|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  | | |  |
|  |  | | |  |
|  | Automatic Engine Idle Shutdown (AEIS) | | |  |
|  | (F000760) | | |  |
|  |  | | |  |
|  |  | | |  |
| Document Type | **Feature Implementation Specification (FIS)** | | |  |
| Template Version | **6.1a** | | |  |
| Document ID | **fis\_aeis\_v1** | | |  |
| Document Location |  | | |  |
| Document Owner | **CP Engelbrecht (CENGELB5)** | | |  |
| 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 |
|  |  | |  |  |
|  |  | |  |  |

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 © 2019, 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 2](#_Toc35000551)

[1 Introduction 2](#_Toc35000552)

[1.1 Document Purpose 2](#_Toc35000553)

[1.2 Document Scope 2](#_Toc35000554)

[1.3 Document Audience 2](#_Toc35000555)

[1.3.1 Stakeholder List 2](#_Toc35000556)

[1.4 Document Organization 2](#_Toc35000557)

[1.4.1 Document Context 2](#_Toc35000558)

[1.4.2 Document Structure 2](#_Toc35000559)

[1.5 Document Conventions 2](#_Toc35000560)

[1.5.1 Requirements Templates 2](#_Toc35000561)

[1.5.1.1 Identification of requirements 2](#_Toc35000562)

[1.5.1.2 Requirements Attributes 2](#_Toc35000563)

[1.6 References 2](#_Toc35000564)

[1.6.1 Ford Documents 2](#_Toc35000565)

[1.6.2 External Documents and Publications 2](#_Toc35000566)

[1.7 Glossary 2](#_Toc35000567)

[1.7.1 Definitions 2](#_Toc35000568)

[1.7.2 Abbreviations 2](#_Toc35000569)

[2 Feature Implementation Overview 2](#_Toc35000570)

[2.1 Description 2](#_Toc35000571)

[2.2 Input Requirements/Documents 2](#_Toc35000572)

[2.3 Lessons Learned 2](#_Toc35000573)

[2.4 Assumptions 2](#_Toc35000574)

[3 Feature Implementation Architecture 2](#_Toc35000575)

[3.1 Functional Architecture 2](#_Toc35000576)

[3.1.1 Description 2](#_Toc35000577)

[3.1.2 Function List 2](#_Toc35000578)

[3.1.3 Signal List 2](#_Toc35000579)

[3.2 Physical Architecture 2](#_Toc35000580)

[3.2.1 E/E Architecture 2](#_Toc35000581)

[3.2.1.1 E/E Architecture Variants 2](#_Toc35000582)

[3.2.1.1.1 E/E Architecture “Architecture Variant 1” 2](#_Toc35000583)

[3.2.1.1.2 E/E Architecture “Architecture Variant 2” 2](#_Toc35000584)

[3.2.1.2 E/E Components 2](#_Toc35000585)

[3.2.1.3 E/E Connections 2](#_Toc35000586)

[3.2.1.4 Signal List 2](#_Toc35000587)

[3.2.2 Software Component Architecture 2](#_Toc35000588)

[3.2.2.1 Description 2](#_Toc35000589)

[3.3 Function Deployment 2](#_Toc35000590)

[3.3.1 Deployment Variants 2](#_Toc35000591)

[3.3.1.1 Deployment “Variant 1” 2](#_Toc35000592)

[3.3.2 Function Allocation 2](#_Toc35000593)

[4 Feature Implementation Modeling 2](#_Toc35000594)

[4.1 Component Interaction Diagrams 2](#_Toc35000595)

[4.1.1 Scenario: “System Startup / Shutdown” 2](#_Toc35000596)

[4.1.2 Scenario: “Normal Operation” 2](#_Toc35000597)

[4.2 Component Interface Behavior Diagrams 2](#_Toc35000598)

[5 Feature Implementation Requirements 2](#_Toc35000599)

[5.1 Functional Safety 2](#_Toc35000600)

[5.1.1 ASIL Decomposition of Technical Safety Requirements 2](#_Toc35000601)

[5.2 Requirements on Components 2](#_Toc35000602)

[5.2.1 Component 1 2](#_Toc35000603)

[5.2.1.1 Technology Function “MyLogicalFunctionA\_Component1” 2](#_Toc35000604)

[5.2.1.1.1 Function Interfaces 2](#_Toc35000605)

[5.2.1.1.1.1 Inputs 2](#_Toc35000606)

