EM Registers

<<Feature>>

F003951

|  |  |  |  |
| --- | --- | --- | --- |
| Document Type | **Feature Implementation Specification (FIS)** | |  |
| Template Version | **6.1a** | |  |
| SysML Report Template Version | **O Beta (2021/03/10)** | |  |
| Document ID | **featureimplementationspecification\_dev.docx** | |  |
| Document Location |  | |  |
| Document Owner | **Smith, George (G.) <gsmit560@ford.com>** | |  |
| Document Revision | **FIS0** | |  |
| Document Status | **Draft** | |  |
| Date Issued | **2021/06/07** | |  |
| Date Revised | **2021/06/07** | |  |
| Document Classification | GIS1 Item Number: | **27.60/35** |  |
| GIS2 Classification: | **Confidential** |

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

**Disclaimer**

**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.**

This document contains information developed and accumulated by and for FORD MOTOR COMPANY. As such, it is a proprietary document, which, if disseminated to unauthorized persons, would provide others with restricted information, data, or procedures not otherwise available, exposing the FORD MOTOR COMPANY to potential harm.

Employees and suppliers having custody of this specification or authorized to use it must be cognizant of its proprietary nature and ensure that the information herein is not made available to unauthorized persons.

FORD MOTOR COMPANY reserves the right to protect this work as an unpublished copyrighted work in the event of an inadvertent or deliberate unauthorized publication. FORD MOTOR COMPANY also reserves its rights under copyright laws to protect this work as a published work.

This document or portions thereof shall not be distributed outside FORD MOTOR COMPANY without prior written consent. Refer all questions concerning disclosure to the author(s) or any duly authorized representative of Ford Motor Company.

**Copyright** © **2021 Ford Motor Company**

**Contents**

1 Introduction 6

1.1 Document Purpose 6

1.2 Document Scope 6

1.3 Document Audience 6

1.3.1 Stakeholder List 6

1.4 Document Organization 6

1.4.1 Document Context 6

1.4.2 Document Structure 6

1.5 Document Conventions 7

1.5.1 Requirements Templates 7

1.5.1.1 Identification of requirements 7

1.5.1.2 Requirement Attributes 7

2 Feature Implementation Overview 8

2.1 Description 8

2.2 Input Requirements 8

2.3 Assumptions 9

2.4 References 9

2.4.1 Ford Documents 9

2.4.2 External Documents and Publications 10

2.5 Glossary 10

2.5.1 Definitions 10

2.5.2 Abbreviations 11

3 Feature Implementation Architecture 12

3.1 Functional Architecture 12

3.1.1 Description 12

3.1.2 Function List 12

3.1.2.1 Functions of 13

3.1.2.2 Functions of 14

3.1.3 Signal List 14

3.2 Physical Architecture 18

3.2.1 E/E Architecture 18

3.2.1.1 E/E Architecture Variants 19

3.2.1.2 E/E Components 20

3.2.1.3 E/E Connections 20

3.2.1.4 Signal List 22

3.2.2 Software Component Architecture 22

3.2.2.1 Description 22

3.3 Function Deployment 22

3.3.1 Deployment Variants 23

3.3.1.1 Main (Only) variant 23

3.3.1.2 Deployment “Variant 1” 23

3.3.2 Function Allocation 24

3.3.2.1 Functional Safety 25

4 Feature Implementation Modeling 27

4.1 Component Interaction Diagrams 27

4.1.1 Scenario: “System Startup / Shutdown” 27

4.1.2 Scenario: “Normal Operation” 27

4.1.3 Functional Safety 28

4.1.3.1 Fault Handling Time Analysis 28

4.1.3.2 Requirements Derivation Diagram 29

4.2 Component Interface Behavior Diagrams 29

5 Feature Implementation Requirements 30

6 Open Concerns 31

7 Revision History 32

8 Appendix 33

8.1 Data Dictionary 33

8.1.1 Logical Signals 33

8.1.2 Logical Parameters 33

8.1.3 Technical Signals 33

8.1.3.1 GSDB Signals 34

8.1.3.2 Service Oriented Communication 34

8.1.3.3 Hardwired Signals 34

8.1.3.4 Diagnostic Interfaces 34

8.1.4 Technical Parameters 36

8.1.5 Mappings 36

8.1.6 Technical Interfaces 37

8.1.6.1 AIS Interfaces 37

8.1.6.2 Service Oriented Communcation (SoC) Interfaces 38

8.1.6.3 AUTOSAR Ports (SW Interfaces) 38

8.1.7 Messages 38

8.1.7.1 CAN Bus “<Bus Name>” 38

8.1.7.2 LIN Bus “<Bus Name>” 39

8.1.8 Encoding Types 39

**List of Figures**

Figure 1: 21

Figure 2:  24

Figure 3: 30

**List of Tables**

Table 1‑1: Electrical Architecture(s) referenced in this document 6

Table 1‑2: Functions referenced in this document 6

Table 2‑1: Ford Internal Documents 10

Table 2‑2: Ford Internal Documents *(not specified in SysML model)* 11

Table 2‑3: External Documents and Publications 11

Table 2‑4: External Documents and Publications *(not specified in SysML model)* 11

Table 2‑5: Definitions Used In This Document 12

Table 2‑6: Abbreviations used in this document 12

Table 3‑1: List of Functions 15

Table 3‑2: List of ‑‑ Functions 15

Table 3‑3 List of signals sent by 18

Table 3‑4 List of signals received by 19

Table 3‑5: Electrical Components 21

Table 3‑6: E/E Connections for 23

Table 3‑7: Function Allocation Table 26

Table 3‑8: Function Allocation Table 27

Table 3‑9: Architectural Redundancy Summary 27

Table 4‑1: Fault Handling Time Table 30

Table 5‑1: Input Signal mappings of ‑ 34

Table 5‑2: Output Signal mappings of ‑ 35

Table 5‑3: Parameter mappings of ‑ 36

Table 5‑4: Component Specific Requirements 37

Table 5‑5: Inherited Requirements 37

Table 5‑6: Input Signal mappings of ‑ 39

Table 5‑7: Output Signal mappings of ‑ 40

Table 5‑8: Parameter mappings of ‑ 40

Table 5‑9: Component Specific Requirements 40

Table 5‑10: Inherited Requirements 40

Table 5‑11: Input Signal mappings of Component: ‑ 41

Table 5‑12: Output Signal mappings of Component: ‑ 42

# 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

This FIS describes the deployment of the feature EM Registers <Feature> to the following electrical architecture(s):

*No Electrical Architecture found.*

## Document Audience

The FIS is authored by - . 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.

### Stakeholder List

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

## References

### Ford Documents

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

| **Reference** | **Title** | **Doc. ID** | **Revision** | **Document Location** |
| --- | --- | --- | --- | --- |
| Ford GIS Standard | Ford GIS Standard |  |  |  |

Table 1‑2: Ford internal Documents

### External Documents and Publications

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

| **Reference** | **Document / Publication** |
| --- | --- |
| IEEE Std 1012-2004 IEEE Standard for Software Verification and Validation |  |
| ISO/IEC 19500-2:2003 |  |
| UML Testing Profile (UTP), v1.2 |  |
| Wikipedia | https://en.wikipedia.org/wiki/Concept |

Table 1‑3: External documents and publications

## Glossary

### Definitions

| **Definition** | **Description** |
| --- | --- |
| APIM | Auxillary Protocol Interface Module (Sync Gen 4.2) with Cluster |
| BeforeClosedPosition | The last stationary positions that the registers were in before the registers were closed. To be used to return the registers back to their previous position when the registers are requested to open by the User. |
| calibrate | A process that will provide future accurate positioning for the registers. The process entails moving the registers to the end points and recording the feedback from the sensors. This feedback is used to recalibrate the future voltage commands to move the registers. |
| CIS | Controls Interface Specification |
| High speed | Approximately more than 52 mph (83 kph) |
| HOME | Value that denotes the Home position the registers will move to when the car is turned off. |
| LastPosition | Denotes the last stationary position that the registers were in. For example the position before the vehicle was turned OFF, before a Cycling option was selected, or before another feature requested to move the registers. |
| Low speed | Approximately 12 to 36 mph (19 to 58 kph ) |
| MaxPositionError | The maximum degree error allowed for a register position before the position is considered incorrect. |
| Medium speed | Approximately 36 mph to 52 mph (58 to 83 kph) |
| RCCM | Remote Climate Control Module |
| Specified point to point rate | Rate at which register vanes move. |
| StaticSelection | Register movement selection that is not a dynamic option |
| status of EM registers |  |
| term | A representation of a Concept expressed in Natural Language. In the vocabulary of a Domain of Discourse a term enables common understanding of domain concepts. |
| term glossary | A term glossary is a table of agreed upon definitions for terms used in project development that may provide clarity or avoid confusion to stakeholders. |
| Threshold | The minimum change that the register will move to. |
| TLA | Three Letter Acronym |
| Very Low Speed | Approximately 0 to 12 mph (0 to 19 kph) |

Table 1‑4: Definitions used in this document

### Abbreviations

| **Abbr.** | **Stands for** | **Description** |
| --- | --- | --- |
| ATLA | Another Three Letter Acronym |  |

Table 1‑5: Abbreviations used in this document.

# Feature Implementation Overview

## Description

F003951 EM Registers

## Input Requirements/Documents

|  |  |  |  |
| --- | --- | --- | --- |
| **Reference**  (Reference as listed in ch. “References”) | **Section/Requirement** | **Description** | **Derived Requirement**  (optional – reference to requirement in ch. “Feature Implementation Requirements”) |
| **Feature/Function Requirements** | | | |
|  | <Example:  id + title of relevant Function Spec> | <Example: “Function requirements of Logical Function …”> | <Note: If you reference a requirement in this column, then that requirement should have a trace link in its [“Source”/”Source Req.” attribute](http://wiki.ford.com/display/RequirementsEngineering/Requirements+Attributes) field pointing back to the input requirement (or to a requirement inside the input document) given in this table row> |
|  |  |  |  |
| **Ford Engineering Standards** | | | |
|  | <Example: some SDS (requirement)> |  |  |
|  |  |  |  |
| **Legal Regulations** | | | |
|  | Compliance with FMVSS101 | The Feature shall comply with FMVSS101. |  |
|  |  |  |  |
| **Industry Standards** | | | |
|  | ISO 26262 | The system shall be developed according to Ford's implementation of Functional Safety. |  |
|  |  |  |  |
| **Other Sources** | | | |
|  | Air Register Life Cycle from IP-0114 | Registers shall withstand at least 7000 cycles. |  |
|  | Air Register Applied Loads from IP-0117 | Register shall withstand loads of 100N and remain functional. |  |
|  |  |  |  |
|  | Personalizable | EM Registers shall be personalizable. (PPP and/or Preset) |  |
|  | HMI Feedback | EM Registers shall give visual feedback to the User through the HMI. |  |
|  | Registers remember users last chosen position Memory | EM Registers shall remember Users last chosen registers directions. |  |
|  | HVAC Vent Air Flow Direction Interface | EM Registers shall control the direction of air exiting the Registers by interfacing with the HMI. |  |
|  | Fast and Accurate | EM Registers shall provide fast and accurate control of the air vents. |  |
|  | Electrically Actuated | EM Registers has electrically actuated and not human actuated air vents. |  |
|  |  |  |  |

Table 2‑1: Input Requirements/Documents

## Lessons Learned

No lessons learned specified.

## Assumptions

No Assumptions specified.

