq | <PDC to ECG> |
| Individual Map Light Request  (Variant 2) | Network | CAN (High Speed) | Request Individual Map/Dome lamps turn on/off. | MaplampFl\_D\_Rq  MaplampFl\_D\_Rq  Maplamp2Rl\_D\_Rq  Maplamp2Rr\_D\_Rq  Maplamp3Rl\_D\_Rq  Maplamp3Rr\_D\_Rq | <PDC to ECG> |
| All Lights Request  (Variant 2) | Network | CAN FD | Request All Lights On/Off from SYNC. | OverheadLamps\_All\_Rq | <ECG to OZM> |
| Door Defeat Request  (Variant 2) | Network | CAN FD | Request Door Defeat Enable/Disable from SYNC. | DrDefeat\_D\_Rq | <ECG to OZM> |
| Individual Map Light Request  (Variant 2) | Network | CAN FD | Request Individual Map/Dome lamps turn on/off. | MaplampFl\_D\_Rq  MaplampFl\_D\_Rq  Maplamp2Rl\_D\_Rq  Maplamp2Rr\_D\_Rq  Maplamp3Rl\_D\_Rq  Maplamp3Rr\_D\_Rq | <ECG to OZM> |
| Feedback to Sync (Variant 2) | Network | CAN FD | Feedback to Door Defeat Status to Sync screen. | DrDefeat\_D\_Stat | <OZM to ECG> |
| Feedback to Sync (Variant 2) | Network | CAN FD | Feedback to Door Defeat Status to Sync screen. | DrDefeat\_D\_Stat | <OZM to ECG> |
| Vehicle Speed Request  (Variant 2) | Network | CAN FD | Communicate vehicle speed in kph. | Veh\_V\_Actlbrk | <ABS to OZM> |
| Door Ajar Request (Variant 2) | Network | CAN FD | Provide Door Ajar status for Fr, Fl, Rr, Rl vehicle door. | DrStatDrv\_B\_Actl DrStatPsngr\_B\_Actl DrStatRl\_B\_Actl DrStatRr\_B\_Actl | <BCM to OZM> |
| Liftgate Ajar Request(Variant 2) | Network | CAN FD | Provide Liftgate Ajar status. | DrStatTgate\_B\_Actl | <BCM to OZM> |
| Alarm Courtesy Request(Variant 2) | Network | CAN FD | Trigger lights to Turn ON when perimeter alarm turns On. | PerimeterAlarm\_Courtesy\_Rq | <BCM to OZM> |
| Crash Courtesy Request(Variant 2) | Network | CAN FD | Trigger lights to Turn On post-crash. | Crash\_Courtesy\_Rq | <BCM to OZM> |
| Silent Mode/Dark Mode(Variant 2) | Network | CAN FD | Provide Silent Mode Status. | SilentMode\_Rq | <BCM to OZM> |
| BCM Configuration (Variant 2) | Network | CAN FD | Provide detail about BCM configuration status. | BcmNotConfigured\_Cntrl | <BCM to OZM> |
| Welcome Farewell Mode Request (Variant 2) | Network | CAN FD | Provide Welcome Farewell Master Mode Request. | VehWlcmFrwl\_D\_Stat | <BCM to OZM> |
| Welcome Farewell Substate Request (Variant 2) | Network | CAN FD | Provide Welcome Farewell Sub Mode Request | VehWlcmFrwlMde\_D\_Stat | <BCM to OZM> |
| Ignition Status (Variant 2) | Network | CAN FD | Provide Ignition Status | Ignition\_Status | <BCM to OZM> |
| All Lights On/OFF Request (Variant 2) | Network | LIN | Global On/OFF command | <Give a list of relevant messages > | <OZM to Lights> |
| Door Defeat Enable/Disable Request (Variant 2) | Network | LIN | Door Defeat Enable/Disable Command | <Give a list of relevant messages > | <OZM to Lights> |
| Local On/OFF Request  (Variant 2) | Network | LIN | Local On/Off command | <Give a list of relevant messages > | <OZM to Lights> |
| Courtesy ON/OFF Request  (Variant 2) | Network | LIN | Courtesy Ramp On/Off command. | <Give a list of relevant messages > | <OZM to Lights> |
| Vehicle Speed Request  (Variant 1) | Network | LIN | Communicate vehicle speed in kph. | Veh\_Speed | <BCM to OHC> |
| Vehicle Speed Available  (Variant 1) | Network | LIN | Check for Vehicle speed availability. | Veh\_Speed\_Avail | <BCM to OHC> |
| Door Ajar Request  (Variant 1) | Network | LIN | Provide Door Ajar status for Fr, Fl, Rr, Rl vehicle door. | DF\_Door\_Ajar\_Status PF\_Door\_Ajar\_Status DR\_Door\_Ajar\_Status PF\_Door\_Ajar\_Status | <BCM to OHC> |
| Liftgate Ajar Request  (Variant 1) | Network | LIN | Provide Liftgate Ajar status. | LG\_Ajar\_Status | <BCM to OHC> |
| Alarm Courtesy Request  (Variant 1) | Network | LIN | Trigger lights to Turn ON when perimeter alarm turns On. | Alarm\_Lights\_Courtesy\_Cmd | <BCM to OHC> |
| Crash Courtesy Request  (Variant 1) | Network | LIN | Trigger lights to Turn On post-crash. | Crash\_Courtesy\_Rqst | <BCM to OHC> |
| Silent Mode/Dark Mode  (Variant 1) | Network | LIN | Provide Silent Mode Status. | SilentMode\_Status | <BCM to OHC> |
| BCM Configuration  (Variant 1) | Network | LIN | Provide detail about BCM configuration status. | NotConfigured\_Cmd | <BCM to OHC> |
| Welcome Farewell Mode Request  (Variant 1) | Network | LIN | Provide Welcome Farewell Master Mode Request. | WelcomeFarewell\_State | <BCM to OHC> |
| Welcome Farewell Substate Request  (Variant 1) | Network | LIN | Provide Welcome Farewell Sub Mode Request | WelcomeFarewell\_Substate | <BCM to OHC> |
| Ignition Status  (Variant 1) | Network | LIN | Provide Ignition Status | Ignition\_Status | <BCM to OHC> |
| Day Night Status  (Variant 1) | Network | LIN | Provide Day and Night Status | Day\_Night\_Status | <BCM to OHC> |
| Park Status  (Variant 1) | Network | LIN | Provide status whether vehicle is in Park or not | Park\_Status | <BCM to OHC> |
| Dimming Level  (Variant 1) | Network | LIN | Provide Dimming Level | Dimming\_Lvl | <BCM to OHC> |
| All Lights On/OFF Request (Variant 1) | Network | SBC | Global On/OFF command | <Give a list of relevant messages > | <OHC to Lights> |
| Door Defeat Enable/Disable Request (Variant 1) | Network | SBC | Door Defeat Enable/Disable Command | <Give a list of relevant messages > | <OHC to Lights> |
| Cargo Defeat ON/OFF Request  (Variant 1) | Network | SBC | Cargo Defeat On/Off. | <Give a list of relevant messages > | <OHC to Lights> |
| Courtesy ON/OFF Request  (Variant 1) | Network | SBC | Courtesy Ramp On/Off command. | <Give a list of relevant messages > | <OHC to Lights> |

Table 3‑4: E/E Connections

#### Signal List

***#Hint:*** *Refer to the* [*Data Dictionary*](#_Data_Dictionary) *-* [*Technical Signals*](#_Technical_Signals)*.*

###LSG\_OverheadLamps\_All\_Rq \_00001### LSG\_OverheadLamps\_All\_Rq

CAN Signal indicating button press to toggle All Lights On/Off.

|  |  |  |
| --- | --- | --- |
| **ASIL** | | QM |
| **Encoding Type Name** | | Discrete |
| Note: An encoding is either discrete or continuous. Delete fields below which are not needed | | |
| **Value**  (Discrete  Encoding) | **0x0** | Not Pressed |
| **0x1** | Pressed |
| **Unit** | |  |

###LSG\_DrDefeat\_Rq\_00002### LSG\_DrDefeat\_D\_Rq

CAN Signal requesting Door Defeat/Dome Defeat to enable or disable.

|  |  |  |
| --- | --- | --- |
| **ASIL** | | QM |
| **Encoding Type Name** | | Discrete |
| Note: An encoding is either discrete or continuous. Delete fields below which are not needed | | |
| **Value**  (Discrete  Encoding) | **0x0** | Disable |
| **0x1** | Enable |
| **Unit** | |  |

###LSG\_DrDefeat\_D\_Stat\_00003### LSG\_DrDefeat\_D\_Stat

CAN signal sending feedback to SYNC updating Door Defeat Status.

|  |  |  |
| --- | --- | --- |
| **ASIL** | | QM |
| **Encoding Type Name** | | Discrete |
| Note: An encoding is either discrete or continuous. Delete fields below which are not needed | | |
| **Value**  (Discrete  Encoding) | **0x0** | Disable |
| **0x1** | Enable |
| **Unit** | |  |

###LSG\_MaplampFl\_D\_Rq\_00004### LSG\_MaplampFl\_D\_Rq

CAN signal requesting Local On/OFF for front row left lamp. Other CAN signals like MaplampFr\_D\_Rq, Maplamp2Rl\_D\_Rq , Maplamp2Rr\_D\_Rq , Maplamp3Rl\_D\_Rq , Maplamp3Rr\_D\_Rq request local on/off for second row and third row lamp.