[5.2.1.1.1.2 Outputs 2](#_Toc35000607)

[5.2.1.1.1.3 Parameters 2](#_Toc35000608)

[5.2.1.1.1.4 Interface Requirements 2](#_Toc35000609)

[5.2.1.1.2 Function Requirements 2](#_Toc35000610)

[5.2.1.1.2.1 Component Specific Requirements 2](#_Toc35000611)

[5.2.1.2 (Technology) Function “MyLogicalFunctionB\_Component1” 2](#_Toc35000612)

[5.2.2 Component 2 2](#_Toc35000613)

[5.3 Requirements on Connections 2](#_Toc35000614)

[5.3.1 Networks 2](#_Toc35000615)

[5.3.1.1 “CAN Bus xxx” 2](#_Toc35000616)

[5.3.1.1.1 Protocol Requirements 2](#_Toc35000617)

[5.3.1.1.2 Electrical Requirements 2](#_Toc35000618)

[5.3.1.2 “LIN Bus xxx” 2](#_Toc35000619)

[5.3.1.2.1 Protocol Requirements 2](#_Toc35000620)

[5.3.1.2.1.1 Schedule Table 2](#_Toc35000621)

[5.3.1.2.2 Electrical Requirements 2](#_Toc35000622)

[5.3.1.3 “Ethernet xxx” 2](#_Toc35000623)

[5.3.2 HW I/Os 2](#_Toc35000624)

[5.3.2.1 “HW I/O xxx” 2](#_Toc35000625)

[5.4 Requirements on Development Process 2](#_Toc35000626)

[6 Open Concerns 2](#_Toc35000627)

[7 Revision History 2](#_Toc35000628)

[8 Appendix 2](#_Toc35000630)

[8.1 Data Dictionary 2](#_Toc35000631)

[8.1.1 Logical Signals 2](#_Toc35000632)

[8.1.2 Logical Parameters 2](#_Toc35000633)

[8.1.3 Technical Signals 2](#_Toc35000634)

[8.1.3.1 GSDB Signals 2](#_Toc35000635)

[8.1.3.2 HW I/Os 2](#_Toc35000636)

[8.1.3.3 Diagnostic Interfaces 2](#_Toc35000637)

[8.1.3.3.1 DTCs 2](#_Toc35000638)

[8.1.3.3.2 DIDs 2](#_Toc35000639)

[8.1.4 Technical Parameters 2](#_Toc35000640)

[8.1.5 Mappings 2](#_Toc35000641)

[8.1.6 Technical Interfaces 2](#_Toc35000642)

[8.1.6.1 AIS Interfaces 2](#_Toc35000643)

[8.1.6.1.1 Publisher Interfaces 2](#_Toc35000644)

[8.1.6.1.2 Subscriber Interfaces 2](#_Toc35000645)

[8.1.6.2 AUTOSAR Ports 2](#_Toc35000646)

[8.1.7 Messages/APIs 2](#_Toc35000647)

[8.1.7.1 CAN Bus “<Bus Name>” 2](#_Toc35000648)

[8.1.7.2 LIN Bus “<Bus Name>” 2](#_Toc35000649)

[8.1.7.3 AUTOSAR Interfaces 2](#_Toc35000650)

[8.1.7.4 SOA Service Contracts 2](#_Toc35000651)

[8.1.8 Encoding Types 2](#_Toc35000652)

**List of Figures**

[Figure 3‑1: Functional Architecture 2](#_Toc35000653)

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

[Figure 3‑3 E/E Architecture (Network Topology Style) 2](#_Toc35000655)

[Figure 3‑4: AUTOSAR compliant SW Component Architecture 2](#_Toc35000656)

**List of Tables**

[Table 1‑1: Electrical Architecture(s) referenced in this document 2](#_Toc35000657)

[Table 1‑2: Ford internal Documents 2](#_Toc35000658)

[Table 1‑3: External documents and publications 2](#_Toc35000659)

[Table 1‑4: Definitions used in this document 2](#_Toc35000660)

[Table 1‑5: Abbreviations used in this document. 2](#_Toc35000661)

[Table 6: Input Requirements/Documents 2](#_Toc35000662)

[Table 3‑1: List of Functions 2](#_Toc35000663)

[Table 3‑2: Electrical Components 2](#_Toc35000664)

[Table 3‑3: E/E Connections 2](#_Toc35000665)

[Table 3‑4: Sample Deployment Diagram 2](#_Toc35000666)