# Feature Implementation Architecture

## Functional Architecture

### Description

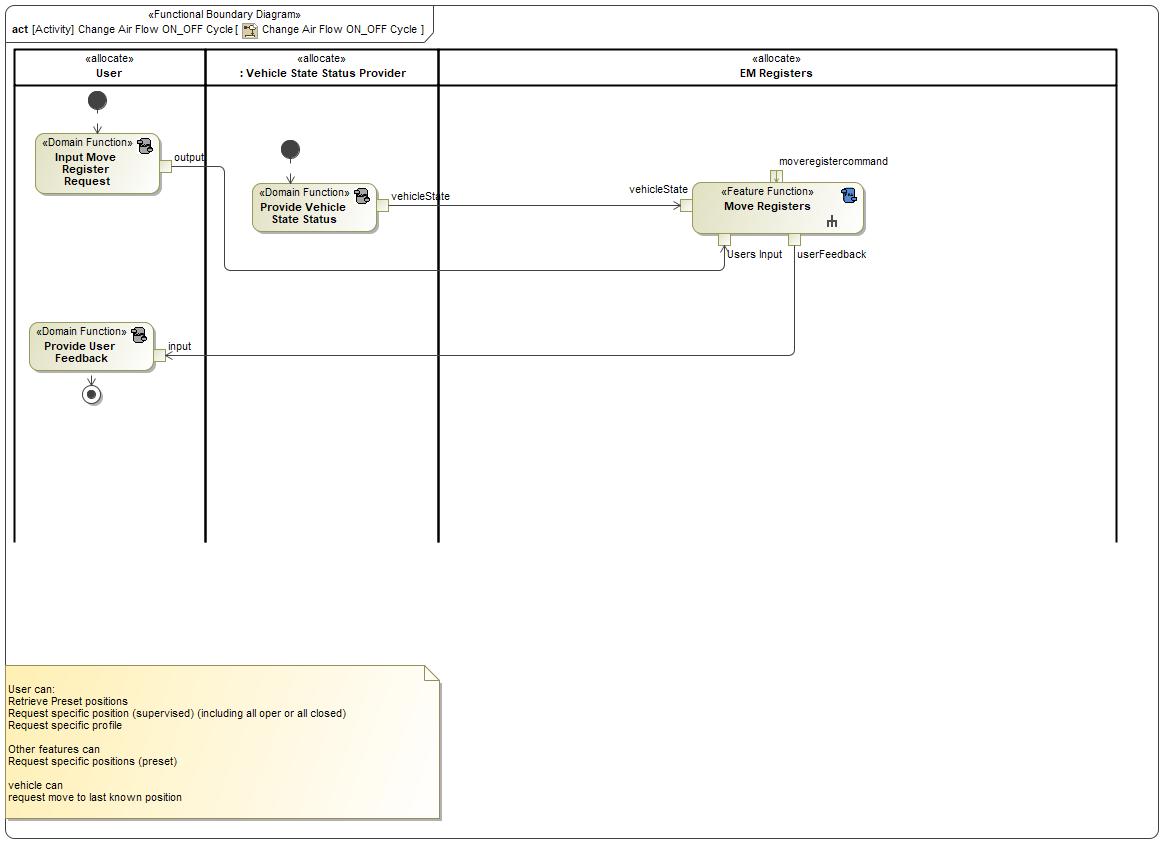


Figure 3‑1: Change Air Flow ON\_OFF Cycle

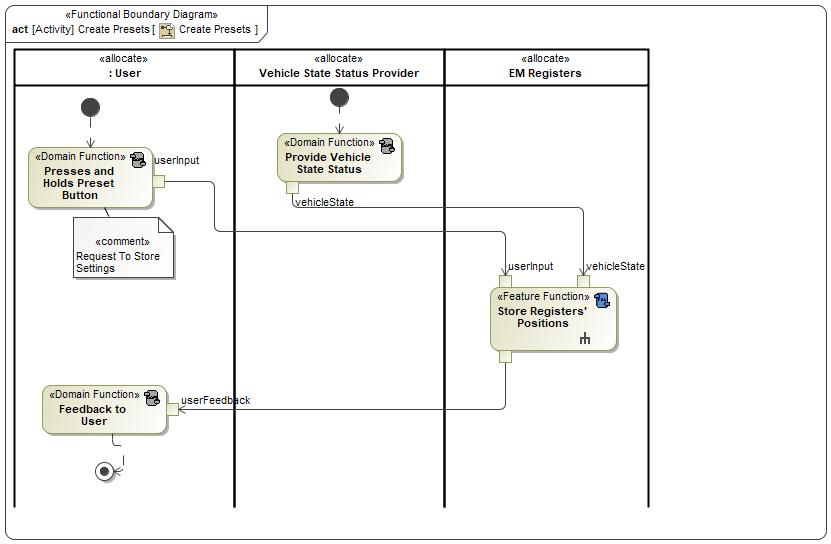


Figure 3‑1: Create Presets

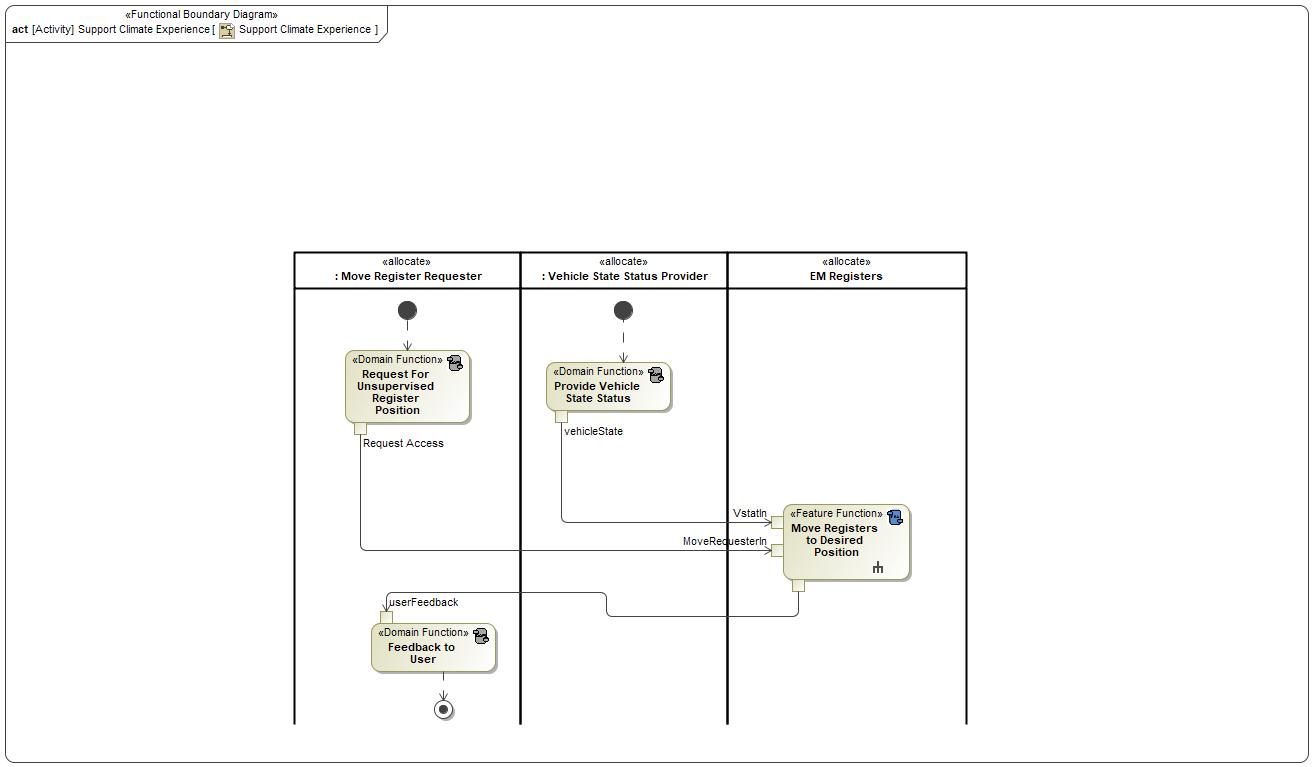


Figure 3‑1: Support Climate Experience

### 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:

### Signal List

|  |  |  |
| --- | --- | --- |
| **Signal Name** | **Description** | **Details** |
| **Actuate Register Command** | Signal requesting actuation of register position changes | Satisfies:  *No reqs. satisfied* |
| **EM\_DesiredPositionsRequest** | Manual move register requests | Satisfies:  *No reqs. satisfied* |
| **RegisterPositionHMIFeedback** | Register sensor position feedback to HMI | Satisfies:  *No reqs. satisfied* |
| **RegisterPositionsSensorFeedback** | Signal to provide register position feedback to actuator | Satisfies:  *No reqs. satisfied* |
| **vehicleStatus** | Vehicle on off status used by register move commands | Satisfies:  *No reqs. satisfied* |