|  |  |  |
| --- | --- | --- |
| **ASIL** | | QM |
| **Encoding Type Name** | | Discrete |
| Note: An encoding is either discrete or continuous. Delete fields below which are not needed | | |
| **Value**  (Discrete  Encoding) | **0x0** | Not Pressed |
| **0x1** | Pressed |
| **Unit** | |  |

###LSG\_Veh\_V\_Actlbrk\_00005### LSG\_Veh\_V\_Actlbrk

CAN Signal communicating vehicle speed in kph.

|  |  |  |
| --- | --- | --- |
| **ASIL** | | QM |
| **Encoding Type Name** | | Continuous |
| Note: An encoding is either discrete or continuous. Delete fields below which are not needed | | |
| **Value**  (Continuous Encoding) | **Min Value** | 0 |
| **Max Value** | 655.34 |
| **Resolution** | 0.01 |
| **Offset** | 0 |
| **Unit** | | KPH |

###LSG\_DrStatDrv\_B\_Actl\_00006### LSG\_DrStatDrv\_B\_Actl

CAN Signal communicating Door Ajar Status for Driver Door. Other CAN signals *DrStatPsngr\_B\_Actl, DrStatRl\_B\_Actl, DrStatRr\_B\_Actl* also communicate the door ajar status for front passenger door, rear left and rear right passenger door respectively.

|  |  |  |
| --- | --- | --- |
| **ASIL** | | QM |
| **Encoding Type Name** | | Discrete |
| Note: An encoding is either discrete or continuous. Delete fields below which are not needed | | |
| **Value**  (Discrete  Encoding) | **0x0** | Closed |
| **0x1** | Ajar |
| **Unit** | |  |

###LSG\_DrStatTgate\_B\_Actl\_00007### LSG\_DrStatTgate\_B\_Actl

CAN signal communicating Liftgate Ajar Status.

|  |  |  |
| --- | --- | --- |
| **ASIL** | | QM |
| **Encoding Type Name** | | Discrete |
| Note: An encoding is either discrete or continuous. Delete fields below which are not needed | | |
| **Value**  (Discrete  Encoding) | **0x0** | Closed |
| **0x1** | Ajar |
| **Unit** | |  |

###LSG\_PerimeterAlarm\_Courtesy\_Rq\_00008### LSG\_PerimeterAlarm\_Courtesy\_Rq

CAN signal requesting overhead light to turn on/off based on Perimeter Alarm status.

|  |  |  |
| --- | --- | --- |
| **ASIL** | | QM |
| **Encoding Type Name** | | Discrete |
| Note: An encoding is either discrete or continuous. Delete fields below which are not needed | | |
| **Value**  (Discrete  Encoding) | **0x0** | Set OFF |
| **0x1** | Set ON |
| **Unit** | |  |

###LSG\_Crash\_Courtesy\_Rq\_00009### LSG\_Crash\_Courtesy\_Rq

CAN Signal requesting overhead lights to turn ON post-crash.

|  |  |  |
| --- | --- | --- |
| **ASIL** | | QM |
| **Encoding Type Name** | | Discrete |
| Note: An encoding is either discrete or continuous. Delete fields below which are not needed | | |
| **Value**  (Discrete  Encoding) | **0x0** | Null |
| **0x1** | Set ON |
| **Unit** | |  |

###LSG\_SilentMode\_Rq\_00010### LSG\_SilentMode\_Rq

CAN Signal requesting overhead lights to go in silent mode or dark police car mode.

|  |  |  |
| --- | --- | --- |
| **ASIL** | | QM |
| **Encoding Type Name** | | Discrete |
| Note: An encoding is either discrete or continuous. Delete fields below which are not needed | | |
| **Value**  (Discrete  Encoding) | **0x0** | Null |
| **0x1** | ON |
| **Unit** | |  |

###LSG\_BcmNotConfigured\_Cntrl\_00011### LSG\_BcmNotConfigured\_Cntrl

CAN Signal requesting overhead lights to turn on if BCM is not configured.

|  |  |  |
| --- | --- | --- |
| **ASIL** | | QM |
| **Encoding Type Name** | | Discrete |
| Note: An encoding is either discrete or continuous. Delete fields below which are not needed | | |
| **Value**  (Discrete  Encoding) | **0x0** | Set OFF |
| **0x1** | Set On |
| **Unit** | |  |

###LSG\_VehWlcmFrwl\_D\_Stat\_00012### LSG\_VehWlcmFrwl\_D\_Stat

CAN Signal communicating Master Mode of Welcome/Farewell command.

|  |  |  |
| --- | --- | --- |
| **ASIL** | | QM |
| **Encoding Type Name** | | Discrete |
| Note: An encoding is either discrete or continuous. Delete fields below which are not needed | | |
| **Value**  (Discrete  Encoding) | **0x0** | Null |
| **0x1** | Welcome |
| **0x2** | Farewell |
| **0x3** | Run start |
| **Unit** | |  |

###LSG\_VehWlcmFrwlMde\_D\_Stat\_00013### LSG\_VehWlcmFrwlMde\_D\_Stat

CAN Signal communicating sub modes of welcome/farewell.

|  |  |  |
| --- | --- | --- |
| **ASIL** | | QM |
| **Encoding Type Name** | | Discrete |
| Note: An encoding is either discrete or continuous. Delete fields below which are not needed | | |
| **Value**  (Discrete  Encoding) | **0x0** | Null |
| **0x1** | Approach |
| **0x2** | Illuminated Entry |
| **0x3** | CourtesyLightingAll |
| **0x4** | CourtesyLightingDelayAll |
| **0x5** | CourtesyLightingExtended |
| **0x6** | CourtesyLightingDelayExt |
| **0x7** | Illuminated Exit |
| **Unit** | |  |

###LSG\_Ignition\_Status\_00014### LSG\_Ignition\_Status

CAN Signal communicating the vehicle ignition status.

|  |  |  |
| --- | --- | --- |
| **ASIL** | | QM |
| **Encoding Type Name** | | Discrete |
| Note: An encoding is either discrete or continuous. Delete fields below which are not needed | | |
| **Value**  (Discrete  Encoding) | **0x0** | Unknown |
| **0x1** | Off |
| **0x2** | Accessory |
| **0x4** | Run |
| **0x8** | Start |
| **0xF** | Invalid |
| **Unit** | |  |
|  | |  |
|  | |  |

#### Software Component Architecture

**#Classification:** Optional – For features with in-house SW development only (remove section otherwise)

***#Hint:*** *For Features with in-house SW development (specifically in an Agile Environment) it is required, that the development team documents and agrees on at least their SW interfaces to the outside world early in the process.*

#### Description

**#Hint**: Provide some informal description of the characteristics of the chosen Software Component Architecture. Also give some graphical representation of the Software Component Architecture. SysML Internal Block Diagrams or [AUTOSAR](http://wiki.ford.com/display/RequirementsEngineering/Data+Flow+Diagram?src=contextnavpagetreemode) Virtual Function Bus models could be used to depict such a Software Component Architecture.

**#Link:** [*SysML – Internal Block Diagrams*](https://pd3.spt.ford.com/sites/SystemsEngineering/SEC/sysml-teamsite/SysML%20Wiki/Internal%20Block%20Diagram%20Basics.aspx) and [*AUTOSAR*](https://www.autosar.org/)

This Software Component Architecture … <add some explanatory text here>

Figure 3‑6: AUTOSAR compliant SW Component Architecture

## Function Deployment

***#Hint:*** *This section lists and details the deployment variants of the feature.*

### Deployment Variants

**#Classification:** Mandatory – State “No Variants defined”, if not used.

**#Hint:** If there is more than 1 variant of deployment, the different variant should be listed and described below. Deployment variants are very much driven by E/E architecture variants (refer to section *E/E Architecture Variants*). Nevertheless, Feature/Function variant options might also drive additional deployment variants.

**#For Functional Safety:** Specify each deployment variant in a separate FIS.

|  |  |  |
| --- | --- | --- |
| **Deployment Variant Name** | Variant Description | Variant Condition (optional) |
| Variant 1: Physical Switch | Activation and Deactivation of Overhead Lights using All Lights Button (Global On/Off), Door Defeat using Physical switch. | With Overhead Console. |
| Variant 1: Reductive Switch Design | Activation and Deactivation of Overhead Lights using All Lights Button (Global On/Off), Door Defeat using soft switch on sync.  Dual activation/deactivation through physical switch as well as through soft touch on sync. | No Overhead Console |

#### Deployment “Variant 1”

***#Classification:*** *Optional*

***#Hint:*** *Add a deployment diagram (e.g. a SysML Activity Diagram where the actions represent the Technology Functions and the swimlanes represent the components) and some explanatory text about the variant to this section. The naming of the Technology Functions should make clear, what Logical Function it had been derived from (e.g. VehicleSpeedCalculation(Wheelbased)\_ABS)*

This deployment variant … <add some explanatory text here>

Table 3‑5: Sample Deployment Diagram

### Function Allocation

***#Hint:*** *The “Function Allocation Table” shows the mapping of the Logical Functions and the corresponding Technology Functions of a feature to components of the physical architecture as also shown in the deployment diagrams. Typically, there is a 1:1 relationship between (Atomic) Logical and Technology Functions. For details refer to the* *RE Wiki pages* [*“Deriving Implemented Functions from Logical Functions”*](http://wiki.ford.com/display/RequirementsEngineering/Deriving+Implemented+Functions+from+Logical+Functions) *and “*[*Cascade Requirements*](http://wiki.ford.com/display/RequirementsEngineering/Cascade+Requirements%23CascadeRequirements-CascadingVsTraceability)*” When applying MBSE methods please refer to Guideline for Alignment of SW QoS with Ford Starting Model (SysML) for how Logical and Technology Functions in the Ford Starting Model align to Atomic Logical Functions and Technology Functions in RE.*