[Table 3‑5: Function Allocation Table (Basic) 2](#_Toc35000667)

[Table 3‑6: Function Allocation Table (Functional Safety Extension) 2](#_Toc35000668)

[Table 5‑1: ASIL Decomposition Table 2](#_Toc35000669)

[Table 5‑2: Input Signal mappings of Function “MyLogicalFunctionA\_Component1” 2](#_Toc35000670)

[Table 5‑3: Output Signal mappings of Function “MyLogicalFunctionA\_Component1”A 2](#_Toc35000671)

[Table 5‑4: Parameter mappings of Function “MyLogicalFunctionA\_Component1” 2](#_Toc35000672)

[Table 5‑5: Component Specific Requirements 2](#_Toc35000673)

[Table 5‑6: Inherited Requirements 2](#_Toc35000674)

[Table 6‑1: Open Concerns 2](#_Toc35000675)

# 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 Automatic Engine Idle Shutdown 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 CP Engelbrecht (cengelb5) the AEIS 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.*

## 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** |
| --- | --- | --- | --- | --- |
|  | AEIS Feature Document | aeis feature document\_v2.2 | FD0 | VSEM F000760 |
|  |  |  |  |  |

Table 1‑2: 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 1‑3: External documents and publications

## Glossary

### Definitions

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

Table 1‑4: Definitions used in this document

# Feature Implementation Overview

### Abbreviations

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

Table 1‑5: Abbreviations used in this document.

## Description

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

The purpose of Automatic Engine Idle Shutdown (AEIS) is to shutdown the Powerpack and Ignition when the vehicle has been left running and the system does not detect any "Intent to Drive" for a specified amount of time. (Refer to the Feature Document for a more detailed description)

AEIS is implemented on all platforms and electrical architectures.

The feature is implemented in the ECM/PCM and interfaces directly with the following modules:

* BCM
* IPC
* APIM (via IPC)
* APIM (directly)

## 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).

* Legal requirements
* Electrical requirements
* Communication requirements
* Diagnostic requirements

## 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.

* Configuration issues between Cluster and PCM
* Owner’s Manual

## 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 course of 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.

No Assumptions Specified

# 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)

This is the basic view of the functionality of AEIS. Please see the Deployment Diagram for a more comprehensive view.



Figure 3‑1: Functional Architecture

### 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** |
| Fn000482 | AEIS - Automatic Engine Idle Shutdown - PCM Function | Main AEIS functionality; detect driver presence and shutdown powerpack |
| Fn008514 | PttB PCM Function | Disable AEIS while PttB is Active |
| Fn002329 | Power Take Off - Power Take Off - PCM Function | Disable AEIS while PTO is Active |
| Fn009454 | Driver Info - AEIS Cluster Messages - IPC Function | Display driver information messages on the Instrument Panel Cluster |
| Fn009491 | AEIS - Ignition Cutoff - BCM Function | Send request to BCM to turnoff ignition and PCM |
|  | Vehicle Settings Disable | Disable Feature in Vehicle Settings for Keycycle |
|  | Vehicle Speed | Determines Feature State |
|  | Transmission State | Determines Feature State |
|  | Pedal State | Determines Feature State |
|  | Powerpack Shutdown | Engine Shutdown request |

Table 3‑1: List of Functions

### Signal List

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

## 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).

The feature is receiving a configuration independence update, allowing the feature to notify the required modules in the system boundary to automatically configure depending on key type.

Below are all the possible combinations containing old or new ECU software and how they will be configured for each case.

It is assumed that IPC and APIM will override their internal config when they see valid inputs and keep their configuration if it’s an uncoordinated change.

|  |  |  |  |
| --- | --- | --- | --- |
| ECM | IPC | SYNC | Config |
| Old | Old | Old | Direct config |
| Old | Old | New | SYNC doesn’t receive new CAN Signal so uses old Direct config |
| Old | New | Old | Cluster doesn’t receive new CAN Signal so uses old Direct config |
| Old | New | New | Neither receive new CAN Signal so use old Direct config |
| New | Old | Old | Direct config - new CAN signal is ignored |
| New | Old | New | Cluster uses old Direct config, but have mis-config protection on SYNC |
| New | New | Old | SYNC uses old Direct config, but have mis-config protection on Cluster |
| New | New | New | CAN message for config - Both modules have mis-config protection |

Table 3‑1.1: Possible Configuration States

##### 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 3‑2 E/E Architecture (SysML Style)) or enhanced the network topology, which Netcom generates from its master Excel sheet diagrams (refer to Figure 3‑3 E/E Architecture (Network Topology Style)) according to the needs of the feature.