Table 3‑3: List of Logical Signals

## Physical Architecture

### E/E Architecture

#### E/E Architecture Variants

*No E/E Architecture Variant found.*

##### E/E Architecture “Architecture Variant”: EMR Impementation Activity Diagram

This E/E Architecture variant … <add some explanatory text here>

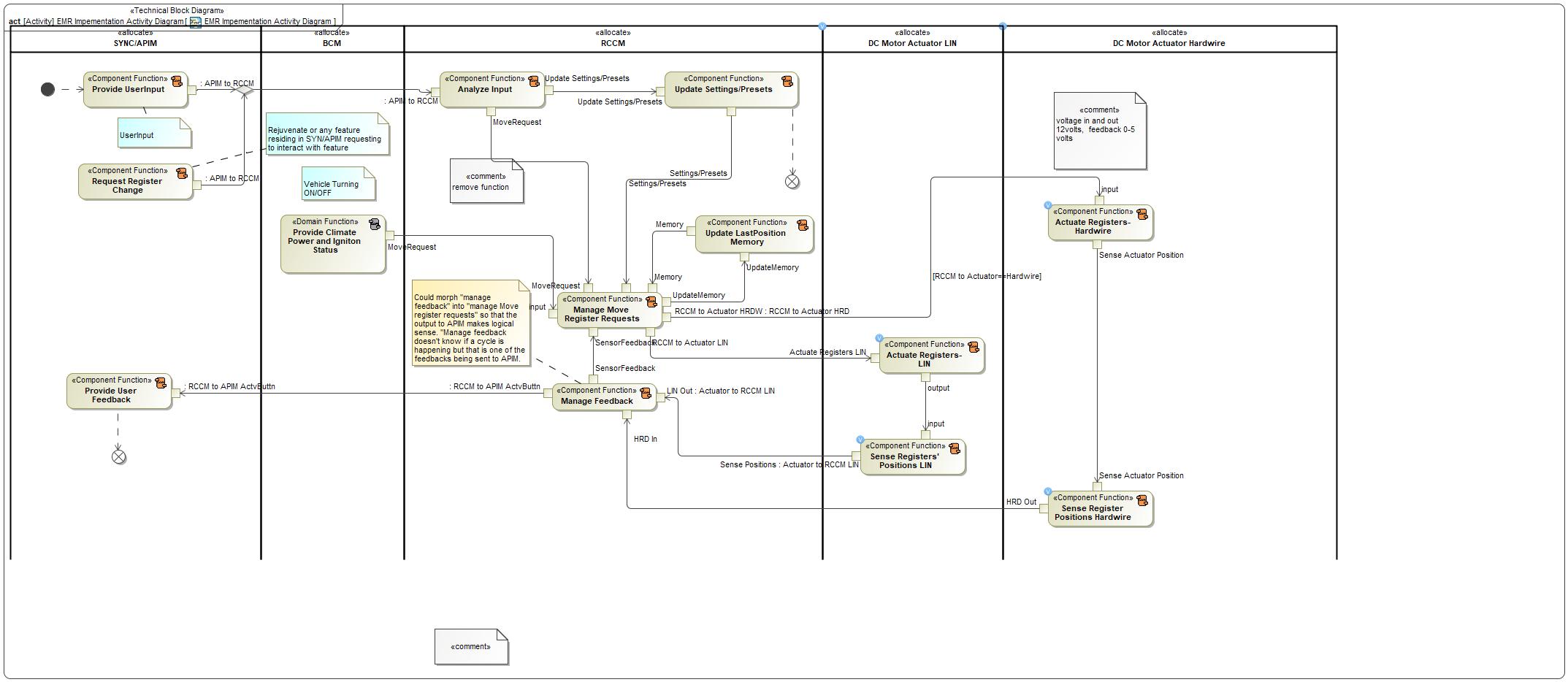


Figure 3‑2-1: EMR Impementation Activity Diagram

##### E/E Architecture “Architecture Variant”: Physical Architecture

This E/E Architecture variant … <add some explanatory text here>

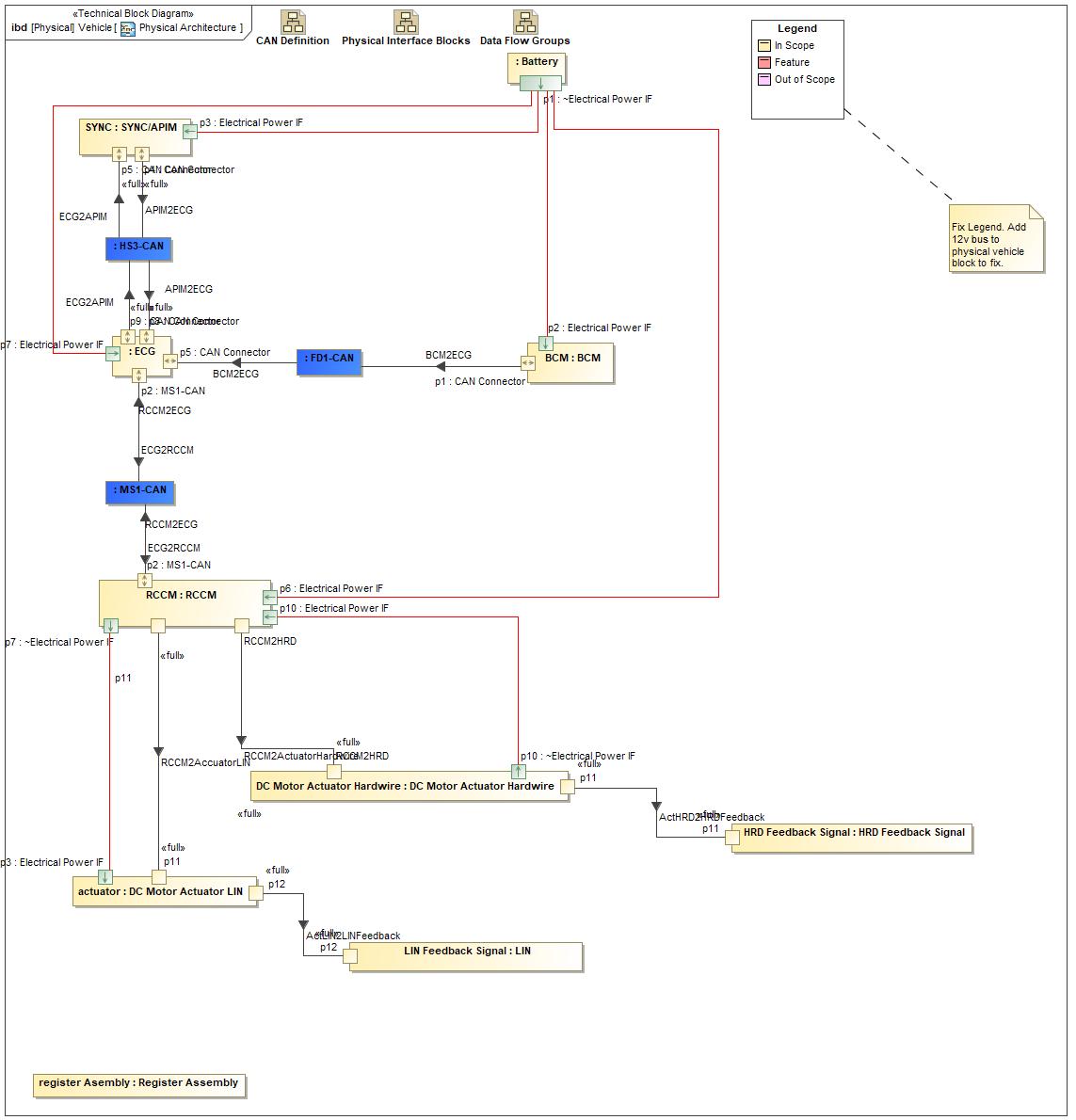


Figure 3‑2-2: Physical Architecture

##### E/E Architecture “Architecture Variant”: Technology Signals

This E/E Architecture variant … <add some explanatory text here>

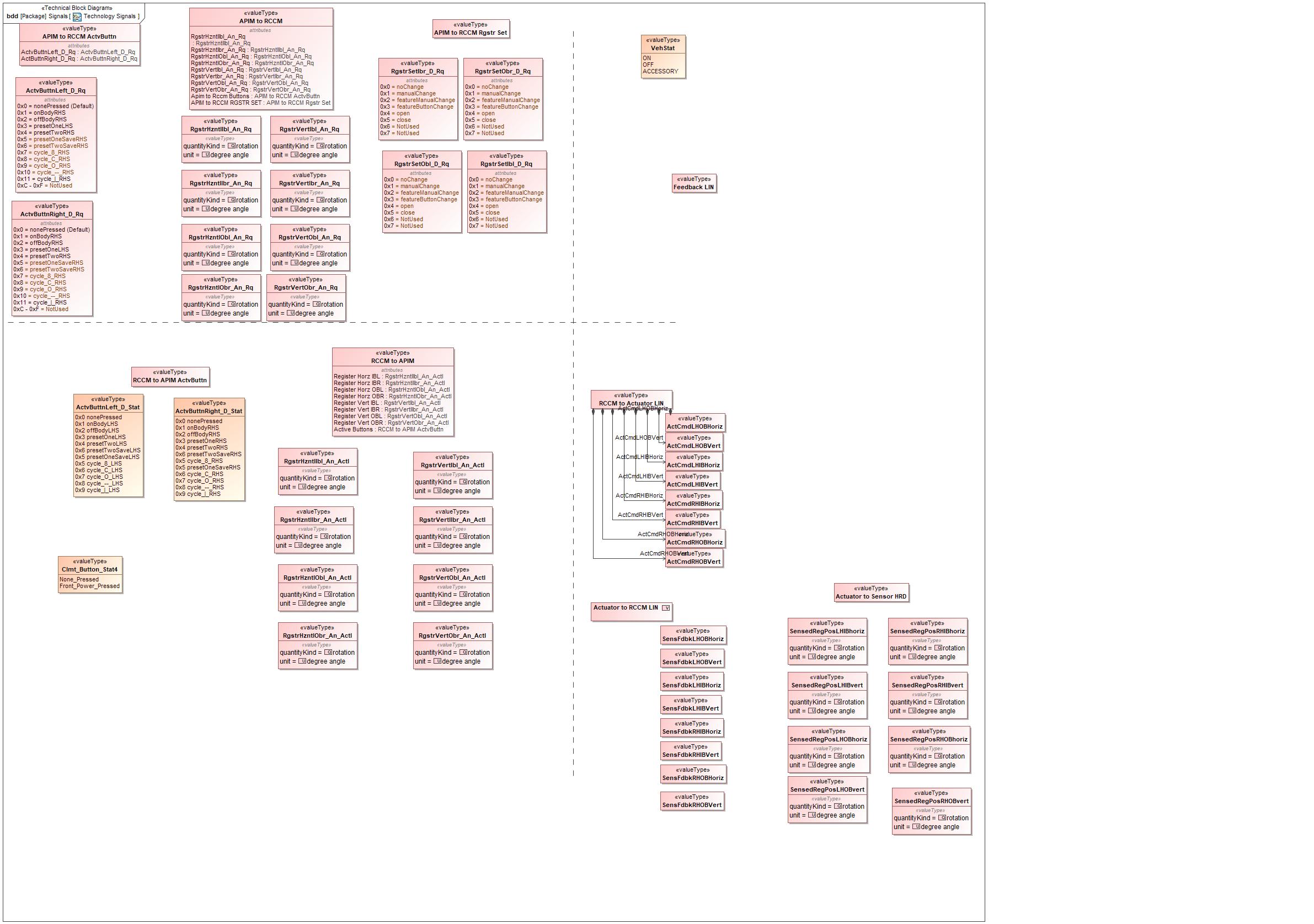


Figure 3‑2-3: Technology Signals

##### E/E Architecture “Architecture Variant”: Logical to Physical Signal Mapping

This E/E Architecture variant … <add some explanatory text here>

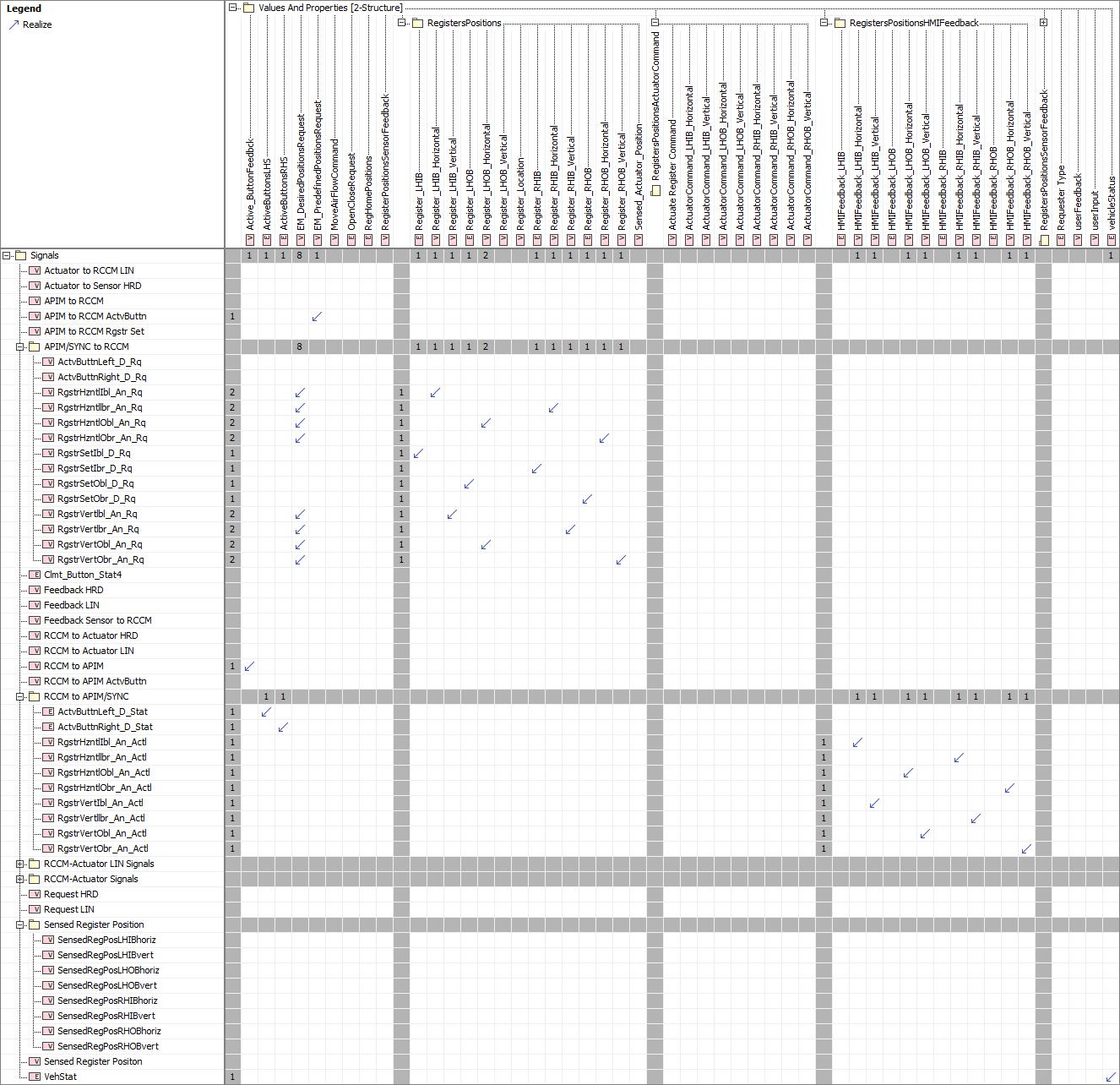


Figure 3‑2-4: Logical to Physical Signal Mapping

#### E/E Components

|  |  |
| --- | --- |
| Component Name | **Description** |

Table 3‑6: Electrical Components

#### E/E Connections

*No E/E Connections found.*

#### Signal List

|  |  |  |
| --- | --- | --- |
| **Signal Name** | **Description** | **Details** |
| **SensFdbkLHIBHoriz** | Sense left hand horizontal inboard register vane positions and forward to RCCM | Satisfies:  *No reqs. satisfied* |
| **RgstrVertllbr\_An\_Actl** | Aim Status of Righthand Inboard Register Vertical Vanes | Satisfies:  *No reqs. satisfied* |
| **RgstrSetObr\_D\_Rq** | Change Aim Command for Righthand Intboard Register | Satisfies:  *No reqs. satisfied* |
| **Request LIN** | Place holder for LIN signals 707 program | Satisfies:  *No reqs. satisfied* |
| **SensFdbkRHIBHoriz** | Sense right hand horizontal inboard register vane positions and forward to RCCM | Satisfies:  *No reqs. satisfied* |
| **APIM to RCCM Rgstr Set** |  | Satisfies:  *No reqs. satisfied* |
| **ActCmdLHIBHoriz** | Signal from RCCM to left hand inboard horizontal vane register actuator | Satisfies:  *No reqs. satisfied* |
| **ActvButtnLeft\_D\_Stat** | Subfeature Selection Status for Left Side Registers:  \*NonePressed  \*OnBodyLHS  \*OffBodyLHS  \*PresetOneLHS  \*PresetTwoLHS  \*Cycle\_8\_LHS  \*Cycle\_C\_LHS  \*Cycle\_O\_LHS  \*Cycle\_--\_LHS  \*Cycle\_I\_LHS | Satisfies:  *No reqs. satisfied* |
| **Feedback LIN** | Place holder for LIN feedback signals 707 program | Satisfies:  *No reqs. satisfied* |
| **RgstrVertlbr\_An\_Rq** | Command Position of Vertical Vanes for Righthand Inboard Register | Satisfies:  *No reqs. satisfied* |
| **RCCM to Actuator HRD** |  | Satisfies:  *No reqs. satisfied* |
| **RgstrHzntlIbl\_An\_Actl** | Aim Status of Lefthand Inboard Register Horizontal Vanes | Satisfies:  *No reqs. satisfied* |
| **Actuator to Sensor HRD** |  | Satisfies:  *No reqs. satisfied* |
| **SensFdbkLHOBHoriz** | Sense left hand horizontal outboard register vane positions and forward to RCCM | Satisfies:  *No reqs. satisfied* |
| **ActCmdLHOBVert** | Signal from RCCM to left hand outboard vertical vane register actuator | Satisfies:  *No reqs. satisfied* |
| **ActCmdRHOBVert** | Signal from RCCM to right hand outboard vertical vane register actuator | Satisfies:  *No reqs. satisfied* |
| **SensedRegPosRHOBhoriz** |  | Satisfies:  *No reqs. satisfied* |
| **RgstrVertObl\_An\_Actl** | Aim Status of Righthand Inboard Register Vertical Vanes | Satisfies:  *No reqs. satisfied* |
| **ActCmdRHOBHoriz** | Signal from RCCM to right hand outboard horizontal vane register actuator | Satisfies:  *No reqs. satisfied* |
| **RgstrHzntlObr\_An\_Actl** | Aim Status of Righthand Outboard Register Horizontal Vanes | Satisfies:  *No reqs. satisfied* |
| **RgstrHzntlObl\_An\_Rq** | Command Position of Horizontal Vanes for Righthand Outboard Register | Satisfies:  *No reqs. satisfied* |
| **RgstrVertObr\_An\_Actl** | Aim Status of Righthand Outboard Register Vertical Vanes | Satisfies:  *No reqs. satisfied* |
| **SensedRegPosLHOBhoriz** |  | Satisfies:  *No reqs. satisfied* |
| **RgstrHzntllbr\_An\_Actl** | Aim Status of Righthand Inboard Register Horizontal Vanes | Satisfies:  *No reqs. satisfied* |
| **Feedback Sensor to RCCM** |  | Satisfies:  *No reqs. satisfied* |
| **RgstrHzntllbr\_An\_Rq** | Command Position of Horizontal Vanes for Righthand Inboard Register | Satisfies:  *No reqs. satisfied* |
| **SensedRegPosRHOBvert** |  | Satisfies:  *No reqs. satisfied* |
| **ActCmdLHOBHorizLin** | Place holder for LIN signals | Satisfies:  *No reqs. satisfied* |
| **SensFdbkLHOBVert** | Sense left hand vertical outboard register vane positions and forward to RCCM | Satisfies:  *No reqs. satisfied* |
| **SensedRegPosLHIBhoriz** |  | Satisfies:  *No reqs. satisfied* |
| **ActCmdRHIBVert** | Signal from RCCM to right hand inboard vertical vane register actuator | Satisfies:  *No reqs. satisfied* |
| **ActCmdLHIBVert** | Signal from RCCM to left hand inboard vertica vane register actuator | Satisfies:  *No reqs. satisfied* |
| **ActCmdLHIBHorizLin** | Place holder for LIN signals | Satisfies:  *No reqs. satisfied* |
| **VehStat** | Vehicle status signals used to store register positions when vehicle is on | Satisfies:  *No reqs. satisfied* |
| **RgstrVertObl\_An\_Rq** | Command Position of Vertical Vanes for Lefthand Outboard Register | Satisfies:  *No reqs. satisfied* |
| **RgstrVertObr\_An\_Rq** | Command Position of Vertical Vanes for Righthand Outboard Register | Satisfies:  *No reqs. satisfied* |
| **SensedRegPosLHIBvert** |  | Satisfies:  *No reqs. satisfied* |
| **SensFdbkLHIBVert** | Sense left hand vertical inboard register vane positions and forward to RCCM | Satisfies:  *No reqs. satisfied* |
| **Sensed Register Positon** |  | Satisfies:  *No reqs. satisfied* |
| **SensFdbkRHOBVert** | Sense right hand vertical inboard register vane positions and forward to RCCM | Satisfies:  *No reqs. satisfied* |
| **Feedback HRD** | Place holder for hardwire feedback from potentiometer to actuator | Satisfies:  *No reqs. satisfied* |
| **RgstrVertIbl\_An\_Actl** | Aim Status of Lefthand Inboard Register Vertical Vanes | Satisfies:  *No reqs. satisfied* |
| **RgstrSetObl\_D\_Rq** | Change Aim Command for Lefthand Outboard Register | Satisfies:  *No reqs. satisfied* |
| **SensedRegPosLHOBvert** |  | Satisfies:  *No reqs. satisfied* |
| **SensedRegPosRHIBhoriz** |  | Satisfies:  *No reqs. satisfied* |
| **ActCmdRHIBHoriz** | Signal from RCCM to right hand inboard horizontal vane register actuator | Satisfies:  *No reqs. satisfied* |
| **ActvButtnLeft\_D\_Rq** | Subfeature Selection Commands for Left Side Registers | Satisfies:  *No reqs. satisfied* |
| **SensFdbkRHOBHoriz** | Sense right hand horizontal outboard register vane positions and forward to RCCM | Satisfies:  *No reqs. satisfied* |
| **RgstrSetIbl\_D\_Rq** | Change Aim Command for Lefthand Inboard Register | Satisfies:  *No reqs. satisfied* |
| **Request HRD** | Place holder for hardwire signals | Satisfies:  *No reqs. satisfied* |
| **RgstrHzntlObl\_An\_Actl** | Aim Status of Lefthand Outboard Register Horizontal Vanes | Satisfies:  *No reqs. satisfied* |
| **Clmt\_Button\_Stat4** | Climate button signal | Satisfies:  *No reqs. satisfied* |
| **ActvButtnRight\_D\_Rq** | Subfeature Selection Commands for Right Side Registers | Satisfies:  *No reqs. satisfied* |
| **APIM to RCCM ActvButtn** |  | Satisfies:  *No reqs. satisfied* |
| **RgstrHzntlObr\_An\_Rq** | Command Position of Horizontal Vanes for Righthand Outboard Register | Satisfies:  *No reqs. satisfied* |
| **RCCM to APIM** |  | Satisfies:  *No reqs. satisfied* |
| **SensedRegPosRHIBvert** |  | Satisfies:  *No reqs. satisfied* |
| **RgstrVertlbl\_An\_Rq** | Command Position of Vertical Vanes for Lefthand Inboard Register | Satisfies:  *No reqs. satisfied* |
| **ActvButtnRight\_D\_Stat** | Subfeature Selection Status for Right Side Registers:  \*NonePressed  \*OnBodyLHS  \*OffBodyLHS  \*PresetOneLHS  \*PresetTwoLHS  \*Cycle\_8\_LHS  \*Cycle\_C\_LHS  \*Cycle\_O\_LHS  \*Cycle\_--\_LHS  \*Cycle\_I\_LHS | Satisfies:  *No reqs. satisfied* |
| **SensFdbkRHIBVert** | Sense right hand vertical inboard register vane positions and forward to RCCM | Satisfies:  *No reqs. satisfied* |
| **ActCmdLHIBVertLin** | Place holder for LIN signals | Satisfies:  *No reqs. satisfied* |
| **RgstrHzntlIbl\_An\_Rq** | Command Position of Horizontal Vanes for Lefthand Inboard Register | Satisfies:  *No reqs. satisfied* |
| **RgstrSetIbr\_D\_Rq** | Change Aim Command for Righthand Intboard Register | Satisfies:  *No reqs. satisfied* |
| **ActCmdLHOBHoriz** | Signal from RCCM to left hand inboard horizontal vane register actuator | Satisfies:  *No reqs. satisfied* |

Table 3‑8: List of Technical Signals

### Software Component Architecture

*Not supported by MagicDraw report generation.*

#### Description

*Not supported by MagicDraw report generation.*

## Function Deployment

### Deployment Variants

*Not supported by MagicDraw report generation.*

#### Deployment “Variant 1”

This deployment variant … <add some explanatory text here>

No description specified.