*For Functional Safety critical features the second table (“****Error! Reference source not found.****”) has to be additionally filled in*

* *to map Technical Safety Requirements (TSRs) to Technology Functions and hence Components and*
* *to assign an ASIL level to Components and TSRs*

***#Link:***[*RE Wiki - Deriving Implemented Functions from Logical Functions*](http://wiki.ford.com/display/RequirementsEngineering/Deriving+Implemented+Functions+from+Logical+Functions)

[*RE Wiki - Cascade Requirements*](http://wiki.ford.com/display/RequirementsEngineering/Cascade+Requirements#CascadeRequirements-CascadingVsTraceability)

[*Functional Safety Sharepoint*](https://pd3.spt.ford.com/sites/GlobalFunctionalSafety/Pages/default.aspx)

[*Guideline for Alignment of SW QoS with Ford Starting Model (SysML)*](http://wiki.ford.com/display/RequirementsEngineering/Alignment+with+the+Ford+Starting+Model)

| Component | Technology Function Name | Logical Function Name |
| --- | --- | --- |
|
| PDC | ECU | Control Local On/OFF |
| Control Global On/OFF |
| Control Door Defeat |
| BCM | ECU | Arbitration |
| Provide Perimeter Alarm Courtesy Request |
| Provide Crash Courtesy Request |
| Provide Silent Mode Status |
| Provide Vehicle Welcome Farewell Master State |
| Provide Vehicle Welcome Farewell Sub State |
| Provide Door/Liftgate Ajar Status |
| Provide Ignition Status |
| Provide Vehicle Speed Availability (Variant 1) |
| Provide Dimming Level (Variant 1) |
| Provide Day Night Status (Variant 1) |
| Provide Vehicle Speed (Variant 1) |
| ABS | ECU | Provide Vehicle Speed (Variant 2) |
| OZM | ECU | Provide Global On/OFF (Variant 2) |
| Provide Local On/OFF (Variant 2) |
| Provide Door Defeat On/OFF (Variant 2) |
| Provide Courtesy Ramp On/OFF (Variant 2) |
| Provide Cargo Defeat On/OFF (Variant 2) |
| OHC | ECU | Provide Global On/OFF (Variant 1) |
| Provide Cargo Defeat On/OFF (Variant 1) |
| Provide Door Defeat On/OFF (Variant 1) |
| Provide Courtesy Ramp On/OFF (Variant 1) |
| Map/Dome Switch | ECU | Provide Local On/OFF |
| Overhead Lights | Activate/Deactivate | Provide Illumination |

Table 3‑6: Function Allocation Table (Basic)

# Feature Implementation Modeling

***#Hint:*** *This chapter shall give deployment specific refinements of the customer use cases, which are defined in the Feature Document. Based on activity diagrams and possibly sequence diagrams, this chapter shows, how the overall functionality is decomposed in activities / functions, which can be allocated to single physical components. In addition, this chapter shows how the individual components / activities collaborate.*

## Component Interaction Diagrams

***#Hint:*** *This chapter would typically list sequence charts or activity diagrams with swim lanes for all involved physical components to illustrate the interactions between components for relevant scenarios.*

*End-to-end timing constraints could be placed in the diagrams. End-to-end timing would be relevant if more than 2 nodes (sender / receiver) are involved when information flows from signal generation (e.g. button pressed by user) to signal consumption (e.g. light is switched on).*

### Scenario: “System Startup / Shutdown”

***#Hint:*** *This chapter shall define specific scenarios / sequences (e.g. power-up initialization, ignition after-run, NM sleep / wake up, MicroHybrid start / stop inhibition.*

### Scenario: “Normal Operation”

**Figure 12:** Sample Scenario “Normal Operation”

## Component Interface Behavior Diagrams

*#Hint: For complex (application level) interface protocols a protocol state machine would be more appropriate than a bunch of sequence diagrams to illustrate the interactions between components. So, this section would typically show a (protocol) state machine.*

# Feature Implementation Requirements

***#Hint:*** *The Feature Implementation Specification is first of all an architecture document. It shows the Functional and the E/E architecture as well as the deployment of the Functional one to the E/E one.*

## Functional Safety

**#Classification**: Functional Safety only – If not used, remove content and state “Not Applicable”

***#Hint:*** *If feature is not Functional Safety critical, remove subsections from this chapter and state “Feature is not Functional Safety critical”*

### ASIL Decomposition of Technical Safety Requirements

**#Classification**: Functional Safety only – If not used, remove content and state “Not Applicable”

***#Hint:*** *Sometimes an ASIL decomposition of Technical Safety Requirements (TSRs) is required. The (input) TSRs, which are to be decomposed, are derived from FSRs. Those input TSRs are to be specified in this chapter (right above the corresponding ASIL decomposition table). For each input TSR add one “ASIL Decomposition Table”. In the “ASIL Decomposition Table” the derived, decomposed TSRs are referenced by ID and Title. Those TSRs are however not specified in the FIS but in the ECU Functional Specfication.*

**#Macro:** [Add Ins -> Add Requirement macro](http://wiki.ford.com/pages/viewpage.action?pageId=174654231) (select “**Func./Tech. Safety Requirement**” as type)

***#Link:***[*Functional Safety Sharepoint*](https://pd3.spt.ford.com/sites/GlobalFunctionalSafety/Pages/default.aspx) *- Functional Safety Concept*

<Place the input TSR here above the decomposition table>

| **Input TSR** | <Provide the ID of the TSR which shall be decomposed. That TSR is given above> | |
| --- | --- | --- |
| **Decomposition Rationale** | <Give a reason why the decomposition was performed> | |
| **Method for Decomposition** | Choose a Method | |
| **TSR 1 after Decomposition** | **TSR ID** | <Provide the ID of the decomposed TSR> |
| **TSR Title** | <Provide the title of the decomposed TSR> |
| **ASIL** |  |
| **Rationale** | <Provide a reason and thought behind that particular requirement. Should include how the requirement is able to independently fulfill the needs of the parent requirement> |
| **Satisfied by** | <Provide an Technology Function, physical signal, or physical component satisfying the requirement. This element shall be independent of the element satisfied by the other half of the ASIL decomposition.> |
| **TSR 2 after Decomposition** | **TSR ID** | <Provide the ID of the decomposed TSR> |
| **TSR Title** | <Provide the title of the decomposed TSR> |
| **ASIL** |  |
| **Rationale** | <Provide a reason and thought behind that particular requirement. Should include how the requirement is able to independently fulfill the needs of the parent requirement> |
| **Satisfied by** | <Provide an Technology Function, physical signal, or physical component satisfying the requirement. This element shall be independent of the element satisfied by the other half of the ASIL decomposition.> |
| **TSR for Independence**  *Note: should consider commonly used input, output and processing*  *Note: additional row should be added if additional* *requirements for Independence are necessary* | **TSR ID** |  |
| **TSR Title** |  |
| **ASIL** |  |
| **Rationale** |  |

Table 5‑1: ASIL Decomposition Table

## Requirements on Components

### Body Control Module

#### Technology Function “MyLogicalFunctionA\_Component1”

***#Classification:*** *Mandatory*

***#Hint:*** *Technology Functions are split into two parts:*

* *subsection Function Interfaces: defines the mapping of the Logical Signals/Parameters to Technical (i.e. physical) Signals/Parameters, which is ECU/allocation specific. It also specifies interface details on how signals are published or subscribed*
* *subsection Function Requirements: defines, which requirements are reused/carried over unchanged from the Logical Function and which requirements are need to be modified due to deployment specific circumstances.*

**#Link:** [*RE Wiki – Deriving an Implemented Function*](http://wiki.ford.com/display/RequirementsEngineering/Deriving+Implemented+Functions+from+Logical+Functions)*#*

[*RE Wiki - Cascade Requirements*](http://wiki.ford.com/display/RequirementsEngineering/Cascade+Requirements#CascadeRequirements-CascadingVsTraceability)

*RE-Wiki – How to manage requirements in VSEM – Implemented Functions*

##### Function Interfaces

***#Hint:***

*The subsections “Inputs”, “Outputs” and “Parameters” below map the Logical Signals / Parameters their Technical (i.e. physical) counterparts, which are sent and received by the Technology Function. This is done by mapping tables, which reference Data Dictionary entries. If Data Dictionary entries are not sufficient, the Interface Requirements section may be used to specify additional requirements. If possible also link those requirements to the tables in the subsections “Inputs”, “Outputs” and “Parameters”.*

*How to …:*

1. *If the technical signal / parameter does not yet exist in the GSDB or elsewhere in VSEM, create those in the “Technical Signals”/”Technical Parameters” section of the “Data Dictionary”. Use* [*Add Ins -> Add Requirement macro*](http://wiki.ford.com/display/RequirementsEngineering/How+to+use+the+Specification+Templates#AddNewRequirement) *(select “Technical Signal”/”Technical Parameter” as type).*