This E/E Architecture variant shows the features and interfaces necessary to actualize AEIS at the ECU level.



Figure 3‑2 E/E Architecture

##### E/E Architecture “Architecture 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** |
| ECM/PCM | Powertrain Control Module |
| BCM | Body Control Module |
| IPC | Instrument Panel Cluster |
| APIM | Auxiliary Protocol Interface Module |
| SCCM | Steering Column Control Module |
| MCHLD | Multiplexes CAN Signals |
| PEDIN | Pedal Inputs |
| TRANS | Transmission State Inputs |
| PttB | Power to the Box |
| PTO | Power Take Off |
| CCSD | Computer Controlled Shut Down |
| Engine | Block Representing the Engine |
| AEIS | Automatic Engine Idle Shutdown core functions |

Table 3‑2: 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** | **Description** | **Allocated Messages** | **Connected Nodes** |
| CAN FD | CAN FD1 |  | 0x43F, 0x225 | ECM/PCM |
| HS1 CAN | High Speed CAN | High Speed CAN 1 | 0x43F | BCM |
| HS3 CAN | High Speed CAN | High Speed CAN 3 | 0x283, 0x225 | IPC, APIM |

Table 3‑3: E/E Connections

#### Signal List

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

|  |  |  |
| --- | --- | --- |
| **Signal Name** | **Type** | **Description** |
| EngIdlShutDown\_D\_Stat | CAN | Signal containing the Status of automatic shutdown |
| EngIdlShutDown\_B\_RqDrv | CAN | Signal from IPC containing Vehicle Operator Disable Request |
| EcoInhbt\_B\_RqHvac | CAN | Signal containing Eco Inhibit Status |
| EngIdlShutDown\_D\_Type | CAN | Signal containing vehicle configuration and AEIS type |

## 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) |
| Base | All Vehicles | Type Keyless |

#### 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 3‑4: Functional Deployment Diagram

#### Deployment “Variant 2”



### 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 (“Function Allocation Table (Functional Safety Extension)”) 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 |
| --- | --- | --- | --- |
| SCCM | | Steering Wheel OK Button |  |
| APIM | | Vehicle Settings |  |
| IPC | | Display Driver Info Messages |  |
| BCM | | Power down PCM |  |
| PCM | MCHLD | Multiplex CAN Signals |  |
| PEDIN | Provide Clutch Pedal Position and QF |  |
| Provide Brake Pedal Position and QF |
| Provide Acc Pedal Position and QF |
| TRANS | Provide Gear Position and QF |  |
| AEIS | Activate AEIS |  |
| Disable AEIS |
| Disable Requests | PttB |  |
| PTO |
| GPF/DPF Regen |
| ELHLDS | Provide Ignition Status |  |
| VSPDC | Provide Vehicle Speed and QF |  |
| CCSD | Shutdown Engine |  |

Table 3‑5: 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

Feature is not Functional Safety critical. The HARA resulted in no ASIL ratings greater than QM.

## Requirements on Components

### BCM

#### Implemented Function “Switch off Run/Start BUS”

***#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)* |
| pfis\_d\_st\_clstr | EngIdlShutDown\_D\_Stat | N/A | N/A |  |

Table 5‑2: Input Signal mappings of Function “Switch off Run/Start BUS”

###### Outputs

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Logical Signal Name** | **Technical Signal Name** | **Mapping Details**  *(Conditional)* | **Publisher Interface** | **Connection**  *(Optional)* |
| N/A | N/A | N/A | N/A | N/A |
|  |  |  |  |  |

Table 5‑3: Output Signal mappings of Function “Switch off Run/Start BUS”

###### Parameters

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Logical Parameter Name** | **Technical Parameter Name** | **Mapping Details** *(Conditional)* | **Method** | **Method Details** |
| Name should be a Word reference to the “*Logical Parameters*” name bookmark in the Data Dictionary | Name should be a Word reference to the “*Technical Parameters*” name bookmark in the Data Dictionary | If mapping is not 1:1 you might reference a Mapping description object from the *Mappings* section | Choose an item. | Depends on Method selection. For Method 2 a DID including start bit and length could be given. For Central Car Config a signal could be referenced |
|  |  |  |  |  |