### Function Allocation

| Component | Technology Function Name | Logical Function Name |
| --- | --- | --- |
|
| SYNC/APIM | Provide UserInput | *No logical function allocated* |
| Provide User Feedback | *No logical function allocated* |
| Request Register Change | *No logical function allocated* |
| Provide Climate System ON/OFF Selection | *No logical function allocated* |
|  |
| BCM |  |
| RCCM | Manage Move Register Requests | *No logical function allocated* |
| Manage Feedback | *No logical function allocated* |
| Analyze Input | *No logical function allocated* |
| Update Settings/Presets | *No logical function allocated* |
| Update LastPosition Memory | *No logical function allocated* |
| Provide Climate System ON/OFF Selection | *No logical function allocated* |
| Provide IgnitionStatus | *No logical function allocated* |
|  |

Table 3‑9: Function Allocation Table (Basic)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Component | | Technology Function Name | TSR | |
| Name | ASIL |  | ID | ASIL |
| SYNC/APIM |  | Provide UserInput | * Provide User Input * Coordinate System for Each Register * Drag Dots for HMI * Air Flow Symbol for Axial FLow from Register * Air Flow Lines for HMI * Drag Dot Coordinate Units * Provide Climate Power and Ignition Status * RCCM to APIM Feedback Response to Vane Closure Request * APIM Vane Closure Value * Register Stuck Feedback behavior. * Registers Active in Manual Mode * SYN Range of Coverage Coordinate Box * SYNC Behavior During Air Motion * SYNC CLOSED Register Symbol Behavior * SYNC Drag Dot Behavior for Air Motion * SYNC Process for CLOSING Register * SYNC Process for Opening register * SYNC Screen Behavior Mode Man Panel Vents OFF IN Manual Mode * Time Required to Change Commands * Units of Register Postion Signal |  |
|  | Provide User Feedback | * Provide User Feedback * SYNC Memory for drag dots * SYNC Screen Behavior Climate System ON/OFF * SYNC Screen Behavior Vehicle ON/OFF * Sense Registers' Positions * Closed vent fuction |  |
|  | Request Register Change | * Request Air Flow Change * Rejuvenate Control of Registers |  |
|  | Provide Climate System ON/OFF Selection | * SYNC Screen Behavior Climate System ON/OFF |  |
|  |  |  |
| BCM |  |  |  |
| RCCM |  | Manage Move Register Requests | * Manage Move Register Requests * Memorize Last Register Static Position * Default Value for Memorizable Presets * Fixed Preset Target Values * Opening Behavior * RCCM to APIM Feedback Response to Vane Closure Request * Registers Active in Manual Mode * Rejuvenate User-Forced Cancellation * SYNC Provide Drag Dots to Last Static Position * SYNC provides Drag Dots to Fixed and Memorized Preset Positions * SYNC Screen Behavior when Climate System is in EATC Mode * Values for Air Motion * Values for Rejuvenate Air Motion * Fixed Preset Target Values Allternative * Fixed Preset Target Values Alternative Part 2 |  |
|  | Manage Feedback | * Manage Feedback * SYNC Drag Dot Behavior for Closing Register * SYNC Drag Dot Behavior for Opening Register |  |
|  | Analyze Input | * Analyse & Arbitrate Input * Coordinate System for Each Register * Default Value for Last Static Position * Fixed Preset Target Values * Fixed Preset Target Values Allternative * Fixed Preset Target Values Alternative Part 2 |  |
|  | Update Settings/Presets | * SYNC Memory for drag dots * Preset Default State * SYNC Behavior for Memorizable Presets * SYNC & RCCM Memorizing for Memorizable Presets |  |
|  | Update LastPosition Memory | * Update LastPosition Memory * Closed Register Memorization |  |
|  | Provide Climate System ON/OFF Selection | * SYNC Screen Behavior Climate System ON/OFF |  |
|  | Provide IgnitionStatus | *No reqs. satisfied by tech. fx.* |  |
|  |  |  |

Table 3‑10: Function Allocation Table (Functional Safety Extension)

# Feature Implementation Modeling

## Component Interaction Diagrams

### Scenario: “System Startup / Shutdown”

### Scenario: “Normal Operation”

No “Feature Scenario Diagram” found.

## Component Interface Behavior Diagrams

*Not supported by MagicDraw report generation.*

*See Appendix Section 8.1.9 for Technology State Machines*

# Feature Implementation Requirements

## Functional Safety

### ASIL Decomposition of Technical Safety Requirements

<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

### SYNC/APIM

SYNC/APIM

#### Technology Function 1806768575.jpg **Provide UserInput**

Implementation function that processes user input to the SYNC and forwards it to the RCCM

##### Function Interfaces

###### Inputs

(No inputs have been defined)

###### Outputs

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Outputs: Provide UserInput | | | | | | |
| **Logical Signal Name** | **Technical Signal Name** | | | **Mapping Details** *(Conditional)* | **Publisher Interface** | **Connection**  (*Optional)* |
| Review in model  User Input | User Input :  APIM to RCCM | | |  |  |  |
| [Req. Template](http://wiki.ford.com/display/RequirementsEngineering/Requirements+Attributes) Version | | 6.1 | End of Outputs | | | |

Table 5‑3: Output Signal mappings of Function Provide UserInput

###### 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 5‑2: Parameter mappings of Function “MyLogicalFunctionA\_Component1”

###### Interface Requirements

##### Function Requirements

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **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 5‑3: Component Specific Requirements

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

Table 5‑4: Inherited Requirements

###### Component Specific Requirements

SYN Range of Coverage Coordinate Box

Satisfied by:

* Functions:
  + Provide UserInput

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Requirement ID: | | | | | | | |
| **Rationale** |  | | | | | | |
| **Acceptance Criteria** |  | | | | | | |
| **Notes** |  | | | | | | |
| **Source** |  | | | | | **Owner** |  |
| **Source Req.** | * 744999135.jpg Detect User Input | | | | | **V&V Method** |  |
| **Type** |  | | | **Priority** |  | **Status** | In-Progress |
| [Req. Template](http://wiki.ford.com/display/RequirementsEngineering/Requirements+Attributes) Version | | 6.0 | End of Requirement | | | | |

Provide User Input

When user Interacts with HMI Screen to move Register Static Aiming, SYNC shall send out "0x1=manualChange" for the entire duration of time user is touching the screen.

Satisfied by:

* Functions:
  + Provide UserInput

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Requirement ID: | | | | | | | |
| **Rationale** |  | | | | | | |
| **Acceptance Criteria** | view register movement based on user desired input | | | | | | |
| **Notes** | Provides user register move requests to the RCCM | | | | | | |
| **Source** |  | | | | | **Owner** | George Smith |
| **Source Req.** | * 744999135.jpg Provide User Input * 744999135.jpg Detect User Input | | | | | **V&V Method** |  |
| **Type** |  | | | **Priority** | 2 - Medium | **Status** | Ready for Review |
| [Req. Template](http://wiki.ford.com/display/RequirementsEngineering/Requirements+Attributes) Version | | 6.0 | End of Requirement | | | | |

SYNC Process for Opening register

User shall single -tap Drag Dot to open airflow of any individual closed register. A single tap shall be defined as a press and hold < X milliseconds.

Satisfied by:

* Functions:
  + Provide UserInput

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Requirement ID: | | | | | | | |
| **Rationale** |  | | | | | | |
| **Acceptance Criteria** |  | | | | | | |
| **Notes** |  | | | | | | |
| **Source** |  | | | | | **Owner** |  |
| **Source Req.** | * 744999135.jpg Detect User Input | | | | | **V&V Method** |  |
| **Type** |  | | | **Priority** |  | **Status** | In-Progress |
| [Req. Template](http://wiki.ford.com/display/RequirementsEngineering/Requirements+Attributes) Version | | 6.0 | End of Requirement | | | | |

Drag Dots for HMI

SYNC shall provide a visible Drag Dot that the user can touch and drag around the HMI Screen to adjust aiming of the register.

Satisfied by:

* Functions:
  + Provide UserInput

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Requirement ID: | | | | | | | |
| **Rationale** |  | | | | | | |
| **Acceptance Criteria** | Observation of Drag Dot on SYNC screen | | | | | | |
| **Notes** |  | | | | | | |
| **Source** |  | | | | | **Owner** | George Smith |
| **Source Req.** | * 744999135.jpg Detect User Input | | | | | **V&V Method** |  |
| **Type** |  | | | **Priority** | 2 - Medium | **Status** | In-Progress |
| [Req. Template](http://wiki.ford.com/display/RequirementsEngineering/Requirements+Attributes) Version | | 6.0 | End of Requirement | | | | |

Coordinate System for Each Register

SYNC shall provide a non visible "range of coverage" box around each register on the HMI screen. This box provides the area of coverage for each register relative to the Cockpit as depicted by the Wireframe image. SYNC shall provide coordiate tracking of the Drag Dots within these zones in (X,Y) format, where X is the cross car componet of the Drag Dot and Y is the up down componet of the Drag Dot. The origin (0,0) will be in the lower left corner of the box. The maximum values shall be (150, 90) in the upper right corner of the box. HMI shall return a linearly proportional position of the Drag Dot anywhere within the box that it is positioned. The nomimal units of the Drag Dot shall be degrees which coincide with the Register Vane roations required to proivde air to the relative position as depicted on the Cockpit HMI wireframe image.

Satisfied by:

* Functions:
  + Analyze Input
  + Provide UserInput

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Requirement ID: | | | | | | | |
| **Rationale** |  | | | | | | |
| **Acceptance Criteria** | Observation of vane movement when user changes register position on SYNC screen | | | | | | |
| **Notes** |  | | | | | | |
| **Source** |  | | | | | **Owner** | George Smith |
| **Source Req.** | * 744999135.jpg Move to Requested Position | | | | | **V&V Method** |  |
| **Type** |  | | | **Priority** | 2 - Medium | **Status** | In-Progress |
| [Req. Template](http://wiki.ford.com/display/RequirementsEngineering/Requirements+Attributes) Version | | 6.0 | End of Requirement | | | | |

SYNC Process for CLOSING Register

User shall double -tap Drag Dot to close airflow of any individual register. A double tap shall be defined as two taps x milliseconds apart.

Satisfied by:

* Functions:
  + Provide UserInput

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Requirement ID: | | | | | | | |
| **Rationale** |  | | | | | | |
| **Acceptance Criteria** |  | | | | | | |
| **Notes** |  | | | | | | |
| **Source** |  | | | | | **Owner** |  |
| **Source Req.** | * 744999135.jpg Detect Register Feedback Positions | | | | | **V&V Method** |  |
| **Type** |  | | | **Priority** |  | **Status** | In-Progress |
| [Req. Template](http://wiki.ford.com/display/RequirementsEngineering/Requirements+Attributes) Version | | 6.0 | End of Requirement | | | | |

Drag Dot Coordinate Units

The RCCM nominal units of the Drag Dot shall be degrees which coincide with the Register Vane rotations required to provide air to the relative position as depicted on the Cockpit HMI wireframe image.

Satisfied by:

* Functions:
  + Provide UserInput

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Requirement ID: | | | | | | | |
| **Rationale** |  | | | | | | |
| **Acceptance Criteria** | Obsersvation of SYNC screen | | | | | | |
| **Notes** |  | | | | | | |
| **Source** |  | | | | | **Owner** | George Smith |
| **Source Req.** | * 744999135.jpg Detect User Input | | | | | **V&V Method** |  |
| **Type** |  | | | **Priority** | 2 - Medium | **Status** | In-Progress |
| [Req. Template](http://wiki.ford.com/display/RequirementsEngineering/Requirements+Attributes) Version | | 6.0 | End of Requirement | | | | |

SYNC CLOSED Register Symbol Behavior

SYNC shall remove AIR Flow lines and provide a CLOSED Register symbol superimposed upon the Register when RCCM provides confirmation that the Register is closed.

Satisfied by:

* Functions:
  + Provide UserInput

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Requirement ID: | | | | | | | |
| **Rationale** |  | | | | | | |
| **Acceptance Criteria** |  | | | | | | |
| **Notes** |  | | | | | | |
| **Source** |  | | | | | **Owner** |  |
| **Source Req.** | * 744999135.jpg Detect User Input | | | | | **V&V Method** |  |
| **Type** |  | | | **Priority** |  | **Status** | In-Progress |
| [Req. Template](http://wiki.ford.com/display/RequirementsEngineering/Requirements+Attributes) Version | | 6.0 | End of Requirement | | | | |

Register Stuck Feedback behavior.

If Register Feedback is unchanged for >120 seconds after RCCM has issued move command, move command will be terminated

Satisfied by:

* Functions:
  + Provide UserInput

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Requirement ID: | | | | | | | |
| **Rationale** |  | | | | | | |
| **Acceptance Criteria** | Observation of EM Register movement | | | | | | |
| **Notes** |  | | | | | | |
| **Source** |  | | | | | **Owner** | George Smith |
| **Source Req.** | * 744999135.jpg Register Controller Functionality | | | | | **V&V Method** |  |
| **Type** |  | | | **Priority** | 2 - Medium | **Status** | In-Progress |
| [Req. Template](http://wiki.ford.com/display/RequirementsEngineering/Requirements+Attributes) Version | | 6.0 | End of Requirement | | | | |

Air Flow Lines for HMI

SYNC shall provide Air Flow Visualization lines emenating from the registers and in the direction of Air Flow once that is provided by RCCM. Air Flow Lines shall be shown continuously and only change if direction of air flow is changed or closed or if system is turned off. If AIr Motion is active, Sync will continuously move Air Flow lines as RCCM relays incremental status of moving Air Flow

Satisfied by:

* Functions:
  + Provide UserInput

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Requirement ID: | | | | | | | |
| **Rationale** |  | | | | | | |
| **Acceptance Criteria** | Observation of SYNC screen | | | | | | |
| **Notes** |  | | | | | | |
| **Source** |  | | | | | **Owner** | George Smith |
| **Source Req.** | * 744999135.jpg Detect User Input | | | | | **V&V Method** |  |
| **Type** |  | | | **Priority** | 2 - Medium | **Status** | In-Progress |
| [Req. Template](http://wiki.ford.com/display/RequirementsEngineering/Requirements+Attributes) Version | | 6.0 | End of Requirement | | | | |

RCCM to APIM Feedback Response to Vane Closure Request

When "Manage Move Requests" component function receives a command to close a register, it shall provide feedback to APIM by setting max value=255 for the Rear/Vertical Vanes angle in the following signals: RgstrVertObl\_An\_Actl=255, RgstrVertObr\_An\_Actl=255, RgstrVertIbr\_An\_Actl=255 , and RgstrVertIbl\_An\_Actl=255.

Satisfied by:

* Functions:
  + Actuate Registers-Hardwire
  + Manage Move Register Requests
  + Provide UserInput

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Requirement ID: | | | | | | | |
| **Rationale** |  | | | | | | |
| **Acceptance Criteria** | Test to detect value 255 when register receives a close command | | | | | | |
| **Notes** |  | | | | | | |
| **Source** |  | | | | | **Owner** | George Smith |
| **Source Req.** | * 744999135.jpg Detect Register Feedback Positions | | | | | **V&V Method** |  |
| **Type** |  | | | **Priority** | 2 - Medium | **Status** | In-Progress |
| [Req. Template](http://wiki.ford.com/display/RequirementsEngineering/Requirements+Attributes) Version | | 6.0 | End of Requirement | | | | |

SYNC Screen Behavior Mode Man Panel Vents OFF IN Manual Mode

If Climate system is in manual mode and the Mode Man Panel Vents are set to off then the display shall not show air flow lines nor Drag dots

Satisfied by:

* Functions:
  + Provide UserInput

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Requirement ID: | | | | | | | |
| **Rationale** |  | | | | | | |
| **Acceptance Criteria** |  | | | | | | |
| **Notes** |  | | | | | | |
| **Source** |  | | | | | **Owner** |  |
| **Source Req.** | * 744999135.jpg Detect User Input | | | | | **V&V Method** |  |
| **Type** |  | | | **Priority** |  | **Status** | In-Progress |
| [Req. Template](http://wiki.ford.com/display/RequirementsEngineering/Requirements+Attributes) Version | | 6.0 | End of Requirement | | | | |

Time Required to Change Commands

Time for the system to respond (Registers start movement) to button pushed or movement of the drag dot shall not be more than 150 Milli-seconds. Time for Display to update from Register movement shall not be more 250 Milli-seconds af??????????