*Note: Bookmarks are automatically created for the ID, the name and the description of each signal / parameter in the “Data Dictionary”, if you use the macros.*

*Note: The corresponding Logical Signal/Parameter should already exist in the Data Dictionary, because it should have been created, when specifying the Logical Function (refer to the Logical Signals/Technical Signals).*

1. *Reference the signal / parameter name bookmark from the Data Dictionary in the tables below.*
2. *If the mapping is not 1:1 (e.g. a Logical Signal gets split into 2 Technical Signals) the Mapping Details need to be specified by a Mappings object in the Data Dictionary.*
3. *For “Publisher Interfaces” and “Subscriber Interfaces” column you may (optionally) link to AIS Interfaces in the Data Dictionary.*
4. *For “Messages” to be referenced in the “E/E Connections” column proceed similarly. Example: <ConnectionName>::<MessageName> refers to the Message which is sent on bus <Connection Name> and which is given in section Messages of the “Data Dictionary”. Alternatively, for CAN you could link directly to the message from the CMDB (e.g.* [*CGEA 1.3*](https://www.vsemweb.ford.com/tc/launchapp?-attach=true&-s=226TCSession&-o=n0SJN9h0x3NrTDAAAAAAAAAAAAA&servername=Production_Server) *or* [*FNV2*](https://www.vsemweb.ford.com/tc/launchapp?-attach=true&-s=226TCSession&-o=jXfpx2PHx3NrTDAAAAAAAAAAAAA&servername=Production_Server)*).*

**#Link:** [*RE Wiki – Adding a Technical Interface*](http://wiki.ford.com/display/RequirementsEngineering/Adding+a+Technical+Interface)

**#Link:** [*RE Wiki – Adding a Signal or Parameter Mapping*](http://wiki.ford.com/display/RequirementsEngineering/Adding+a+Signal+or+Parameter+Mapping)

###### Inputs

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Logical Signal Name** | **Technical Signal Name** | **Mapping Details** *(Conditional)* | **Subscriber Interface** | **Connection**  (*Optional)* |
| Provide Lock Unlock Status |  |  |  |  |
| Provide Ignition Status | Ignition\_Status |  |  |  |
| Provide Door Ajar Status | DrStatDrv\_B\_Actl DrStatPsngr\_B\_Actl DrStatRl\_B\_Actl DrStatRr\_B\_Actl  DrStatTgate\_B\_Actl |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

Table 5‑2: Input Signal mappings of Function “MyLogicalFunctionA\_Component1”

###### Outputs

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Logical Signal Name** | **Technical Signal Name** | **Mapping Details**  *(Conditional)* | **Publisher Interface** | **Connection**  *(Optional)* |
| Provide Perimeter Alarm Status | PerimeterAlarm\_Courtesy\_Rq |  | BCM - OZM |  |
| Provide Crash Courtesy Status | Crash\_Courtesy\_Rq |  | BCM - OZM |  |
| Provide Silent Mode Status | SilentMode\_Rq |  | BCM - OZM |  |
| Provide Ignition Status | Ignition\_Status |  | BCM - OZM |  |
| Provide Door Ajar Status | DrStatDrv\_B\_Actl DrStatPsngr\_B\_Actl DrStatRl\_B\_Actl DrStatRr\_B\_Actl  DrStatTgate\_B\_Actl |  | BCM - OZM |  |
| Provide Vehicle Welcome Farewell Mode | VehWlcmFrwl\_D\_Stat |  | BCM - OZM |  |
| Provide Vehicle Welcome Farewell Sub Mode. | VehWlcmFrwlMde\_D\_Stat |  | BCM - OZM |  |
|  |  |  |  |  |

Table 5‑3: Output Signal mappings of Function “MyLogicalFunctionA\_Component1”A

###### Parameters

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Logical Parameter Name** | **Technical Parameter Name** | **Mapping Details** *(Conditional)* | **Method** | **Method Details** |
| CourtesyLight\_Timeout | CourtesyLight\_Timeout |  |  |  |
| Batterysaver\_Timeout | Batterysaver\_Timeout |  |  |  |
| LE\_Courtesy\_RampUp\_Duration\_Cfg | LE\_Courtesy\_RampUp\_Duration\_Cfg |  |  |  |
| LE\_Courtesy\_RampDn\_Duration\_Cfg | LE\_Courtesy\_RampDn\_Duration\_Cfg |  |  |  |
| NonLE\_Courtesy\_RampUp\_Duration\_Cfg | NonLE\_Courtesy\_RampUp\_Duration\_Cfg |  |  |  |
| BSaveTimeOut\_CourtesyDemand\_Cfg | BSaveTimeOut\_CourtesyDemand\_Cfg |  |  |  |

Table 5‑4: Parameter mappings of Function “MyLogicalFunctionA\_Component1”

###### Interface Requirements

***#Hint:*** *This section provides a place where to specify interface specific requirements of the Technology Function, if Interface objects from* *Technical Interfaces* *and Mappings objects cannot be used (e.g. requirement is not covered by AIS attribute). Deployment specific requirements, which are not related to the interface directly, should be specified in section Function Requirements.*

*Naming Convention for interface requirements:*

*“PubIfReq\_TechnicalSignalName: InterfaceAttribute” (e.g. “PubIfReq\_Veh\_V\_ActlBrk: Timing”)*

*“SubIfReq\_TechnicalSignalName: InterfaceAttribute” (e.g. “SubIfReq\_Veh\_V\_ActlBrk: Missing/Invalid Signal”)*

*“MapReq\_LogicalSignalName\_TechnicalSignalName” (e.g. “MapReq\_LSG\_VehicleSpeed\_Veh\_V\_ActlBrk”)*

*For a selection of interface attributes refer to the list below*

*List of possible interface attribute groups/attributes of a signal subscriber:*

|  |  |
| --- | --- |
| *Timing* | *Signal refresh rate, Publishing Interval (ms), Publisher Latency Requirements, Signal Transmit Cycle Time, End-to-End Latency Requirements* |
| *Wakeup / Sleep* | *Publishing Network Sleep Inhibitor, Updates Signal while asleep, Network Wake Up, fresh data on Network wakeup, Max latency before signal is valid on Network wakeup* |
| *Reset* | *Fresh data on ECU Reset, Max latency before signal is valid on reset* |
| *Robustness/Integrity* | *Checksum, Counter, Quality Factor, MAC, Functional Safety Relevant,* |
| *Functional* | *ECU Power Mode, Functional Voltage Range (Min, Max), Performance Voltage Range (Min, Max), CAN Node Type, Standardization Category, Fault Type,* |

*List of possible interface attributes/attribute groups of a signal publisher:*

|  |  |
| --- | --- |
| *Timing* | *Subscribing Interval (ms), Subscriber Latency Requirements, End-to-End Latency Requirements* |
| *Missing/Invalid Signals Strategy* | *Missing Message Strategy, Use Last Signal Value when Missing Message, Timeout period when Last Signal Value cannot be used for missing message, Use Default Value when Missing Message, Missing Message Default Value, Missing Message DTC, Update Bit, Update Bit Signal Logic* |
| *Robustness/Integrity* | *Checksum, Counter, MAC, Quality Factor, Functional Safety Relevant, ASIL Rating* |
| *Wakeup / Sleep* | *Network Wake Up, Subscribing Network Sleep Inhibitor* |
| *Routing* | *Gateway Required, Gateway Message Type, Max Gateway Latency* |
| *Functional* | *ECU Power Mode, Functional Voltage Range (Min, Max), Performance Voltage Range (Min, Max), CAN Node Type* |

##### Function Requirements

***#Hint:*** *The table “Component Specific Requirements” below lists those requirements of the Logical Function, which are removed/modified/added in context of the specific component, which the Technology Function is allocated to. If “Modification” is set to “Replaced” or “Added” specify the new requirement in subsection “Component Specific Requirements”. Ideally, the table should remain empty (100% reuse/carry over of the Logical Function requirements). That is, modifications of the requirement set in context of the FIS should be kept to a minimum.*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Requirement ID**  (of Logical Function) | **Requirement Title** | **Modification** | **Requirement ID**  (of Technology Function) | **Comment** |
| REQ\_abc |  | Removed | -- |  |
| REQ\_def |  | Replaced | REQ\_xyz |  |
| -- |  | Added | REQ\_123 |  |

Table 5‑5: Component Specific Requirements

***#Hint:*** *Optionally, the table “Inherited Requirements” below defines which requirements of the corresponding Logical Function are reused without change by the ECU. This table is optional, because the set of unchanged requirements can be derived implicitly from the list of requirements of the Logical Function and those listed in the table “Component Specific Requirements”.*

|  |  |  |
| --- | --- | --- |
| **Requirement ID**  (of Logical Function) | **Requirement Title** | **Comment** |
|  |  |  |
|  |  |  |
| … |  |  |

Table 5‑6: Inherited Requirements

###### Component Specific Requirements

***#Hint:*** *If in table “Component Specific Requirements” requirements of the Logical Function are marked as modified/added place the modified/added requirements in this section.*

#### (Technology) Function “MyLogicalFunctionB\_Component1”

### Phoenix Domain Controller (PDC)

###### Inputs

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Logical Signal Name** | **Technical Signal Name** | **Mapping Details** *(Conditional)* | **Subscriber Interface** | **Connection**  (*Optional)* |
| Control Global |  |  | Sync Display - PDC | LVDS |
| Control Local |  |  | Sync Display - PDC | LVDS |
| Control Door Defeat |  |  | Sync Display - PDC | LVDS |
| Feedback | DrDefeat\_D\_Stat |  | OZM – PDC | CAN |