Table 5‑4: Parameter mappings of Function “Switch off Run/Start BUS”

###### 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** |
|  | BCM to Shutdown PCM | The BCM waits 1 to 4 seconds after confirmation of engine shutdown and then cuts power to PCM |
|  |  |  |
| … |  |  |

Table 5‑6: Inherited Requirements

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Warning No.** | **CAN Signal** | **Message State** | **Type** | **Description** | **Display Options** | **Tx** | **Rx** |
| 1 | EngIdlShutDown\_D\_Stat | 0x0:  AEIS is disabled or more than 30 seconds away from shutting down  0x1:  The Instrument Panel displays a 30 second count-down prior to engine shutdown.  0x2:  The BCM waits 1 to 4 seconds after confirmation of engine shutdown and then turns off run/start bus |  | On receiving a value of 0x1 the IPC should display a shutdown pending message and give Driver the option to prevent shutdown  On receiving a value of 0x2 the BCM shall turn off the run/start bus (KL15) | N/A | ECM/PCM | BCM /IPC |

Table 22: CAN Signals for Function “Power down PCM\_BCM”

###### 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”

### IPC

#### Implemented Function “Display Messages to Driver”

##### Function Interfaces

###### Inputs

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Logical Signal Name** | **Technical Signal Name** | **Mapping Details** *(Conditional)* | **Subscriber Interface** | **Connection**  (*Optional)* |
| pfis\_d\_st\_clstr | EngIdlShutDown\_D\_Stat |  | NA |  |
| TBD | EngIdlShutDown\_D\_Type |  |  |  |

Table 21: Input Signal mappings of Function “Control Ignition Switch Position\_BCM”

###### Outputs

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Logical Signal Name** | **Technical Signal Name** | **Mapping Details**  *(Conditional)* | **Publisher Interface** | **Connection**  *(Optional)* |
| mux\_idlshtdwn\_rq | EngIdlShutDown\_B\_RqDrv |  | NA |  |
|  |  |  |  |  |

Table 22: Output Signal mappings of Function “Control Ignition Switch Position\_BCM”

###### Parameters

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Logical Parameter Name** | **Technical Parameter Name** | **Mapping Details** *(Conditional)* | **Method** | **Method Details** |
| NA | NA | NA | Choose an item. | NA |

Table 23: Parameter mappings of Function “Control Ignition Switch Position\_BCM”

###### Interface Requirements

See Data Dictionary: ‘AIS Interfaces’

##### Function Requirements

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Warning No.** | **CAN Signal** | **Message State** | **Type** | **Description** | **Display Options** | **Tx** | **Rx** |
| 1 | EngIdlShutDown\_D\_Stat | 0x0:  AEIS is disabled or more than 30 seconds away from shutting down  0x1:  The Instrument Panel displays a 30 second count-down prior to engine shutdown.  0x2:  The BCM waits 1 to 4 seconds after confirmation of engine shutdown and then cuts power to PCM |  | On receiving a value of 0x1 the IPC should display a shutdown pending message and give Driver the option to prevent shutdown  On receiving a value of 0x2 the BCM shall turn off the run/start bus (KL15) | TBD | ECM/PCM | IPC |
| 2 | EngIdlShutDown\_B\_RqDrv | 0x0:  No inhibit (AEIS feature enabled)  0x1:  Inhibit (AEIS feature disabled) |  | Sent by the IPC to enable/disable the AEIS feature at any time. | “30min Max Idle” - ON/OFF | IPC | ECM/PCM |
| 3 | EngIdlShutDown\_D\_Type | 0x0:  Feature disabled  0x1:  Type Keyed  0x2:  Type Keyless  0x3:  Type Fleet |  | Sent by ECM to tell IPC which messages to display.  AEIS SYNC and Warning messages will be displayed when a value of 0x2 is received by the IPC | 0x0:  None  0x1:  None  None  0x2:  30 seconds with override  0x3:  30 seconds with NO override | ECM/PCM | IPC |

Table 24: Warning Proposals for Function “Display Messages to Driver\_IPC”

### ECM/PCM

#### Implemented Function “Control Automatic Engine Idle Shutdown”

##### Function Interfaces

###### Inputs

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Logical Signal Name** | **Technical Signal Name** | **Mapping Details** *(Conditional)* | **Subscriber Interface** | **Connection**  (*Optional)* |
| mux\_idlshtdwn\_rq | EngIdlShutDown\_B\_RqDrv |  | NA |  |
|  |  |  |  |  |