Satisfied by:

* Functions:
  + Provide UserInput

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Requirement ID: | | | | | | | |
| **Rationale** |  | | | | | | |
| **Acceptance Criteria** | Observation of EM Register movement | | | | | | |
| **Notes** |  | | | | | | |
| **Source** |  | | | | | **Owner** | George Smith |
| **Source Req.** | * 744999135.jpg Register Controller Functionality | | | | | **V&V Method** |  |
| **Type** |  | | | **Priority** | 2 - Medium | **Status** | In-Progress |
| [Req. Template](http://wiki.ford.com/display/RequirementsEngineering/Requirements+Attributes) Version | | 6.0 | End of Requirement | | | | |

SYNC Drag Dot Behavior for Air Motion

SYNC shall remove Drag Dot when Air Motion intiates and during cycling.

Satisfied by:

* Functions:
  + Provide UserInput

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Requirement ID: | | | | | | | |
| **Rationale** |  | | | | | | |
| **Acceptance Criteria** |  | | | | | | |
| **Notes** |  | | | | | | |
| **Source** |  | | | | | **Owner** |  |
| **Source Req.** | * 744999135.jpg Detect User Input | | | | | **V&V Method** |  |
| **Type** |  | | | **Priority** |  | **Status** | In-Progress |
| [Req. Template](http://wiki.ford.com/display/RequirementsEngineering/Requirements+Attributes) Version | | 6.0 | End of Requirement | | | | |

Units of Register Postion Signal

The nomimal units of the Register Command Position Signal (Drag Dot position) shall be degrees which coincide with the Register Vane roations required to proivde air to the relative position as depicted on the Cockpit HMI wireframe image.

Satisfied by:

* Functions:
  + Provide UserInput

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Requirement ID: | | | | | | | |
| **Rationale** |  | | | | | | |
| **Acceptance Criteria** | Observation of X,Y provided table | | | | | | |
| **Notes** |  | | | | | | |
| **Source** |  | | | | | **Owner** | George Smith |
| **Source Req.** | * 744999135.jpg Register Controller Functionality | | | | | **V&V Method** |  |
| **Type** |  | | | **Priority** | 2 - Medium | **Status** | In-Progress |
| [Req. Template](http://wiki.ford.com/display/RequirementsEngineering/Requirements+Attributes) Version | | 6.0 | End of Requirement | | | | |

Provide Climate Power and Ignition Status

When RCCM detects "Provide Climate Power and Ignition Status" component function, it shall provide a "vehicle state" and "climate state" when feature is activated

Satisfied by:

* Functions:
  + Provide UserInput

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Requirement ID: | | | | | | | |
| **Rationale** |  | | | | | | |
| **Acceptance Criteria** |  | | | | | | |
| **Notes** |  | | | | | | |
| **Source** |  | | | | | **Owner** | George Smith |
| **Source Req.** | * 744999135.jpg Register Controller Functionality | | | | | **V&V Method** |  |
| **Type** |  | | | **Priority** | 2 - Medium | **Status** | In-Progress |
| [Req. Template](http://wiki.ford.com/display/RequirementsEngineering/Requirements+Attributes) Version | | 6.0 | End of Requirement | | | | |

Registers Active in Manual Mode

If Climate system is in manual mode, Registers shall only show as active on HMI if PANEL VENTS is slected by itself or in any combination with the other manual mode buttons (FLOOR VENTS and DEFROSTER VENTS).

Satisfied by:

* Functions:
  + Manage Move Register Requests
  + Provide UserInput

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Requirement ID: | | | | | | | |
| **Rationale** |  | | | | | | |
| **Acceptance Criteria** | Observation of Registers active on HMI for the specified conditions | | | | | | |
| **Notes** |  | | | | | | |
| **Source** |  | | | | | **Owner** | George Smith |
| **Source Req.** | * 744999135.jpg Move to Requested Position | | | | | **V&V Method** |  |
| **Type** |  | | | **Priority** | 2 - Medium | **Status** | In-Progress |
| [Req. Template](http://wiki.ford.com/display/RequirementsEngineering/Requirements+Attributes) Version | | 6.0 | End of Requirement | | | | |

Air Flow Symbol for Axial FLow from Register

SYNC shall prosition a Drag Dot if airflow for register is postioned as normal to the plane of the screen when RCCM provides Air Flow coordinates on/near the center of the register.

Satisfied by:

* Functions:
  + Provide UserInput

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Requirement ID: | | | | | | | |
| **Rationale** |  | | | | | | |
| **Acceptance Criteria** | Observation of SYNC screen | | | | | | |
| **Notes** |  | | | | | | |
| **Source** |  | | | | | **Owner** | George Smith |
| **Source Req.** | * 744999135.jpg Detect User Input | | | | | **V&V Method** |  |
| **Type** |  | | | **Priority** | 2 - Medium | **Status** | In-Progress |
| [Req. Template](http://wiki.ford.com/display/RequirementsEngineering/Requirements+Attributes) Version | | 6.0 | End of Requirement | | | | |

SYNC Behavior During Air Motion

SYNC shall ignore any screen attempts to adjust Registers or Button Pushes during Air Motion except for Register closure or cancellation via Air Motion Button press.

Satisfied by:

* Functions:
  + Provide UserInput

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Requirement ID: | | | | | | | |
| **Rationale** |  | | | | | | |
| **Acceptance Criteria** |  | | | | | | |
| **Notes** |  | | | | | | |
| **Source** |  | | | | | **Owner** |  |
| **Source Req.** | * 744999135.jpg Detect User Input | | | | | **V&V Method** |  |
| **Type** |  | | | **Priority** |  | **Status** | In-Progress |
| [Req. Template](http://wiki.ford.com/display/RequirementsEngineering/Requirements+Attributes) Version | | 6.0 | End of Requirement | | | | |

APIM Vane Closure Value

The value of 255 for signals RgstrVertObl\_An\_Actl, RgstrVertObr\_An\_Actl, RgstrVertIbr\_An\_Actl , and RgstrVertIbl\_An\_Actl shall be used to signal to the APIM that the register is closed

Satisfied by:

* Functions:
  + Provide UserInput

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Requirement ID: | | | | | | | |
| **Rationale** |  | | | | | | |
| **Acceptance Criteria** | Observation that registers move to close position for value of 255 | | | | | | |
| **Notes** |  | | | | | | |
| **Source** |  | | | | | **Owner** | George Smith |
| **Source Req.** | * 744999135.jpg Move to Requested Position | | | | | **V&V Method** |  |
| **Type** |  | | | **Priority** | 2 - Medium | **Status** | In-Progress |
| [Req. Template](http://wiki.ford.com/display/RequirementsEngineering/Requirements+Attributes) Version | | 6.0 | End of Requirement | | | | |

#### Technology Function -682842197.jpg **Provide User Feedback**

Implementation function that provides the user feedback on register move requests

##### Function Interfaces

###### Inputs

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Inputs: Provide User Feedback | | | | | | |
| **Logical Signal Name** | **Technical Signal Name** | | | **Mapping Details** *(Conditional)* | **Subscriber Interface** | **Connection**  (*Optional)* |
| Review in model  Feedback | Feedback :  RCCM to APIM ActvButtn | | |  |  |  |
| [Req. Template](http://wiki.ford.com/display/RequirementsEngineering/Requirements+Attributes) Version | | 6.1 | End of Inputs | | | |

Table 5‑2: Input Signal mappings of Function Provide User Feedback

###### Outputs

(No outputs have been defined)

###### 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 5‑2: Parameter mappings of Function “MyLogicalFunctionA\_Component1”

###### Interface Requirements

##### Function Requirements

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **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 5‑3: Component Specific Requirements

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

Table 5‑4: Inherited Requirements

###### Component Specific Requirements

Provide User Feedback

When "Provide User Feedback" component function receives the signal "RCCM to APIM", it shall provide feedback to the user

Satisfied by:

* Functions:
  + Provide User Feedback

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Requirement ID: | | | | | | | |
| **Rationale** |  | | | | | | |
| **Acceptance Criteria** | Observation of feedback on SYNC Screen | | | | | | |
| **Notes** |  | | | | | | |
| **Source** |  | | | | | **Owner** | George Smith |
| **Source Req.** | * 744999135.jpg Detect Register Feedback Positions | | | | | **V&V Method** |  |
| **Type** |  | | | **Priority** | 2 - Medium | **Status** | Ready for Review |
| [Req. Template](http://wiki.ford.com/display/RequirementsEngineering/Requirements+Attributes) Version | | 6.0 | End of Requirement | | | | |

SYNC Screen Behavior Vehicle ON/OFF

When the "ignition=off", the HMI display shall disable move register command selections.

Satisfied by:

* Functions:
  + Provide User Feedback

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Requirement ID: | | | | | | | |
| **Rationale** |  | | | | | | |
| **Acceptance Criteria** | Observation of not display in Sync Screen | | | | | | |
| **Notes** |  | | | | | | |
| **Source** |  | | | | | **Owner** | George Smith |
| **Source Req.** | * 744999135.jpg Detect User Input | | | | | **V&V Method** |  |
| **Type** |  | | | **Priority** | 2 - Medium | **Status** | In-Progress |
| [Req. Template](http://wiki.ford.com/display/RequirementsEngineering/Requirements+Attributes) Version | | 6.0 | End of Requirement | | | | |

Sense Registers' Positions

When "Sense Registers' Positions" component function receives the actuators position, it shall output "Actuators to RCCM"

Satisfied by:

* Functions:
  + Provide User Feedback
  + Sense Register Positions Hardwire
  + Sense Registers' Positions LIN

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Requirement ID: | | | | | | | |
| **Rationale** |  | | | | | | |
| **Acceptance Criteria** | Observation of change in register position based on airflow out of outlets | | | | | | |
| **Notes** |  | | | | | | |
| **Source** |  | | | | | **Owner** | George Smith |
| **Source Req.** | * 744999135.jpg Move to Requested Position | | | | | **V&V Method** |  |
| **Type** |  | | | **Priority** | 2 - Medium | **Status** | Ready for Review |
| [Req. Template](http://wiki.ford.com/display/RequirementsEngineering/Requirements+Attributes) Version | | 6.0 | End of Requirement | | | | |

Closed vent fuction

Satisfied by:

* Functions:
  + Actuate Registers-LIN
  + Provide User Feedback

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Requirement ID: | | | | | | | |
| **Rationale** |  | | | | | | |
| **Acceptance Criteria** | Comparison of voltages with potentiometer table | | | | | | |
| **Notes** |  | | | | | | |
| **Source** |  | | | | | **Owner** | George Smith |
| **Source Req.** | * 744999135.jpg Detect Register Feedback Positions | | | | | **V&V Method** |  |
| **Type** |  | | | **Priority** | 2 - Medium | **Status** | In-Progress |
| [Req. Template](http://wiki.ford.com/display/RequirementsEngineering/Requirements+Attributes) Version | | 6.0 | End of Requirement | | | | |

SYNC Memory for drag dots

SYNC shall remember the positions of drag dots after user has moved and dropped them within their range of motion on HMI. SYNC will continue to show drag dot in that position until one of the following condition occurs:

1. User moves Drag Dot again

2. Registers Dynamic Mode is terminated

3. System goes inactive

4. Control is passed to another feature

Satisfied by:

* Functions:
  + Provide User Feedback
  + Update Settings/Presets

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Requirement ID: | | | | | | | |
| **Rationale** |  | | | | | | |
| **Acceptance Criteria** | Observation of drag dot position compared to register position | | | | | | |
| **Notes** |  | | | | | | |
| **Source** |  | | | | | **Owner** | George Smith |
| **Source Req.** | * 744999135.jpg Detect User Input | | | | | **V&V Method** |  |
| **Type** |  | | | **Priority** | 2 - Medium | **Status** | In-Progress |
| [Req. Template](http://wiki.ford.com/display/RequirementsEngineering/Requirements+Attributes) Version | | 6.0 | End of Requirement | | | | |

SYNC Screen Behavior Climate System ON/OFF

When the climate system display is turned off, it shall disable air flow lines and drag dots.

Satisfied by:

* Functions:
  + Provide Climate System ON/OFF Selection
  + Provide User Feedback

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Requirement ID: | | | | | | | |
| **Rationale** |  | | | | | | |
| **Acceptance Criteria** | Observation of no air flow line displayed on SYNC Screen | | | | | | |
| **Notes** |  | | | | | | |
| **Source** |  | | | | | **Owner** | George Smith |
| **Source Req.** | * 744999135.jpg Detect User Input | | | | | **V&V Method** |  |
| **Type** |  | | | **Priority** | 2 - Medium | **Status** | In-Progress |
| [Req. Template](http://wiki.ford.com/display/RequirementsEngineering/Requirements+Attributes) Version | | 6.0 | End of Requirement | | | | |

#### Technology Function 27255033.jpg **Request Register Change**

Implementation function that receives register move requests from other features through SYNC and forwards them to the RCCM

##### Function Interfaces

###### Inputs

(No inputs have been defined)

###### Outputs

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Outputs: Request Register Change | | | | | | |
| **Logical Signal Name** | **Technical Signal Name** | | | **Mapping Details** *(Conditional)* | **Publisher Interface** | **Connection**  (*Optional)* |
| Review in model  Change Rq | Change Rq :  APIM to RCCM | | |  |  |  |
| [Req. Template](http://wiki.ford.com/display/RequirementsEngineering/Requirements+Attributes) Version | | 6.1 | End of Outputs | | | |

Table 5‑3: Output Signal mappings of Function Request Register Change

###### 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 5‑2: Parameter mappings of Function “MyLogicalFunctionA\_Component1”

###### Interface Requirements

##### Function Requirements

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **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 5‑3: Component Specific Requirements

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

Table 5‑4: Inherited Requirements

###### Component Specific Requirements

Request Air Flow Change

"Request Air Flow Change" component function shall output the signal "APIM to RCCM"

Satisfied by:

* Functions:
  + Request Register Change

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Requirement ID: | | | | | | | |
| **Rationale** |  | | | | | | |
| **Acceptance Criteria** | Observation of air flow change request in SYNC Screen | | | | | | |
| **Notes** | Process register move commands from other features | | | | | | |
| **Source** |  | | | | | **Owner** | George Smith |
| **Source Req.** | * 744999135.jpg Register Controller Functionality | | | | | **V&V Method** |  |
| **Type** |  | | | **Priority** | 2 - Medium | **Status** | Ready for Review |
| [Req. Template](http://wiki.ford.com/display/RequirementsEngineering/Requirements+Attributes) Version | | 6.0 | End of Requirement | | | | |

Rejuvenate Control of Registers

Register Control will be passed from APIM to Rejuvenate Feature when commanded.

Satisfied by:

* Functions:
  + Request Register Change

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Requirement ID: | | | | | | | |
| **Rationale** |  | | | | | | |
| **Acceptance Criteria** | Observation of user input ignored when Rejuvenate feature is using EM Register Feature | | | | | | |
| **Notes** |  | | | | | | |
| **Source** |  | | | | | **Owner** | George Smith |
| **Source Req.** | * 744999135.jpg Register Controller Functionality | | | | | **V&V Method** |  |
| **Type** |  | | | **Priority** | 2 - Medium | **Status** | In-Progress |
| [Req. Template](http://wiki.ford.com/display/RequirementsEngineering/Requirements+Attributes) Version | | 6.0 | End of Requirement | | | | |

#### Technology Function -2026144780.jpg **Provide Climate System ON/OFF Selection**

Implementation function that determines the status of the climate system (on/off) and forwards it to the RCCM

##### Function Interfaces

###### Inputs

(No inputs have been defined)

###### Outputs

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Outputs: Provide Climate System ON/OFF Selection | | | | | | |
| **Logical Signal Name** | **Technical Signal Name** | | | **Mapping Details** *(Conditional)* | **Publisher Interface** | **Connection**  (*Optional)* |
| Review in model  Climate OnOff | Climate OnOff :  Clmt\_Button\_Stat4 | | |  |  |  |
| [Req. Template](http://wiki.ford.com/display/RequirementsEngineering/Requirements+Attributes) Version | | 6.1 | End of Outputs | | | |

Table 5‑3: Output Signal mappings of Function Provide Climate System ON/OFF Selection

###### 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 5‑2: Parameter mappings of Function “MyLogicalFunctionA\_Component1”

###### Interface Requirements

##### Function Requirements

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **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 5‑3: Component Specific Requirements

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

Table 5‑4: Inherited Requirements

###### Component Specific Requirements

SYNC Screen Behavior Climate System ON/OFF

When the climate system display is turned off, it shall disable air flow lines and drag dots.