Table 5‑7: Input Signal mappings of Function “MyLogicalFunctionA\_Component1”

###### Outputs

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Logical Signal Name** | **Technical Signal Name** | **Mapping Details**  *(Conditional)* | **Publisher Interface** | **Connection**  *(Optional)* |
| Control Global | OverheadLamps\_All\_Rq |  | PDC – ECG - OZM | CAN |
| Control Local | MaplampFl\_D\_Rq MaplampFr\_D\_RqMaplamp2Rl\_D\_RqMaplamp2Rr\_D\_Rq Maplamp3Rl\_D\_Rq Maplamp3Rr\_D\_Rq |  | PDC – ECG - OZM | CAN |
| Control Door Defeat | DrDefeat\_D\_Rq |  | PDC – ECG - OZM | CAN |

Table 5‑8: Output Signal mappings of Function “MyLogicalFunctionA\_Component1”A

###### Parameters

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Logical Parameter Name** | **Technical Parameter Name** | **Mapping Details** *(Conditional)* | **Method** | **Method Details** |
|  |  |  |  |  |
|  |  |  |  |  |

Table 5‑9: Parameter mappings of Function “MyLogicalFunctionA\_Component1”

###### Interface Requirements

***#Hint:*** *This section provides a place where to specify interface specific requirements of the Technology Function, if Interface objects from* *Technical Interfaces* *and Mappings objects cannot be used (e.g. requirement is not covered by AIS attribute). Deployment specific requirements, which are not related to the interface directly, should be specified in section Function Requirements.*

*Naming Convention for interface requirements:*

*“PubIfReq\_TechnicalSignalName: InterfaceAttribute” (e.g. “PubIfReq\_Veh\_V\_ActlBrk: Timing”)*

*“SubIfReq\_TechnicalSignalName: InterfaceAttribute” (e.g. “SubIfReq\_Veh\_V\_ActlBrk: Missing/Invalid Signal”)*

*“MapReq\_LogicalSignalName\_TechnicalSignalName” (e.g. “MapReq\_LSG\_VehicleSpeed\_Veh\_V\_ActlBrk”)*

*For a selection of interface attributes refer to the list below*

*List of possible interface attribute groups/attributes of a signal subscriber:*

|  |  |
| --- | --- |
| *Timing* | *Signal refresh rate, Publishing Interval (ms), Publisher Latency Requirements, Signal Transmit Cycle Time, End-to-End Latency Requirements* |
| *Wakeup / Sleep* | *Publishing Network Sleep Inhibitor, Updates Signal while asleep, Network Wake Up, fresh data on Network wakeup, Max latency before signal is valid on Network wakeup* |
| *Reset* | *Fresh data on ECU Reset, Max latency before signal is valid on reset* |
| *Robustness/Integrity* | *Checksum, Counter, Quality Factor, MAC, Functional Safety Relevant,* |
| *Functional* | *ECU Power Mode, Functional Voltage Range (Min, Max), Performance Voltage Range (Min, Max), CAN Node Type, Standardization Category, Fault Type,* |

*List of possible interface attributes/attribute groups of a signal publisher:*

|  |  |
| --- | --- |
| *Timing* | *Subscribing Interval (ms), Subscriber Latency Requirements, End-to-End Latency Requirements* |
| *Missing/Invalid Signals Strategy* | *Missing Message Strategy, Use Last Signal Value when Missing Message, Timeout period when Last Signal Value cannot be used for missing message, Use Default Value when Missing Message, Missing Message Default Value, Missing Message DTC, Update Bit, Update Bit Signal Logic* |
| *Robustness/Integrity* | *Checksum, Counter, MAC, Quality Factor, Functional Safety Relevant, ASIL Rating* |
| *Wakeup / Sleep* | *Network Wake Up, Subscribing Network Sleep Inhibitor* |
| *Routing* | *Gateway Required, Gateway Message Type, Max Gateway Latency* |
| *Functional* | *ECU Power Mode, Functional Voltage Range (Min, Max), Performance Voltage Range (Min, Max), CAN Node Type* |

##### Function Requirements

***#Hint:*** *The table “Component Specific Requirements” below lists those requirements of the Logical Function, which are removed/modified/added in context of the specific component, which the Technology Function is allocated to. If “Modification” is set to “Replaced” or “Added” specify the new requirement in subsection “Component Specific Requirements”. Ideally, the table should remain empty (100% reuse/carry over of the Logical Function requirements). That is, modifications of the requirement set in context of the FIS should be kept to a minimum.*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Requirement ID**  (of Logical Function) | **Requirement Title** | **Modification** | **Requirement ID**  (of Technology Function) | **Comment** |
| REQ\_abc |  | Removed | -- |  |
| REQ\_def |  | Replaced | REQ\_xyz |  |
| -- |  | Added | REQ\_123 |  |

Table 5‑10: Component Specific Requirements

***#Hint:*** *Optionally, the table “Inherited Requirements” below defines which requirements of the corresponding Logical Function are reused without change by the ECU. This table is optional, because the set of unchanged requirements can be derived implicitly from the list of requirements of the Logical Function and those listed in the table “Component Specific Requirements”.*

|  |  |  |
| --- | --- | --- |
| **Requirement ID**  (of Logical Function) | **Requirement Title** | **Comment** |
|  |  |  |
|  |  |  |
| … |  |  |

Table 5‑11: Inherited Requirements

###### Component Specific Requirements

***#Hint:*** *If in table “Component Specific Requirements” requirements of the Logical Function are marked as modified/added place the modified/added requirements in this section.*

#### (Technology) Function “MyLogicalFunctionB\_Component1”

### Overhead Zone Module

###### Inputs

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Logical Signal Name** | **Technical Signal Name** | **Mapping Details** *(Conditional)* | **Subscriber Interface** | **Connection**  (*Optional)* |
| Provide Perimeter Alarm Status | PerimeterAlarm\_Courtesy\_Rq |  | BCM - OZM | CAN |
| Provide Crash Courtesy Status | Crash\_Courtesy\_Rq |  | BCM - OZM | CAN |
| Provide Silent Mode Status | SilentMode\_Rq |  | BCM - OZM | CAN |
| Provide Ignition Status | Ignition\_Status |  | BCM - OZM | CAN |
| Provide Door Ajar Status | DrStatDrv\_B\_Actl DrStatPsngr\_B\_Actl DrStatRl\_B\_Actl DrStatRr\_B\_Actl  DrStatTgate\_B\_Actl |  | BCM - OZM | CAN |
| Provide Vehicle Welcome Farewell Mode | VehWlcmFrwl\_D\_Stat |  | BCM - OZM | CAN |
| Provide Vehicle Welcome Farewell Sub Mode. | VehWlcmFrwlMde\_D\_Stat |  | BCM - OZM | CAN |
| Control Global | OverheadLamps\_All\_Rq |  | PDC – ECG - OZM | CAN |
| Control Local | MaplampFl\_D\_Rq MaplampFr\_D\_Rq Maplamp2Rl\_D\_Rq Maplamp2Rr\_D\_Rq Maplamp3Rl\_D\_Rq Maplamp3Rr\_D\_Rq |  | PDC – ECG - OZM | CAN |
| Control Door Defeat | DrDefeat\_D\_Rq |  | PDC – ECG - OZM | CAN |

Table 5‑12: Input Signal mappings of Function “MyLogicalFunctionA\_Component1”

###### Outputs

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Logical Signal Name** | **Technical Signal Name** | **Mapping Details**  *(Conditional)* | **Publisher Interface** | **Connection**  *(Optional)* |
| All Lights On OFF | All Lights On OFF |  | OZM – Overhead Lights | LIN |
| Local On OFF | Local On OFF |  | OZM – Overhead Lights | LIN |
| Door Defeat On OFF | Door Defeat On OFF |  | OZM – Overhead Lights | LIN |
| Courtesy On OFF | Courtesy On OFF |  | OZM – Overhead Lights | LIN |
|  |  |  |  |  |

Table 5‑13: Output Signal mappings of Function “MyLogicalFunctionA\_Component1”A

###### Parameters

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Logical Parameter Name** | **Technical Parameter Name** | **Mapping Details** *(Conditional)* | **Method** | **Method Details** |
| Vehiclespeed\_Threshold\_Cfg | Vehiclespeed\_Threshold\_Cfg |  |  |  |
|  |  |  |  |  |

Table 5‑14: Parameter mappings of Function “MyLogicalFunctionA\_Component1”

###### Interface Requirements

***#Hint:*** *This section provides a place where to specify interface specific requirements of the Technology Function, if Interface objects from* *Technical Interfaces* *and Mappings objects cannot be used (e.g. requirement is not covered by AIS attribute). Deployment specific requirements, which are not related to the interface directly, should be specified in section Function Requirements.*

*Naming Convention for interface requirements:*

*“PubIfReq\_TechnicalSignalName: InterfaceAttribute” (e.g. “PubIfReq\_Veh\_V\_ActlBrk: Timing”)*

*“SubIfReq\_TechnicalSignalName: InterfaceAttribute” (e.g. “SubIfReq\_Veh\_V\_ActlBrk: Missing/Invalid Signal”)*

*“MapReq\_LogicalSignalName\_TechnicalSignalName” (e.g. “MapReq\_LSG\_VehicleSpeed\_Veh\_V\_ActlBrk”)*