Table 21: Input Signal mappings of Function “Control Automatic Engine Idle Shutdown\_PCM”

###### Outputs

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Logical Signal Name** | **Technical Signal Name** | **Mapping Details**  *(Conditional)* | **Publisher Interface** | **Connection**  *(Optional)* |
| pfis\_d\_st\_clstr | EngIdlShutDown\_D\_Stat |  | NA |  |
| TBD | EngIdlShutDown\_D\_Type |  |  |  |

Table 22: Output Signal mappings of Function “Control Automatic Engine Idle Shutdown\_PCM”

###### Parameters

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Logical Parameter Name** | **Technical Parameter Name** | **Mapping Details** *(Conditional)* | **Method** | **Method Details** |
| NA | NA | NA | Choose an item. | NA |

Table 23: Parameter mappings of Function “Control Automatic Engine Idle Shutdown\_PCM”

###### Interface Requirements

See Data Dictionary: ‘AIS Interfaces’

##### Function Requirements

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Warning No.** | **CAN Signal** | **Message State** | **Type** | **Description** | **Display Options** | **Tx** | **Rx** |
| 1 | EngIdlShutDown\_D\_Stat | 0x0:  AEIS is disabled or more than 30 seconds away from shutting down  0x1:  The Instrument Panel displays a 30 second count-down prior to engine shutdown.  0x2:  The BCM waits 1 to 4 seconds after confirmation of engine shutdown and then cuts power to PCM |  | On receiving a value of 0x1 the IPC should display a shutdown pending message and give Driver the option to prevent shutdown  On receiving a value of 0x2 the BCM shall turn off the run/start bus (KL15) | TBD | IPC | ECM/PCM |
| 2 | EngIdlShutDown\_B\_RqDrv | 0x0:  No inhibit (AEIS feature enabled)  0x1:  Inhibit (AEIS feature disabled) |  | Sent by the IPC to enable/disable the AEIS feature at any time. | “30min Max Idle” - ON/OFF | ECM/PCM | IPC |
| 3 | EngIdlShutDown\_D\_Type | 0x0:  Feature disabled  0x1:  Type Keyed  0x2:  Type Keyless  0x3:  Type Fleet |  | Sent by ECM to tell IPC and APIM which messages to display | 0x0:  None  None  0x1:  None  None  0x2:  30 seconds with override  Disable Slider  0x3:  30 seconds with No override  None | ECM/PCM | IPC  APIM |

Table 24: Outputs for Function “Control Automatic Engine Idle Shutdown\_PCM”

### APIM

#### Implemented Function “Configure Module”

##### Function Interfaces

###### Inputs

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Logical Signal Name** | **Technical Signal Name** | **Mapping Details** *(Conditional)* | **Subscriber Interface** | **Connection**  (*Optional)* |
| TBD | EngIdlShutDown\_D\_Type |  |  |  |

Table 21: Input Signal mappings of Function “Configure Module\_APIM”

###### Outputs

###### Parameters

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Logical Parameter Name** | **Technical Parameter Name** | **Mapping Details** *(Conditional)* | **Method** | **Method Details** |
| NA | NA | NA | Choose an item. | NA |

Table 23: Parameter mappings of Function “Configure Module\_APIM”

###### Interface Requirements

See Data Dictionary: ‘AIS Interfaces’

##### Function Requirements

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Warning No.** | **CAN Signal** | **Message State** | **Type** | **Description** | **Display Options** | **Tx** | **Rx** |
| 1 | EngIdlShutDown\_D\_Type | 0x0:  Feature disabled  0x1:  Type Keyed  0x2:  Type Keyless  0x3:  Type Fleet |  | Sent by ECM to tell SYNC which messages to display.  AEIS SYNC disable option in vehicle settings will be displayed when a value of 0x2 is received by the APIM | 0x0:  None  0x1:  None  0x2:  Disable Slider  0x3:  None | ECM/PCM | APIM |

Table 24: Autoconfig Signals for Function “Configure Module\_APIM”

## 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 |  |  |  |  |
| 2 |  |  |  |  |
| 3 |  |  |  |  |
| 4 |  |  |  |  |

Table 6‑1: Open Concerns

# Revision History

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Revision | Date | Description | Approved by | Responsible |
| A |  | Initial version |  | Cengelb5 |
|  |  |  |  |  |

## 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)

### 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.