Satisfied by:

* Functions:
  + Provide Climate System ON/OFF Selection
  + Provide User Feedback

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Requirement ID: | | | | | | | |
| **Rationale** |  | | | | | | |
| **Acceptance Criteria** | Observation of no air flow line displayed on SYNC Screen | | | | | | |
| **Notes** |  | | | | | | |
| **Source** |  | | | | | **Owner** | George Smith |
| **Source Req.** | * 744999135.jpg Detect User Input | | | | | **V&V Method** |  |
| **Type** |  | | | **Priority** | 2 - Medium | **Status** | In-Progress |
| [Req. Template](http://wiki.ford.com/display/RequirementsEngineering/Requirements+Attributes) Version | | 6.0 | End of Requirement | | | | |

### BCM

BCM

### RCCM

RCCM

#### Technology Function -2126799590.jpg **Manage Move Register Requests**

Software function that manages register move requests

##### Function Interfaces

###### Inputs

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Inputs: Manage Move Register Requests | | | | | | |
| **Logical Signal Name** | **Technical Signal Name** | | | **Mapping Details** *(Conditional)* | **Subscriber Interface** | **Connection**  (*Optional)* |
| Review in model  Move Request | Move Request :  APIM to RCCM | | |  |  |  |
| Review in model  Settings/Presets | Settings/Presets :  APIM to RCCM | | |  |  |  |
| Review in model  SensorFeedback | SensorFeedback : | | |  |  |  |
| Review in model  Memory | Memory : | | |  |  |  |
| Review in model  input | input : | | |  |  |  |
| [Req. Template](http://wiki.ford.com/display/RequirementsEngineering/Requirements+Attributes) Version | | 6.1 | End of Inputs | | | |

Table 5‑2: Input Signal mappings of Function Manage Move Register Requests

###### Outputs

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Outputs: Manage Move Register Requests | | | | | | |
| **Logical Signal Name** | **Technical Signal Name** | | | **Mapping Details** *(Conditional)* | **Publisher Interface** | **Connection**  (*Optional)* |
| Review in model  Actuate Registers | Actuate Registers :  RCCM to Actuator HRD | | |  |  |  |
| Review in model  UpdateMemory | UpdateMemory : | | |  |  |  |
| Review in model  RCCM to Actuator LIN | RCCM to Actuator LIN :  RCCM to Actuator LIN | | |  |  |  |
| [Req. Template](http://wiki.ford.com/display/RequirementsEngineering/Requirements+Attributes) Version | | 6.1 | End of Outputs | | | |

Table 5‑3: Output Signal mappings of Function Manage Move Register Requests

###### 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 5‑2: Parameter mappings of Function “MyLogicalFunctionA\_Component1”

###### Interface Requirements

##### Function Requirements

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **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 5‑3: Component Specific Requirements

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

Table 5‑4: Inherited Requirements

###### Component Specific Requirements

SYNC Provide Drag Dots to Last Static Position

SYNC shall place Drag Dots at last memorized static position provided by RCCM when one of the following events occurs that requires registers to return to last memorized postion:

1. Control is retuned to HMI from another feature

2. Registers Dynamic Mode is terminated

3. Ignition status turns to on

4. Manual Climate Panel mode is switched on (from off condition)

5. Climate system is switched on (from off condition)

6. When Registers are opened.

7. Any other situation that requires register to return to a previous memorized location.

Satisfied by:

* Functions:
  + Manage Move Register Requests

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Requirement ID: | | | | | | | |
| **Rationale** |  | | | | | | |
| **Acceptance Criteria** | Observation of Drag Dots moving to last memorized position when each condition is met | | | | | | |
| **Notes** |  | | | | | | |
| **Source** |  | | | | | **Owner** | George Smith |
| **Source Req.** | * 744999135.jpg Detect User Input | | | | | **V&V Method** |  |
| **Type** |  | | | **Priority** | 2 - Medium | **Status** | In-Progress |
| [Req. Template](http://wiki.ford.com/display/RequirementsEngineering/Requirements+Attributes) Version | | 6.0 | End of Requirement | | | | |

Values for Air Motion

RCCM shall provide point-to-point coordinate values (x, y) that vanes will be sequentially moved to, to provide requested Air Motion patterns.

This rule is only applicable to 707.

Satisfied by:

* Functions:
  + Manage Move Register Requests

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Requirement ID: | | | | | | | |
| **Rationale** |  | | | | | | |
| **Acceptance Criteria** | Observation of vanes moving in selected dynamic pattern | | | | | | |
| **Notes** |  | | | | | | |
| **Source** |  | | | | | **Owner** | George Smith |
| **Source Req.** | * 744999135.jpg Detect User Input | | | | | **V&V Method** |  |
| **Type** |  | | | **Priority** | 2 - Medium | **Status** | In-Progress |
| [Req. Template](http://wiki.ford.com/display/RequirementsEngineering/Requirements+Attributes) Version | | 6.0 | End of Requirement | | | | |

Rejuvenate User-Forced Cancellation

If Register HMI is accessed during Rejuvenate feature execution, control shall revert to EM Register in the same manner as when an Air Motion Pattern is exited/cancelled. HMI shall send Manual Change request

Satisfied by:

* Functions:
  + Manage Move Register Requests

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Requirement ID: | | | | | | | |
| **Rationale** |  | | | | | | |
| **Acceptance Criteria** | Observation of control reverting to EM Registers during rejuvenate feature exit | | | | | | |
| **Notes** |  | | | | | | |
| **Source** |  | | | | | **Owner** | George Smith |
| **Source Req.** | * 744999135.jpg Register Controller Functionality | | | | | **V&V Method** |  |
| **Type** |  | | | **Priority** | 2 - Medium | **Status** | In-Progress |
| [Req. Template](http://wiki.ford.com/display/RequirementsEngineering/Requirements+Attributes) Version | | 6.0 | End of Requirement | | | | |

Default Value for Memorizable Presets

RCCM shall provide a default (Factory) value for Memorizeable Presets (x°, y°) in the event that no Preset has been stored. (Table of Values to be provided) This rule is only applicable to 707

Satisfied by:

* Functions:
  + Manage Move Register Requests

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Requirement ID: | | | | | | | |
| **Rationale** |  | | | | | | |
| **Acceptance Criteria** | Observation of a default preset when pressed | | | | | | |
| **Notes** |  | | | | | | |
| **Source** |  | | | | | **Owner** | George Smith |
| **Source Req.** | * 744999135.jpg Move to Requested Position | | | | | **V&V Method** |  |
| **Type** |  | | | **Priority** | 2 - Medium | **Status** | In-Progress |
| [Req. Template](http://wiki.ford.com/display/RequirementsEngineering/Requirements+Attributes) Version | | 6.0 | End of Requirement | | | | |

SYNC provides Drag Dots to Fixed and Memorized Preset Positions

SYNC shall place a Drag Dot at locations called for by Fixed and Memorized Preset Buttons. Memorized Presets is only applicable to 707.

Satisfied by:

* Functions:
  + Manage Move Register Requests

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Requirement ID: | | | | | | | |
| **Rationale** |  | | | | | | |
| **Acceptance Criteria** | Observation of Drag Dots positioned at Fixed or Memorized Preset Buttons | | | | | | |
| **Notes** |  | | | | | | |
| **Source** |  | | | | | **Owner** | George Smith |
| **Source Req.** | * 744999135.jpg Detect User Input | | | | | **V&V Method** |  |
| **Type** |  | | | **Priority** | 2 - Medium | **Status** | In-Progress |
| [Req. Template](http://wiki.ford.com/display/RequirementsEngineering/Requirements+Attributes) Version | | 6.0 | End of Requirement | | | | |

RCCM to APIM Feedback Response to Vane Closure Request

When "Manage Move Requests" component function receives a command to close a register, it shall provide feedback to APIM by setting max value=255 for the Rear/Vertical Vanes angle in the following signals: RgstrVertObl\_An\_Actl=255, RgstrVertObr\_An\_Actl=255, RgstrVertIbr\_An\_Actl=255 , and RgstrVertIbl\_An\_Actl=255.

Satisfied by:

* Functions:
  + Actuate Registers-Hardwire
  + Manage Move Register Requests
  + Provide UserInput

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Requirement ID: | | | | | | | |
| **Rationale** |  | | | | | | |
| **Acceptance Criteria** | Test to detect value 255 when register receives a close command | | | | | | |
| **Notes** |  | | | | | | |
| **Source** |  | | | | | **Owner** | George Smith |
| **Source Req.** | * 744999135.jpg Detect Register Feedback Positions | | | | | **V&V Method** |  |
| **Type** |  | | | **Priority** | 2 - Medium | **Status** | In-Progress |
| [Req. Template](http://wiki.ford.com/display/RequirementsEngineering/Requirements+Attributes) Version | | 6.0 | End of Requirement | | | | |

Fixed Preset Target Values Allternative

When "Manage Move Requests" component function receives the signal ActvButtnLeft\_D\_Rq=onBodyLHS or ActvButtnLeft\_D\_Rq=offBodyLHS it shall send preset coordinates (X,Y) for each request. (Tables Values to be provided)

Satisfied by:

* Functions:
  + Analyze Input
  + Manage Move Register Requests

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Requirement ID: | | | | | | | |
| **Rationale** |  | | | | | | |
| **Acceptance Criteria** | Observation of ON BODY or OFF BODY positions compared to table positions | | | | | | |
| **Notes** |  | | | | | | |
| **Source** |  | | | | | **Owner** | George Smith |
| **Source Req.** | * 744999135.jpg Move to Requested Position | | | | | **V&V Method** |  |
| **Type** |  | | | **Priority** | 2 - Medium | **Status** | In-Progress |
| [Req. Template](http://wiki.ford.com/display/RequirementsEngineering/Requirements+Attributes) Version | | 6.0 | End of Requirement | | | | |

Opening Behavior

When "Manage Move Requests" component function receives the signal OPEN register, it shall move registers to the last memorized static position provided by RCCM.

Satisfied by:

* Functions:
  + Manage Move Register Requests

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Requirement ID: | | | | | | | |
| **Rationale** |  | | | | | | |
| **Acceptance Criteria** | Observation of registers moving to last memorized static position | | | | | | |
| **Notes** |  | | | | | | |
| **Source** |  | | | | | **Owner** | George Smith |
| **Source Req.** | * 744999135.jpg Move to Requested Position | | | | | **V&V Method** |  |
| **Type** |  | | | **Priority** | 2 - Medium | **Status** | In-Progress |
| [Req. Template](http://wiki.ford.com/display/RequirementsEngineering/Requirements+Attributes) Version | | 6.0 | End of Requirement | | | | |

Fixed Preset Target Values

RCCM shall provide a fixed value for ON BODY and OFF BODY preset (x°, y°) (Table Values to be provided). RCCM shall position registers to those fixed values when the command is received.

Satisfied by:

* Functions:
  + Analyze Input
  + Manage Move Register Requests

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Requirement ID: | | | | | | | |
| **Rationale** |  | | | | | | |
| **Acceptance Criteria** | Observation of ON BODY or OFF BODY positions compared to table positions | | | | | | |
| **Notes** |  | | | | | | |
| **Source** |  | | | | | **Owner** | George Smith |
| **Source Req.** | * 744999135.jpg Move to Requested Position | | | | | **V&V Method** |  |
| **Type** |  | | | **Priority** | 2 - Medium | **Status** | In-Progress |
| [Req. Template](http://wiki.ford.com/display/RequirementsEngineering/Requirements+Attributes) Version | | 6.0 | End of Requirement | | | | |

Registers Active in Manual Mode

If Climate system is in manual mode, Registers shall only show as active on HMI if PANEL VENTS is slected by itself or in any combination with the other manual mode buttons (FLOOR VENTS and DEFROSTER VENTS).

Satisfied by:

* Functions:
  + Manage Move Register Requests
  + Provide UserInput

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Requirement ID: | | | | | | | |
| **Rationale** |  | | | | | | |
| **Acceptance Criteria** | Observation of Registers active on HMI for the specified conditions | | | | | | |
| **Notes** |  | | | | | | |
| **Source** |  | | | | | **Owner** | George Smith |
| **Source Req.** | * 744999135.jpg Move to Requested Position | | | | | **V&V Method** |  |
| **Type** |  | | | **Priority** | 2 - Medium | **Status** | In-Progress |
| [Req. Template](http://wiki.ford.com/display/RequirementsEngineering/Requirements+Attributes) Version | | 6.0 | End of Requirement | | | | |

Fixed Preset Target Values Alternative Part 2

When "Manage Move Requests" component function receives the signal ActvButtnRight\_D\_Rq=onBodyLHS or ActvButtnRight\_D\_Rq=offBodyLHS it shall send preset coordinates (X,Y) for each request. (Tables Values to be provided)

Satisfied by:

* Functions:
  + Analyze Input
  + Manage Move Register Requests

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Requirement ID: | | | | | | | |
| **Rationale** |  | | | | | | |
| **Acceptance Criteria** | Observation of ON BODY or OFF BODY positions compared to table positions | | | | | | |
| **Notes** |  | | | | | | |
| **Source** |  | | | | | **Owner** | George Smith |
| **Source Req.** | * 744999135.jpg Move to Requested Position | | | | | **V&V Method** |  |
| **Type** |  | | | **Priority** | 2 - Medium | **Status** | In-Progress |
| [Req. Template](http://wiki.ford.com/display/RequirementsEngineering/Requirements+Attributes) Version | | 6.0 | End of Requirement | | | | |

Memorize Last Register Static Position

Each time Register Aims changes to a new static position, RCCM will memorize the associated coordinates (x°, y°).

Satisfied by:

* Functions:
  + Manage Move Register Requests

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Requirement ID: | | | | | | | |
| **Rationale** |  | | | | | | |
| **Acceptance Criteria** | Observation of a change in register position | | | | | | |
| **Notes** |  | | | | | | |
| **Source** |  | | | | | **Owner** | George Smith |
| **Source Req.** | * 744999135.jpg Move to Requested Position | | | | | **V&V Method** |  |
| **Type** |  | | | **Priority** | 2 - Medium | **Status** | In-Progress |
| [Req. Template](http://wiki.ford.com/display/RequirementsEngineering/Requirements+Attributes) Version | | 6.0 | End of Requirement | | | | |

SYNC Screen Behavior when Climate System is in EATC Mode

SYNC HMI Screen shall show Register Airflow and Drag Dots in "greyed-out" coloring when Climate is in EATC Mode. Drag Dots, aiming, presets and dynamic modes will work as normal.