*For a selection of interface attributes refer to the list below*

*List of possible interface attribute groups/attributes of a signal subscriber:*

|  |  |
| --- | --- |
| *Timing* | *Signal refresh rate, Publishing Interval (ms), Publisher Latency Requirements, Signal Transmit Cycle Time, End-to-End Latency Requirements* |
| *Wakeup / Sleep* | *Publishing Network Sleep Inhibitor, Updates Signal while asleep, Network Wake Up, fresh data on Network wakeup, Max latency before signal is valid on Network wakeup* |
| *Reset* | *Fresh data on ECU Reset, Max latency before signal is valid on reset* |
| *Robustness/Integrity* | *Checksum, Counter, Quality Factor, MAC, Functional Safety Relevant,* |
| *Functional* | *ECU Power Mode, Functional Voltage Range (Min, Max), Performance Voltage Range (Min, Max), CAN Node Type, Standardization Category, Fault Type,* |

*List of possible interface attributes/attribute groups of a signal publisher:*

|  |  |
| --- | --- |
| *Timing* | *Subscribing Interval (ms), Subscriber Latency Requirements, End-to-End Latency Requirements* |
| *Missing/Invalid Signals Strategy* | *Missing Message Strategy, Use Last Signal Value when Missing Message, Timeout period when Last Signal Value cannot be used for missing message, Use Default Value when Missing Message, Missing Message Default Value, Missing Message DTC, Update Bit, Update Bit Signal Logic* |
| *Robustness/Integrity* | *Checksum, Counter, MAC, Quality Factor, Functional Safety Relevant, ASIL Rating* |
| *Wakeup / Sleep* | *Network Wake Up, Subscribing Network Sleep Inhibitor* |
| *Routing* | *Gateway Required, Gateway Message Type, Max Gateway Latency* |
| *Functional* | *ECU Power Mode, Functional Voltage Range (Min, Max), Performance Voltage Range (Min, Max), CAN Node Type* |

##### Function Requirements

***#Hint:*** *The table “Component Specific Requirements” below lists those requirements of the Logical Function, which are removed/modified/added in context of the specific component, which the Technology Function is allocated to. If “Modification” is set to “Replaced” or “Added” specify the new requirement in subsection “Component Specific Requirements”. Ideally, the table should remain empty (100% reuse/carry over of the Logical Function requirements). That is, modifications of the requirement set in context of the FIS should be kept to a minimum.*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Requirement ID**  (of Logical Function) | **Requirement Title** | **Modification** | **Requirement ID**  (of Technology Function) | **Comment** |
| REQ\_abc |  | Removed | -- |  |
| REQ\_def |  | Replaced | REQ\_xyz |  |
| -- |  | Added | REQ\_123 |  |

Table 5‑15: Component Specific Requirements

***#Hint:*** *Optionally, the table “Inherited Requirements” below defines which requirements of the corresponding Logical Function are reused without change by the ECU. This table is optional, because the set of unchanged requirements can be derived implicitly from the list of requirements of the Logical Function and those listed in the table “Component Specific Requirements”.*

|  |  |  |
| --- | --- | --- |
| **Requirement ID**  (of Logical Function) | **Requirement Title** | **Comment** |
|  |  |  |
|  |  |  |
| … |  |  |

Table 5‑16: Inherited Requirements

###### Component Specific Requirements

***#Hint:*** *If in table “Component Specific Requirements” requirements of the Logical Function are marked as modified/added place the modified/added requirements in this section.*

#### (Technology) Function “MyLogicalFunctionB\_Component1”

## Requirements on Connections

### Networks

#### “CAN Bus xxx”

***#Hint:*** *For CAN most requirements are defined by Netcom and referenced in* [*VSEM “Multiplexing Specifications” section*](https://www.vsemweb.ford.com/tc/launchapp?-attach=true&-s=226TCSession&-o=xcbJ6OwAx3NrTDAAAAAAAAAAAAA&servername=Production_Server)*. Put in this section only those requirements, which deviate from that standard specification.*

*The CAN messages relevant for this feature are listed in the section “Messages” of the “Data Dictionary”.*

##### Protocol Requirements

**#Hint:** *For CAN Ford currently mandates FNOS as SW implementation for the CAN protocol stack. This includes the CAN Network Management and Transport Protocol. If you deviate from this assumption or if you have specific requirements on FNOS, which are not contained in the standard package, put requirements in this section. F*or details the FNOS user guide and application notes could be referenced.

##### Electrical Requirements

**#Hint:** List requirements here, only if they deviate from the SDS CAN.

#### “LIN Bus xxx”

**#Hint:** Place requirements here, which are common to all LIN nodes, but not covered by some SDS LIN.

*The LIN messages relevant for this feature are listed in the section “Messages” of the “Data Dictionary”.*

##### Protocol Requirements

###### Schedule Table

***#Hint:*** *The LIN Schedule Table should be documented in the LDF file. The LDF file could be referenced here*

##### Electrical Requirements

***#Hint:*** *The LIN Schedule Table should be documented in the LDF file. The LDF file could be referenced here*

#### “Ethernet xxx”

***#Hint:*** *On Ethernet (wired or wireless) we will see most likely the DoIP, MQTT or V2x protocols. Those protocols are described in separate specifications and are implemented in the Ford AUTOSAR stack. While DoIP might be not that relevant in this scope, MQTT (together with the Google Protocol Buffer (GPB) serialization of the payload) will become important for all features, which are mapped to a Service Oriented Architecture/Communcation (SoC). Application data (SOA APIs), which is transmitted via MQTT, are listed in the data dictionary section “AUTOSAR Interfaces*

#Hint: Those AUTOSAR Classic (Sender/Receiver and Client/Server) Interfaces, which are used by the feature but not managed in a central repository yet, should be listed here.

SOA Service *”.*

***#Link:***[*http://www.mqtt.org*](http://www.mqtt.org)*, https://developers.google.com/protocol-buffers/docs/proto*

### HW I/Os

**#Hin**t: This section lists all hardwired signals relevant for the feature deployment. Those get typically mapped to VSEM EDAS signals – refer to list of connections in corresponding table in chapter ”E/E Architecture → E/E Connections”. If any specific protocol is used to send/receive signal information or multiplex/demultiplex signals on the HW circuit.

#### “HW I/O xxx”

## Requirements on Development Process

# Open Concerns

***#Hint:*** *The following list presents known issues that have to be discussed or clarified over the course of the on-going requirements engineering.*

| ID | Concern Description | e-Tracker Reference | Status | Solution |
| --- | --- | --- | --- | --- |
| 1 | How to fully capture Wake/Sleep requirements. Currently we have a mismatch between what is captured in old EuCD SRD requirements and what is captured in AIS Publisher Interfaces (Publishing Network Sleep Inhibitor, Network Wake Up) |  | Open | Extend AIS attributes? |
| 2 | Clarify how to export Message list entries from CMDB in VSEM |  | Open |  |
| 3 |  |  |  |  |
| 4 |  |  |  |  |

Table 6‑1: Open Concerns

# Revision History

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Revision | Date | Description | Approved by | Responsible |
| A |  | Initial version |  | Jbaden1 |
|  |  |  |  |  |

## Template Revisions

*#Important: Do not change this section*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Version | Rev. | Date | Description | Responsible |
| 0 | 2 | 2015-08-05 | * TOC corrected * Document Properties adapted to match needs of VBA macros | Awegman1 |
| 1 | 0 | 2015-11-16 | * Revision History moved to chapter 7 * Table-Styles removed | Awegman1 |
| 1 | 1 | 2016-03-02 | * Rework according to PCL example | Jbaden1 |
| 1 | 2 | 2016-03-22 | * V1.3: Footer formating corrected (Issue 19) * “Constraints” chapter renamed to “Input Requirements” (Issue 20) | Jbaden1 |
| 1 | 3 | 2016-04-20 | * Broken Wiki links repaired | Jbaden1 |
| 2 | 0 | 2016-05-23 | * Prepared for Specification\_Macros.dotm v2.0 * Additional explanations added to ch. 2.2 “Input Requirements” (ARL and SDS requirements often go here) | Jbaden1 |
| 2 | 1 | 2016-07-08 | * Template version added to footer | Jbaden1 |
| 2 | 2 | 2016-07-15 | * Sample SysML diagrams added * Data Dictionary reworked * Alignment with relevant sections in SRD templated | Jbaden1 |
| 3 | 0 | 2016-09-05 | * Lessons learned from IPRB incorporated | Jbaden1 |
| 4 | 0 | 2016-09-27 | * Alignment with QPIP Feature Function Ownership workstream. Platform Spec renamed to Feature Implementation Spec | Jbaden1 |
| 4 | 1 | 2016-11-04 | * Chapters “Purpose” and “Scope” reworked. | Jbaden1 |
| 4 | 1 | 2016-11-10 | * Subsection for “Logical Service Interfaces” added. | Jbaden1 |
| 5 | 0 | 2017-01-13 | * Meta data updated for specification macros, version 3.1 * SW Unit chapter removed for the time being * Green boxes added for user hints | Jbaden1 |
| 5 | 1 | 2017-01-18 | * Minor editorial changes (e.g. hyperlinks highlighted in comments) | Jbaden1 |
| 5 | 1b | 2017-01-20 | * Some editorial corrections * Substructure of old Network Communication (now Connections) moved to Requirements on Connections | Jbaden1 |
| 6 | 0 | 2018-07-24 | * CR53: * Add new cover sheet * Add disclaimer section * Add the following meta-data to the doc properties for the the new cover sheet   + DocGis1ItemNumber   + DocGis2Classification   + DocType   + DocStatus   + DocIssueDate   + DocReleaseDate * CR63: Update FuSa sharepoint references in templates | Jbaden1 |
| 6 | 0 | 2018-08-06 | * CR81: Incorporate lessons learned from System Service Spec pilot (Vehicle Speed) into AFS and FIS | Jbaden1 |
| 6 | 0 | 2018-09-28 | * Broken links to RE Wiki repaired | Jbaden1 |
| 6 | 0 | 2018-10-31 | * Minor corrections on cover sheet and in footer to be more GIS compliant and VSEM aligned * “Overview” and “Description” exchanged in headings (following common sense) | Jbaden1 |
| 6 | 0 | 2018-11-30 | * Update of Functional Safety sections after review by Functional Safety Team * Initial support for variant handling | Jbaden1 |
| 6 | 0 | 2018-12-01 | * Variant condition fields added consistently * Links updated | Jbaden1 |
| 6 | 0 | 2018-12-11 | * Variant condition fields removed from mapping/allocation tables * Mapping tables simplified * Explanatory text for “Variants” sections revised | Jbaden1 |
| 6 | 0a | 2019-01-04 | * Chapter heading “Inherited Function Requirements” removed. Corresponding table renamed to “Requirements not cascaded”. * E/E Connection table got another column for allocated messages * Naming conventions for Implemented Functions corrected (FncName\_CmpName instead of FncName\_on\_CmpName) * Editorial corrections on the cover sheet * Explanatory text added to “Ethernet” section in chapter “Requirements on Connections” * AIS templates updated. Linked to Wiki page | Jbaden1 |
| 6 | 0a | 2019-01-04 | * Minor restructuring in FuSa chapter – after aligning with ECU Functional Spec * Bugfix: table 13 renamed from FTTI table to FHT table, includes a bug fix: each FSR is allocated to only one ECU/component | Jbaden1 |
| 6 | 0b | 2019-02-04 | * Change: Chapter “Interface Requirements” added to “Implemented Function xxx” section (to have a single chapter for to collect subscriber/publisher interface and mapping requirements which to not conform to the corresponding Data Dictionary objects) * Change: “CAN Interface” subsection renamed to “AIS Interfaces” again. Although several Subscriber/Publisher interface attributes are probably CAN bus specific, other attributes seem to be well suited for other networks than CAN. * Change: Chapter “ECU Specific Requirements” renamed to “Component Specific Requirements” in chapter “Implemented Function xxx”. Table “Requirements not cascaded” renamed to “Component Specific Requirements” and refined to describe changes from Logical Function requirements set more formally. This is also to help during VSEM import to identify those requirements of the Logical Function which cannot be simply carried over to the ECU. * Change: Explanatory text in section “Implemented Function xxx” improved. | Jbaden1 |
| 6 | 0c | 2019-02-05 | * Change: Layout of AIS Interfaces in Data Dictionary reworked to enable Excel Import | Jbaden1 |
| 6 | 0c | 2019-02-20 | * Bugfix: In AIS Interfaces none-picklist fields formatted as invisible | Jbaden1 |
| 6 | 1a | 2019-02-05 | Functional Safety related changes:   * Table “Architectural Redundancy Summary” updated * Section “Functional Flows for FTTI ‘xyz’” added to chapter “Component Interaction Diagrams” * Fault Tolerant Time Summary section added to Functional Safety chapter * Chapter “HW Metrics” added | Jbaden1 |
| 6 | 1a | 2019-04-02 | Headings of “Architectural Redundancy Summary” table clarified | Jbaden1 |
| 6 | 1a | 2019-04-10 | * ASIL Decomposition table moved from Function Spec into the Feature Implementation Spec (ASIL Decomposition of Technical Safety Requirements) * 2 alternative versions of the Function Allocation Table (Standard variant vs. Functional Safety variant) placed next to each other. | Jbaden1 |
| 6 | 1a | 2019-05-31 | * Function Allocation Table split into a base (non FuSa) part and a FuSa part to allow a more flexible mapping of MBSE functions (Logical and Technology) to RE functions (Atomic Logical and Implemented). | Jbaden1 |
| 6 | 1a | 2019-05-31 | * “Input Requirement” section reworked (symmetrically to all other templates). * Sections “Functional Flows for FTTI xyz” and “Fault Tolerant Time Summary” removed, because guidance is not available yet. * “Reference” and “Glossary” section moved back to introduction, i.e., to the very beginning of the document (such that also section 2 can already rely on it). * Some mostly editorial changes per request from FuSa team. | Jbaden1 |
| 6 | 1a | 2019-07-02 | * "Important" box added on cover sheet which points to the macros * “Input Requirements” section renamed to Input Information (after discussion with FuSa team) | Jbaden1 |
| 6 | 1a | 2019-07-17 | * Chapter “Message List” removed from CAN and LIN specific chapters of section “Requirements on Connections” | Jbaden1 |
| 6 | 1a | 2019-10-08 | * Chapter “ASIL Decomposition of Technical Safety Requirements”: Input TSRs are specified in the chapter right above the decomposition table. | Jbaden1 |
| 6 | 1a | 2019-10-09 | * Chapter “Service Oriented Communication” moved to section “Messages” in the Data Dictionary. Details from Central SW Wiki about FNV2 SOA added | Jbaden1 |
| 6 | 1a | 2019-10-25 | * Minor updates for HW IOs/Signals * Subsection “Functional Safety” removed from chapter “Feature Implementation Modeling”. Per requrest from FuSa team since no guidance is available how to model e.g. FHT timing diagram. | Jbaden1 |
| 6 | 1a | 2019-05-11 | * Copyright notice shortened and moved to cover sheet and added to footer (to be compliant [with Ford copyright guidelines](http://www.fgti.ford.com/client/NewFGTI/CopyrightNotice.html)) * Term “Disclaimer” no longer used for what is actually only a copyright notice | Jbaden1 |
| 6 | 1a | 2019-22-11 | * Some minor modifications for the SOA APIs/MQTT Messages in the section “Messages” of the Data Dictionary (section references Service Contracts via the API name) * Some minor updates of the Input/Output mapping tables in section “Requirements on Components” for mappings to SOA APIs and EDAS signals. | Jbaden1 |
| 6 | 1a | 2019-12-05 | * Upstream Documents section added to “Input Requirements/Documents” table * Custom style table formatting removed | Jbaden1 |
| 6 | 1a | 2020-01-07 | * Some fine tuning for naming conventions of E/E components and connections. * List of HW I/O signal types reduced to RF-A, RF-D, D, A, Networked and PWM. * Protocol column added to the E/E connection table | Jbaden1 |
| 6 | 1a | 2020-01-07 | * “HW Metric” and “Architecture Redundancy Summary” sections removed per request from the Functional Architecture Team (based on Governance Board decision [FSTGB-97](mailto:TrackLite%20%23%20FSTGB-97:%20https://www.tracklite.ford.com/prweb/PRAuth/TrackLiteSSO?pyActivity=@baseclass.RedirectAndRunWraper&ThreadName=WorkLinkThread&bPurgeTargetThread=true&AccessGroupName=FSTGB:ProjectAdministrators&Location=pyActivity%3DWork-.Open%26Action%3DReview%26HarnessPurpose%3DReview%26InsHandle%3DFORD-FSTGB-WORK+FSTGB-97)) * “Functional Safety” chapter moved to “Feature Implementation Requirements” section. “Function Allocation” chapter seemed no longer appropriate. | Jbaden1 |
| 6 | 1a | 2020-01-07 | * Ordering of fields in AIS interfaces tables modified to conform with the Macro Template and the Importer Sheet * Page Header: no longer in bold letters | Jbaden1 |
| 6 | 1a | 2020-03-09 | * Missing doc property “LatestSigMappingID” and “LatestAisInterfaceID” added * doc property “CopyrightDate” re-formatted to text and copyright date field in footer corrected * Version numbering re-initialized as 0.1 * Init value of version/revision date set to “yyyy/mm/dd” instead of “yyyy-mm-dd” to be in line with the “Edit Document Property” dialog * Type of “Latest….ID” doc properties changed from Text to Number | Jbaden1 |
| 6 | 1a | 2020-03-11 | * “Mapping” table removed from template. Has been migrated to macro. | Jbaden1 |
| 6 | 1a | 2020-03-13 | * Separate chapter “Technical Safety Requirements” removed. Content already covered by Allocation Table in chapter Function Allocation. * “Implemented Function” replaced by term “Technology Function” | Jbaden1 |

# Appendix

## Data Dictionary

### Logical Signals

**#Hint:** Logical Signals are managed in VSEM in the [*RE Data Dictionary*](https://www.vsemweb.ford.com/tc/launchapp?-attach=true&-s=226TCSession&-o=SoYl_k7px3NrTD&servername=Production_Server).