Satisfied by:

* Functions:
  + Manage Move Register Requests

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Requirement ID: | | | | | | | |
| **Rationale** |  | | | | | | |
| **Acceptance Criteria** | Observation of Drag Dots on SYNC screen | | | | | | |
| **Notes** |  | | | | | | |
| **Source** |  | | | | | **Owner** | George Smith |
| **Source Req.** | * 744999135.jpg Detect Register Feedback Positions | | | | | **V&V Method** |  |
| **Type** |  | | | **Priority** | 2 - Medium | **Status** | In-Progress |
| [Req. Template](http://wiki.ford.com/display/RequirementsEngineering/Requirements+Attributes) Version | | 6.0 | End of Requirement | | | | |

Manage Move Register Requests

When "Manage Move Register Requests" component function receives the signal "APIM to RCCM", it shall output an actuator request.

Satisfied by:

* Functions:
  + Manage Move Register Requests

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Requirement ID: | | | | | | | |
| **Rationale** |  | | | | | | |
| **Acceptance Criteria** | Observation of RCCM updating register positions | | | | | | |
| **Notes** |  | | | | | | |
| **Source** |  | | | | | **Owner** | George Smith |
| **Source Req.** | * 744999135.jpg Register Controller Functionality * 469980868.jpg Send Move Command to Actuators | | | | | **V&V Method** |  |
| **Type** |  | | | **Priority** | 2 - Medium | **Status** | Ready for Review |
| [Req. Template](http://wiki.ford.com/display/RequirementsEngineering/Requirements+Attributes) Version | | 6.0 | End of Requirement | | | | |

Values for Rejuvenate Air Motion

RCCM shall provide point-to-point coordinate values (x, y)that vanes will will be sequetially moved to, to provide requested Air Motion patterns for Rejuvenate Feature. (Table of Values to be provided)

Satisfied by:

* Functions:
  + Manage Move Register Requests

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Requirement ID: | | | | | | | |
| **Rationale** |  | | | | | | |
| **Acceptance Criteria** | Observation of vanes moving in selected dynamic pattern by another feature | | | | | | |
| **Notes** |  | | | | | | |
| **Source** |  | | | | | **Owner** | George Smith |
| **Source Req.** | * 744999135.jpg Detect User Input | | | | | **V&V Method** |  |
| **Type** |  | | | **Priority** | 2 - Medium | **Status** | In-Progress |
| [Req. Template](http://wiki.ford.com/display/RequirementsEngineering/Requirements+Attributes) Version | | 6.0 | End of Requirement | | | | |

#### Technology Function 1825745290.jpg **Manage Feedback**

Software function that receives register move request positions and feeds back the results to the user

##### Function Interfaces

###### Inputs

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Inputs: Manage Feedback | | | | | | |
| **Logical Signal Name** | **Technical Signal Name** | | | **Mapping Details** *(Conditional)* | **Subscriber Interface** | **Connection**  (*Optional)* |
| Review in model  LIN Out | LIN Out :  Actuator to RCCM LIN | | |  |  |  |
| Review in model  HRD In | HRD In :  Actuator to RCCM LIN | | |  |  |  |
| [Req. Template](http://wiki.ford.com/display/RequirementsEngineering/Requirements+Attributes) Version | | 6.1 | End of Inputs | | | |

Table 5‑2: Input Signal mappings of Function Manage Feedback

###### Outputs

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Outputs: Manage Feedback | | | | | | |
| **Logical Signal Name** | **Technical Signal Name** | | | **Mapping Details** *(Conditional)* | **Publisher Interface** | **Connection**  (*Optional)* |
| Review in model  Feedback | Feedback :  RCCM to APIM ActvButtn | | |  |  |  |
| Review in model  SensorFeedback | SensorFeedback : | | |  |  |  |
| [Req. Template](http://wiki.ford.com/display/RequirementsEngineering/Requirements+Attributes) Version | | 6.1 | End of Outputs | | | |

Table 5‑3: Output Signal mappings of Function Manage Feedback

###### 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 5‑2: Parameter mappings of Function “MyLogicalFunctionA\_Component1”

###### Interface Requirements

##### Function Requirements

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **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 5‑3: Component Specific Requirements

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

Table 5‑4: Inherited Requirements

###### Component Specific Requirements

SYNC Drag Dot Behavior for Closing Register

If Register is closed, DRAG Dot shall be positioned on top of the center of the register with no airflow lines.

Satisfied by:

* Functions:
  + Manage Feedback

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Requirement ID: | | | | | | | |
| **Rationale** |  | | | | | | |
| **Acceptance Criteria** | Observation of Drag Dot position when register is closed | | | | | | |
| **Notes** |  | | | | | | |
| **Source** |  | | | | | **Owner** | George Smith |
| **Source Req.** | * 744999135.jpg Detect User Input | | | | | **V&V Method** |  |
| **Type** |  | | | **Priority** | 2 - Medium | **Status** | In-Progress |
| [Req. Template](http://wiki.ford.com/display/RequirementsEngineering/Requirements+Attributes) Version | | 6.0 | End of Requirement | | | | |

SYNC Drag Dot Behavior for Opening Register

After sending Register Open Command, SYNC shall remove CLOSED symbol and move Drag Dot to previous Register position once provided by RCCM .

Satisfied by:

* Functions:
  + Manage Feedback

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Requirement ID: | | | | | | | |
| **Rationale** |  | | | | | | |
| **Acceptance Criteria** |  | | | | | | |
| **Notes** |  | | | | | | |
| **Source** |  | | | | | **Owner** | George Smith |
| **Source Req.** | * 744999135.jpg Detect User Input | | | | | **V&V Method** |  |
| **Type** |  | | | **Priority** | 2 - Medium | **Status** | In-Progress |
| [Req. Template](http://wiki.ford.com/display/RequirementsEngineering/Requirements+Attributes) Version | | 6.0 | End of Requirement | | | | |

Manage Feedback

When "Manage Feedback" component function receives the signal from actuator, it shall output signal "RCCM to APIM". RCCM will provide Actuator position status to APIM per CSI#XXXX

Satisfied by:

* Functions:
  + Manage Feedback

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Requirement ID: | | | | | | | |
| **Rationale** |  | | | | | | |
| **Acceptance Criteria** | Observation of feedback on SYNC Display Screen | | | | | | |
| **Notes** |  | | | | | | |
| **Source** |  | | | | | **Owner** | George Smith |
| **Source Req.** | * 744999135.jpg Register Controller Functionality * -2006039999.jpg Send Feedback to HMI | | | | | **V&V Method** |  |
| **Type** |  | | | **Priority** | 2 - Medium | **Status** | Ready for Review |
| [Req. Template](http://wiki.ford.com/display/RequirementsEngineering/Requirements+Attributes) Version | | 6.0 | End of Requirement | | | | |

#### Technology Function 1647954732.jpg **Analyze Input**

Software function that updates register move settings and presets. Provides move requests to manage move requests function

##### Function Interfaces

###### Inputs

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Inputs: Analyze Input | | | | | | |
| **Logical Signal Name** | **Technical Signal Name** | | | **Mapping Details** *(Conditional)* | **Subscriber Interface** | **Connection**  (*Optional)* |
| Review in model  Analyze Input | Analyze Input :  APIM to RCCM | | |  |  |  |
| [Req. Template](http://wiki.ford.com/display/RequirementsEngineering/Requirements+Attributes) Version | | 6.1 | End of Inputs | | | |

Table 5‑2: Input Signal mappings of Function Analyze Input

###### Outputs

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Outputs: Analyze Input | | | | | | |
| **Logical Signal Name** | **Technical Signal Name** | | | **Mapping Details** *(Conditional)* | **Publisher Interface** | **Connection**  (*Optional)* |
| Review in model  MoveRequest | MoveRequest :  APIM to RCCM | | |  |  |  |
| Review in model  MoveRequest | MoveRequest :  APIM to RCCM | | |  |  |  |
| [Req. Template](http://wiki.ford.com/display/RequirementsEngineering/Requirements+Attributes) Version | | 6.1 | End of Outputs | | | |

Table 5‑3: Output Signal mappings of Function Analyze Input

###### 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 5‑2: Parameter mappings of Function “MyLogicalFunctionA\_Component1”

###### Interface Requirements

##### Function Requirements

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **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 5‑3: Component Specific Requirements

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

Table 5‑4: Inherited Requirements

###### Component Specific Requirements

Analyse & Arbitrate Input

RCCM shall arbitrate mutiple requests for Register Control according the following hierachy:

1) Rejuvenate

2) HMI Request

Satisfied by:

* Functions:
  + Analyze Input

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Requirement ID: | | | | | | | |
| **Rationale** |  | | | | | | |
| **Acceptance Criteria** | Observation of updated settings or preset changes based on airflow output from registers | | | | | | |
| **Notes** |  | | | | | | |
| **Source** |  | | | | | **Owner** | George Smith |
| **Source Req.** | * 744999135.jpg Register Controller Functionality * -736526547.jpg Provide Display Feedback Positions | | | | | **V&V Method** |  |
| **Type** |  | | | **Priority** | 2 - Medium | **Status** | Ready for Review |
| [Req. Template](http://wiki.ford.com/display/RequirementsEngineering/Requirements+Attributes) Version | | 6.0 | End of Requirement | | | | |

Coordinate System for Each Register

SYNC shall provide a non visible "range of coverage" box around each register on the HMI screen. This box provides the area of coverage for each register relative to the Cockpit as depicted by the Wireframe image. SYNC shall provide coordiate tracking of the Drag Dots within these zones in (X,Y) format, where X is the cross car componet of the Drag Dot and Y is the up down componet of the Drag Dot. The origin (0,0) will be in the lower left corner of the box. The maximum values shall be (150, 90) in the upper right corner of the box. HMI shall return a linearly proportional position of the Drag Dot anywhere within the box that it is positioned. The nomimal units of the Drag Dot shall be degrees which coincide with the Register Vane roations required to proivde air to the relative position as depicted on the Cockpit HMI wireframe image.

Satisfied by:

* Functions:
  + Analyze Input
  + Provide UserInput

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Requirement ID: | | | | | | | |
| **Rationale** |  | | | | | | |
| **Acceptance Criteria** | Observation of vane movement when user changes register position on SYNC screen | | | | | | |
| **Notes** |  | | | | | | |
| **Source** |  | | | | | **Owner** | George Smith |
| **Source Req.** | * 744999135.jpg Move to Requested Position | | | | | **V&V Method** |  |
| **Type** |  | | | **Priority** | 2 - Medium | **Status** | In-Progress |
| [Req. Template](http://wiki.ford.com/display/RequirementsEngineering/Requirements+Attributes) Version | | 6.0 | End of Requirement | | | | |

Fixed Preset Target Values Allternative

When "Manage Move Requests" component function receives the signal ActvButtnLeft\_D\_Rq=onBodyLHS or ActvButtnLeft\_D\_Rq=offBodyLHS it shall send preset coordinates (X,Y) for each request. (Tables Values to be provided)

Satisfied by:

* Functions:
  + Analyze Input
  + Manage Move Register Requests

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Requirement ID: | | | | | | | |
| **Rationale** |  | | | | | | |
| **Acceptance Criteria** | Observation of ON BODY or OFF BODY positions compared to table positions | | | | | | |
| **Notes** |  | | | | | | |
| **Source** |  | | | | | **Owner** | George Smith |
| **Source Req.** | * 744999135.jpg Move to Requested Position | | | | | **V&V Method** |  |
| **Type** |  | | | **Priority** | 2 - Medium | **Status** | In-Progress |
| [Req. Template](http://wiki.ford.com/display/RequirementsEngineering/Requirements+Attributes) Version | | 6.0 | End of Requirement | | | | |

Fixed Preset Target Values

RCCM shall provide a fixed value for ON BODY and OFF BODY preset (x°, y°) (Table Values to be provided). RCCM shall position registers to those fixed values when the command is received.

Satisfied by:

* Functions:
  + Analyze Input
  + Manage Move Register Requests

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Requirement ID: | | | | | | | |
| **Rationale** |  | | | | | | |
| **Acceptance Criteria** | Observation of ON BODY or OFF BODY positions compared to table positions | | | | | | |
| **Notes** |  | | | | | | |
| **Source** |  | | | | | **Owner** | George Smith |
| **Source Req.** | * 744999135.jpg Move to Requested Position | | | | | **V&V Method** |  |
| **Type** |  | | | **Priority** | 2 - Medium | **Status** | In-Progress |
| [Req. Template](http://wiki.ford.com/display/RequirementsEngineering/Requirements+Attributes) Version | | 6.0 | End of Requirement | | | | |

Default Value for Last Static Position

RCCM shall provide a default (Factory) value for "Last Static Position" (x°, y°) in the event that no "Last Position" has been stored. (Table of Values to be provided)

Satisfied by:

* Functions:
  + Analyze Input

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Requirement ID: | | | | | | | |
| **Rationale** |  | | | | | | |
| **Acceptance Criteria** | Observation of default value for "Last Static Position" when preset is pressed | | | | | | |
| **Notes** |  | | | | | | |
| **Source** |  | | | | | **Owner** | George Smith |
| **Source Req.** | * -736526547.jpg Retrieve LastPosition Value | | | | | **V&V Method** |  |
| **Type** |  | | | **Priority** | 2 - Medium | **Status** | In-Progress |
| [Req. Template](http://wiki.ford.com/display/RequirementsEngineering/Requirements+Attributes) Version | | 6.0 | End of Requirement | | | | |

Fixed Preset Target Values Alternative Part 2

When "Manage Move Requests" component function receives the signal ActvButtnRight\_D\_Rq=onBodyLHS or ActvButtnRight\_D\_Rq=offBodyLHS it shall send preset coordinates (X,Y) for each request. (Tables Values to be provided)

Satisfied by:

* Functions:
  + Analyze Input
  + Manage Move Register Requests

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Requirement ID: | | | | | | | |
| **Rationale** |  | | | | | | |
| **Acceptance Criteria** | Observation of ON BODY or OFF BODY positions compared to table positions | | | | | | |
| **Notes** |  | | | | | | |
| **Source** |  | | | | | **Owner** | George Smith |
| **Source Req.** | * 744999135.jpg Move to Requested Position | | | | | **V&V Method** |  |
| **Type** |  | | | **Priority** | 2 - Medium | **Status** | In-Progress |
| [Req. Template](http://wiki.ford.com/display/RequirementsEngineering/Requirements+Attributes) Version | | 6.0 | End of Requirement | | | | |

#### Technology Function -2135922888.jpg **Update Settings/Presets**

Software implementation function that provides the RCCM with an update of the register settings and presets

##### Function Interfaces

###### Inputs

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Inputs: Update Settings/Presets | | | | | | |
| **Logical Signal Name** | **Technical Signal Name** | | | **Mapping Details** *(Conditional)* | **Subscriber Interface** | **Connection**  (*Optional)* |
| Review in model  Update Settings/Presets | Update Settings/Presets :  APIM to RCCM | | |  |  |  |
| [Req. Template](http://wiki.ford.com/display/RequirementsEngineering/Requirements+Attributes) Version | | 6.1 | End of Inputs | | | |

Table 5‑2: Input Signal mappings of Function Update Settings/Presets

###### Outputs

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Outputs: Update Settings/Presets | | | | | | |
| **Logical Signal Name** | **Technical Signal Name** | | | **Mapping Details** *(Conditional)* | **Publisher Interface** | **Connection**  (*Optional)* |
| Review in model  Settings/Presets | Settings/Presets :  APIM to RCCM | | |  |  |  |
| [Req. Template](http://wiki.ford.com/display/RequirementsEngineering/Requirements+Attributes) Version | | 6.1 | End of Outputs | | | |

Table 5‑3: Output Signal mappings of Function Update Settings/Presets

###### 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 5‑2: Parameter mappings of Function “MyLogicalFunctionA\_Component1”

###### Interface Requirements

##### Function Requirements

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **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 5‑3: Component Specific Requirements

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

Table 5‑4: Inherited Requirements

###### Component Specific Requirements

SYNC Memory for drag dots

SYNC shall remember the positions of drag dots after user has moved and dropped them within their range of motion on HMI. SYNC will continue to show drag dot in that position until one of the following condition occurs:

1. User moves Drag Dot again

2. Registers Dynamic Mode is terminated

3. System goes inactive

4. Control is passed to another feature

Satisfied by:

* Functions:
  + Provide User Feedback
  + Update Settings/Presets

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Requirement ID: | | | | | | | |
| **Rationale** |  | | | | | | |
| **Acceptance Criteria** | Observation of drag dot position compared to register position | | | | | | |
| **Notes** |  | | | | | | |
| **Source** |  | | | | | **Owner** | George Smith |
| **Source Req.** | * 744999135.jpg Detect User Input | | | | | **V&V Method** |  |
| **Type** |  | | | **Priority** | 2 - Medium | **Status** | In-Progress |
| [Req. Template](http://wiki.ford.com/display/RequirementsEngineering/Requirements+Attributes) Version | | 6.0 | End of Requirement | | | | |

SYNC Behavior for Memorizable Presets

SYNC shall move move registers to coordiante (x°, y°) stored in Memorizeable Preset when the Memorizable Preset Buttons are tapped momentarily. This rule is only applicable to 707.