**#Link**: [*RE Wiki – Adding a Logical Signal or Parameter*](http://wiki.ford.com/display/RequirementsEngineering/Adding+a+Logical+Signal+or+Parameter)

**#Macro**: Add Ins -> Add Requirement macro (select “Logical Signal” as type)

###LSG\_ABC\_00002### LSG\_ABC

<Signal Description>

|  |  |  |
| --- | --- | --- |
| **ASIL** | | Choose an item. |
| **Encoding Type Name** | | <If you reuse an existing encoding type, fill in the Encoding Type Name in this field and delete fields below. Otherwise leave the Encoding Type Name field blank and fill in relevant fields below> |
| Note: An encoding is either discrete or continuous. Delete fields below which are not needed | | |
| **Value**  (Continuous Encoding) | Min Value |  |
| Max Value |  |
| Resolution |  |
| Offset |  |
| **Value**  (Discrete  Encoding) | Value 1 | Interpretation of value 1 |
| Value 2 | … |
| … | … |
|  |  |
|  |  |
| **Unit** | |  |

### Logical Parameters

**#Hint:** Logical Parameters are managed in VSEM in the [*RE Data Dictionary*](https://www.vsemweb.ford.com/tc/launchapp?-attach=true&-s=226TCSession&-o=SoYl_k7px3NrTD&servername=Production_Server).

**#Link**: [*RE Wiki – Adding a Logical Signal or Parameter*](http://wiki.ford.com/display/RequirementsEngineering/Adding+a+Logical+Signal+or+Parameter)

**#Macro:** Add Ins -> Add Requirement macro (select “Logical Parameter” as type)

### Technical Signals

**#Hint:** This section lists all GSDB + GDT + SW signals relevant for the feature deployment.

**#Link**: [*RE Wiki – Adding a Technical Signal or Parameter*](http://wiki.ford.com/display/RequirementsEngineering/Adding+a+Technical+Signal+or+Parameter)

**#Macro:** Add Ins -> Add Requirement macro (select “Technical Signal” as type)

#### GSDB Signals

**#Hint:** This part of the Data Dictionary lists signals, which should go to the GSDB in VSEM, but do not exist in the GSDB in VSEM yet, but are or will be requested for the GSDB. Those would go temporarily to this section in the [*RE Data Dictionary*](https://www.vsemweb.ford.com/tc/launchapp?-attach=true&-s=226TCSession&-o=SoYl_k7px3NrTD&servername=Production_Server) in VSEM.

#### HW I/Os

**#Hint:** This chapter lists signals, which will be mapped to hardwired I/Os. Those get typically refer to VSEM EDAS signals (or input/output signals of device transmittals in VSEM GDT).

#### Diagnostic Interfaces

**#Hint:** This chapter lists Diagnostic Interfaces (DTCs and DIDs), which get mapped to Logical Parameters in context of the Technology Functions in chapter “Parameters” of the Function Interfaces. Those DTC/DID names should match the names in the diagnostics specification (Part 2).

**#ToDo:** Currently the template below is just a proposal. A macro still needs to be created

##### DTCs

###<DTC\_<ID>>### <DTC Name>

<Some Description of the DTC.

Refer to VSEM document “[Diagnostic Fault Coverage and DTC Numbers](https://www.vsemweb.ford.com/tc/launchapp?-attach=true&-s=226TCSession&-o=yAUtrNhnx3NrTDAAAAAAAAAAAAA&servername=Production_Server)

[Design Consideration](https://www.vsemweb.ford.com/tc/launchapp?-attach=true&-s=226TCSession&-o=yAUtrNhnx3NrTDAAAAAAAAAAAAA&servername=Production_Server)”, what to fill into the attributes below>

|  |  |
| --- | --- |
| **Test Period Time** |  |
| **Test Run Criteria,** |  |
| **Enable Criteria (EC)** |  |
| **Applicable** |  |
| **FailureTypeBytes** |  |
| **Test Period Time** |  |
| **Test Run Criteria,** |  |

##### DIDs

**#Hint**: This section lists diagnostic DID which Technical Parameters get mapped to.

**#Todo**: A proper template derived from the Part 2 spec still needs to be created.

### Technical Parameters

**#Hint:** This section lists all Method 2, Method 3 and calibration parameters relevant for the feature deployment.

**#Link**: [*RE Wiki – Adding a Technical Signal or Parameter*](http://wiki.ford.com/display/RequirementsEngineering/Adding+a+Technical+Signal+or+Parameter)

**#Macro:** [Add Ins -> Add Requirement macro](http://wiki.ford.com/display/RequirementsEngineering/How+to+use+the+Specification+Templates#HowtousetheSpecificationTemplates-AddNewRequirement) (select “Technical Parameter” as type)

### Mappings

**#Hint**: This section lists mapping objects for Logical Signals / Parameters to their GSDB + GDT + SW counterparts (1:N mapping is supported). Mapping objects are managed in VSEM in the [*RE Data Dictionary*](https://www.vsemweb.ford.com/tc/launchapp?-attach=true&-s=226TCSession&-o=SoYl_k7px3NrTD&servername=Production_Server).

**#Link:** [RE Wiki – Adding a Signal or Parameter Mapping](http://wiki.ford.com/display/RequirementsEngineering/Adding+a+Signal+or+Parameter+Mapping)

**#Macro:** Add Ins -> Add Requirement macro (select “Mapping” as type)

### Technical Interfaces

**#Hint:** This section lists port/interface details, which define how network/SW/HW signals are published / subscribed.

**#Link:** [*RE Wiki – Adding a Technical Interface*](http://wiki.ford.com/display/RequirementsEngineering/Adding+a+Technical+Interface)

#### AIS Interfaces

**#Hint:** This chapter lists the AIS subscriber and publisher interface objects (managed in VSEM), which are needed to deploy the feature to the E/E architecture. If AIS interfaces do not yet exist in VSEM, those may temporarily be managed as a workaround in the [*RE Data Dictionary*](https://www.vsemweb.ford.com/tc/launchapp?-attach=true&-s=226TCSession&-o=SoYl_k7px3NrTD&servername=Production_Server).

**#Link:** [System Engineering Portal – AIS Release 3.2](https://pd3.spt.ford.com/sites/fede/vsem-spls/Shared%20Documents/02-ais/methods/AIS%20Methods%20Document.pptx?web=1)  
[RE Wiki - AIS Interfaces](http://wiki.ford.com/display/RequirementsEngineering/Adding+a+Technical+Interface#AddingaTechnicalInterface-AisInterfaces)

[*Publisher Interface AIS in VSEM*](https://www.vsemweb.ford.com/tc/launchapp?-attach=true&-s=226TCSession&-o=zjYtY3Jcx3NrTDAAAAAAAAAAAAA&servername=Production_Server)

[*Subscriber Interface AIS in VSEM*](https://www.vsemweb.ford.com/tc/launchapp?-attach=true&-s=226TCSession&-o=LSYtewY7x3NrTDAAAAAAAAAAAAA&servername=Production_Server)

**#Macro:** Add Ins -> Add Requirement macro (select “AIS Subscriber If” or “AIS Publisher If” as type)

##### Publisher Interfaces

##### Subscriber Interfaces

#### AUTOSAR Ports

**#Hint:** Those AUTOSAR Classic (provided and required) ports, which are used by the feature but are not managed in a central repository yet, could be listed here.

### Messages/APIs

#### CAN Bus “<Bus Name>”

**#Hint:** This section gives the relevant extract from the [Central Message Database (CMDB) in VSEM](https://www.vsemweb.ford.com/tc/launchapp?-attach=true&-s=226TCSession&-o=jXfpx2PHx3NrTDAAAAAAAAAAAAA&servername=Production_Server) .

###<MSG\_MessageID### MessageName

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **CAN ID** | **Transmission Mode** | **Period** | **Signal Names** | **Transmitter(s)** | **Receiver(s)** |
|  |  |  |  |  |  |
|  |
|  |
|  |

#### LIN Bus “<Bus Name>”

#### AUTOSAR Interfaces

**#Hint:** Those AUTOSAR Classic (Sender/Receiver and Client/Server) Interfaces, which are used by the feature but not managed in a central repository yet, should be listed here.

#### SOA Service Contracts

**#Hint:** This part of the Data Dictionary lists Service APIs/MQTT messages and embedded data elements, which are used for the Service Oriented Architecture (SOA). If those APIs/MQTT messages already exist e.g. in the [*Central SW Service Catalog*](http://wiki.ford.com/display/CS/Service+Catalog), simply add a reference to those yet.

Information on FNV2 SOA can be found in the ECG wiki page

* MQTT Topic Naming: [*FNV2-SOA: MQTT Topic and Message Structure*](https://www.eesewiki.ford.com/display/ecg/FNV2-SOA%3A+MQTT+Topic+and+Message+Structure?src=sidebar)
* message syntax and proper naming can be found [*SOA API Messaging Guidelines*](https://www.eesewiki.ford.com/x/Q7rKAg)

For examples what to fill into the table fields below refer to [*Central SW Service Catalog*](http://wiki.ford.com/display/CS/Service+Catalog)

###<ServiceContractID>### Service Contract Name

<Service contract purpose/behavior>

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Messaging Pattern | Frequency  (For Data Broadcast Only) | Message Data Element(s)  (Must Match GPB) or applicable CAN signal | Description of Data Element(s) | Topic Name |
| Choose an item. |  | GBP Data element / CAN Signal name 1 | Detailed encoding of data element 1 |  |
| … |  |  |
| GBP Data element / CAN Signal name 1 | Detailed encoding of data element 3 |  |

### Encoding Types

**#Link:** [*RE Wiki – Adding Encoding Types*](http://wiki.ford.com/display/RequirementsEngineering/Adding+an+Encoding+Type)

**#Macro:** Add Ins -> Add Requirement macro (select “Encoding Type” as type)

Document ends here.