Satisfied by:

* Functions:
  + Update Settings/Presets

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Requirement ID: | | | | | | | |
| **Rationale** |  | | | | | | |
| **Acceptance Criteria** | Observation of registers moving to coordinate positions | | | | | | |
| **Notes** |  | | | | | | |
| **Source** |  | | | | | **Owner** | George Smith |
| **Source Req.** | * 744999135.jpg Detect User Input | | | | | **V&V Method** |  |
| **Type** |  | | | **Priority** | 2 - Medium | **Status** | In-Progress |
| [Req. Template](http://wiki.ford.com/display/RequirementsEngineering/Requirements+Attributes) Version | | 6.0 | End of Requirement | | | | |

Preset Default State

RCCM shall provide preset1 and preset2 default values in a table (40/30 Horz/Vert)tbd

Satisfied by:

* Functions:
  + Update Settings/Presets

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Requirement ID: | | | | | | | |
| **Rationale** |  | | | | | | |
| **Acceptance Criteria** | Observation of presets moving to default value | | | | | | |
| **Notes** |  | | | | | | |
| **Source** |  | | | | | **Owner** | George Smith |
| **Source Req.** | * 744999135.jpg Move to Requested Position | | | | | **V&V Method** |  |
| **Type** |  | | | **Priority** | 2 - Medium | **Status** | In-Progress |
| [Req. Template](http://wiki.ford.com/display/RequirementsEngineering/Requirements+Attributes) Version | | 6.0 | End of Requirement | | | | |

SYNC & RCCM Memorizing for Memorizable Presets

SYNC shall send a memorize postion command to the RCCM when Memorized Preset Button is held for more than 2 seconds. SYNC shall emit an audible beep when it has received confirmation from RCCM via Signals "ActvButtnLeft\_D\_Stat/0x3=presetOneLHS", "ActvButtnRight\_D\_Stat/0x3=presetOneRHS" that the current positions have been memorized. This rule is only applicable to 707.

Satisfied by:

* Functions:
  + Update Settings/Presets

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Requirement ID: | | | | | | | |
| **Rationale** |  | | | | | | |
| **Acceptance Criteria** | Observation of audible beep when preset is held for more than 2 seconds | | | | | | |
| **Notes** |  | | | | | | |
| **Source** |  | | | | | **Owner** | George Smith |
| **Source Req.** | * 744999135.jpg Move to Requested Position | | | | | **V&V Method** |  |
| **Type** |  | | | **Priority** | 2 - Medium | **Status** | In-Progress |
| [Req. Template](http://wiki.ford.com/display/RequirementsEngineering/Requirements+Attributes) Version | | 6.0 | End of Requirement | | | | |

#### Technology Function -301297452.jpg **Update LastPosition Memory**

Software implementation function that provides an update of the last register position to memory

##### Function Interfaces

###### Inputs

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Inputs: Update LastPosition Memory | | | | | | |
| **Logical Signal Name** | **Technical Signal Name** | | | **Mapping Details** *(Conditional)* | **Subscriber Interface** | **Connection**  (*Optional)* |
| Review in model  UpdateMemory | UpdateMemory : | | |  |  |  |
| [Req. Template](http://wiki.ford.com/display/RequirementsEngineering/Requirements+Attributes) Version | | 6.1 | End of Inputs | | | |

Table 5‑2: Input Signal mappings of Function Update LastPosition Memory

###### Outputs

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Outputs: Update LastPosition Memory | | | | | | |
| **Logical Signal Name** | **Technical Signal Name** | | | **Mapping Details** *(Conditional)* | **Publisher Interface** | **Connection**  (*Optional)* |
| Review in model  Memory | Memory : | | |  |  |  |
| [Req. Template](http://wiki.ford.com/display/RequirementsEngineering/Requirements+Attributes) Version | | 6.1 | End of Outputs | | | |

Table 5‑3: Output Signal mappings of Function Update LastPosition Memory

###### 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 5‑2: Parameter mappings of Function “MyLogicalFunctionA\_Component1”

###### Interface Requirements

##### Function Requirements

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **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 5‑3: Component Specific Requirements

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

Table 5‑4: Inherited Requirements

###### Component Specific Requirements

Update LastPosition Memory

RCCM shall memorize position of Register each time it is moved to a new static position. This becomes the new Last Position value.

Satisfied by:

* Functions:
  + Update LastPosition Memory

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Requirement ID: | | | | | | | |
| **Rationale** |  | | | | | | |
| **Acceptance Criteria** | Observation of memory update on SYNC Display | | | | | | |
| **Notes** |  | | | | | | |
| **Source** |  | | | | | **Owner** | George Smith |
| **Source Req.** | * 744999135.jpg Register Controller Functionality * -1557646014.jpg Retrieve Registers' Positions * -1557646014.jpg Save to LastPosition * -1557646014.jpg Save LastPosition to BeforeClosedPosition | | | | | **V&V Method** |  |
| **Type** |  | | | **Priority** | 2 - Medium | **Status** | Ready for Review |
| [Req. Template](http://wiki.ford.com/display/RequirementsEngineering/Requirements+Attributes) Version | | 6.0 | End of Requirement | | | | |

Closed Register Memorization

RCCM shall not memorize the postion for a Register CLOSED command. The previous static value shall remain in memory.

Satisfied by:

* Functions:
  + Update LastPosition Memory

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Requirement ID: | | | | | | | |
| **Rationale** |  | | | | | | |
| **Acceptance Criteria** | Observation of previous preset position when register is closed | | | | | | |
| **Notes** |  | | | | | | |
| **Source** |  | | | | | **Owner** | George Smith |
| **Source Req.** | * -1557646014.jpg Save LastPosition to BeforeClosedPosition | | | | | **V&V Method** |  |
| **Type** |  | | | **Priority** | 2 - Medium | **Status** | In-Progress |
| [Req. Template](http://wiki.ford.com/display/RequirementsEngineering/Requirements+Attributes) Version | | 6.0 | End of Requirement | | | | |

#### Technology Function -2026144780.jpg **Provide Climate System ON/OFF Selection**

Implementation function that determines the status of the climate system (on/off) and forwards it to the RCCM

##### Function Interfaces

###### Inputs

(No inputs have been defined)

###### Outputs

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Outputs: Provide Climate System ON/OFF Selection | | | | | | |
| **Logical Signal Name** | **Technical Signal Name** | | | **Mapping Details** *(Conditional)* | **Publisher Interface** | **Connection**  (*Optional)* |
| Review in model  Climate OnOff | Climate OnOff :  Clmt\_Button\_Stat4 | | |  |  |  |
| [Req. Template](http://wiki.ford.com/display/RequirementsEngineering/Requirements+Attributes) Version | | 6.1 | End of Outputs | | | |

Table 5‑3: Output Signal mappings of Function Provide Climate System ON/OFF Selection

###### 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 5‑2: Parameter mappings of Function “MyLogicalFunctionA\_Component1”

###### Interface Requirements

##### Function Requirements

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **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 5‑3: Component Specific Requirements

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

Table 5‑4: Inherited Requirements

###### Component Specific Requirements

SYNC Screen Behavior Climate System ON/OFF

When the climate system display is turned off, it shall disable air flow lines and drag dots.

Satisfied by:

* Functions:
  + Provide Climate System ON/OFF Selection
  + Provide User Feedback

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Requirement ID: | | | | | | | |
| **Rationale** |  | | | | | | |
| **Acceptance Criteria** | Observation of no air flow line displayed on SYNC Screen | | | | | | |
| **Notes** |  | | | | | | |
| **Source** |  | | | | | **Owner** | George Smith |
| **Source Req.** | * 744999135.jpg Detect User Input | | | | | **V&V Method** |  |
| **Type** |  | | | **Priority** | 2 - Medium | **Status** | In-Progress |
| [Req. Template](http://wiki.ford.com/display/RequirementsEngineering/Requirements+Attributes) Version | | 6.0 | End of Requirement | | | | |

#### Technology Function 1354956762.jpg **Provide IgnitionStatus**

Implementation function that provides the EM Registers feature with the ignition status of the vehicle

##### Function Interfaces

###### Inputs

(No inputs have been defined)

###### Outputs

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Outputs: Provide IgnitionStatus | | | | | | |
| **Logical Signal Name** | **Technical Signal Name** | | | **Mapping Details** *(Conditional)* | **Publisher Interface** | **Connection**  (*Optional)* |
| Review in model  Ig Status  vehicleStatus | Ig Status :  VehStat | | |  |  |  |
| [Req. Template](http://wiki.ford.com/display/RequirementsEngineering/Requirements+Attributes) Version | | 6.1 | End of Outputs | | | |

Table 5‑3: Output Signal mappings of Function Provide IgnitionStatus

###### 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 5‑2: Parameter mappings of Function “MyLogicalFunctionA\_Component1”

###### Interface Requirements

##### Function Requirements

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **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 5‑3: Component Specific Requirements

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

Table 5‑4: Inherited Requirements

###### Component Specific Requirements

## Requirements on Connections

### Networks

#### “CAN Bus xxx”

##### Protocol Requirements

##### Electrical Requirements

#### “LIN Bus xxx”

##### Protocol Requirements

###### Schedule Table

##### Electrical Requirements

#### “Ethernet xxx”

### HW I/Os

#### “HW I/O xxx”

## Requirements on Development Process

# Open Concerns

| 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

No Revision History found.

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

## Template Revisions

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 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

Actuate Register Command

Signal requesting actuation of register position changes

|  |  |  |
| --- | --- | --- |
| **ASIL** | | Choose an item. |
| **Encoding Type Name** | |  |
| 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 |  |
| Value 2 | … |
| … | … |
|  |  |
|  |  |
| **Unit** | |  |

Table: Signal Details of Actuate Register Command

EM\_DesiredPositionsRequest

Manual move register requests

|  |  |  |
| --- | --- | --- |
| **ASIL** | | Choose an item. |
| **Encoding Type Name** | |  |
| 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 |  |
| Value 2 | … |
| … | … |
|  |  |
|  |  |
| **Unit** | |  |

Table: Signal Details of EM\_DesiredPositionsRequest

RegisterPositionHMIFeedback

Register sensor position feedback to HMI

|  |  |  |
| --- | --- | --- |
| **ASIL** | | Choose an item. |
| **Encoding Type Name** | |  |
| 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 |  |
| Value 2 | … |
| … | … |
|  |  |
|  |  |
| **Unit** | |  |

Table: Signal Details of RegisterPositionHMIFeedback

RegisterPositionsSensorFeedback

Signal to provide register position feedback to actuator

|  |  |  |
| --- | --- | --- |
| **ASIL** | | Choose an item. |
| **Encoding Type Name** | |  |
| 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 |  |
| Value 2 | … |
| … | … |
|  |  |
|  |  |
| **Unit** | |  |

Table: Signal Details of RegisterPositionsSensorFeedback

vehicleStatus

Vehicle on off status used by register move commands

|  |  |  |
| --- | --- | --- |
| **ASIL** | | Choose an item. |
| **Encoding Type Name** | |  |
| 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 |  |
| Value 2 | … |
| … | … |
|  |  |
|  |  |
| **Unit** | |  |

Table: Signal Details of vehicleStatus

### Logical Parameters

### Technical Signals

SensFdbkLHIBHoriz

Sense left hand horizontal inboard register vane positions and forward to RCCM

|  |  |  |
| --- | --- | --- |
| **ASIL** | | Choose an item. |
| **Init Default Value** | |  |
| **Encoding Type Name** | |  |
| 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 |  |
| Value 2 | … |
| … | … |
|  |  |
|  |  |
| **Unit** | |  |

Table: Signal Details of SensFdbkLHIBHoriz

RgstrVertllbr\_An\_Actl

Aim Status of Righthand Inboard Register Vertical Vanes

|  |  |  |
| --- | --- | --- |
| **ASIL** | | Choose an item. |
| **Init Default Value** | |  |
| **Encoding Type Name** | |  |
| 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 |  |
| Value 2 | … |
| … | … |
|  |  |
|  |  |
| **Unit** | |  |

Table: Signal Details of RgstrVertllbr\_An\_Actl

RgstrSetObr\_D\_Rq

Change Aim Command for Righthand Intboard Register

|  |  |  |
| --- | --- | --- |
| **ASIL** | | Choose an item. |
| **Init Default Value** | |  |
| **Encoding Type Name** | |  |
| 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 |  |
| Value 2 | … |
| … | … |
|  |  |
|  |  |
| **Unit** | |  |

Table: Signal Details of RgstrSetObr\_D\_Rq

Request LIN

Place holder for LIN signals 707 program

|  |  |  |
| --- | --- | --- |
| **ASIL** | | Choose an item. |
| **Init Default Value** | |  |
| **Encoding Type Name** | |  |
| 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 |  |
| Value 2 | … |
| … | … |
|  |  |
|  |  |
| **Unit** | |  |

Table: Signal Details of Request LIN

SensFdbkRHIBHoriz

Sense right hand horizontal inboard register vane positions and forward to RCCM

|  |  |  |
| --- | --- | --- |
| **ASIL** | | Choose an item. |
| **Init Default Value** | |  |
| **Encoding Type Name** | |  |
| 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 |  |
| Value 2 | … |
| … | … |
|  |  |
|  |  |
| **Unit** | |  |

Table: Signal Details of SensFdbkRHIBHoriz

APIM to RCCM Rgstr Set

|  |  |  |
| --- | --- | --- |
| **ASIL** | | Choose an item. |
| **Init Default Value** | |  |
| **Encoding Type Name** | |  |
| 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 |  |
| Value 2 | … |
| … | … |
|  |  |
|  |  |
| **Unit** | |  |

Table: Signal Details of APIM to RCCM Rgstr Set

ActCmdLHIBHoriz

Signal from RCCM to left hand inboard horizontal vane register actuator

|  |  |  |
| --- | --- | --- |
| **ASIL** | | Choose an item. |
| **Init Default Value** | |  |
| **Encoding Type Name** | |  |
| 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 |  |
| Value 2 | … |
| … | … |
|  |  |
|  |  |
| **Unit** | |  |

Table: Signal Details of ActCmdLHIBHoriz

ActvButtnLeft\_D\_Stat

Subfeature Selection Status for Left Side Registers:

\*NonePressed

\*OnBodyLHS

\*OffBodyLHS

\*PresetOneLHS

\*PresetTwoLHS

\*Cycle\_8\_LHS

\*Cycle\_C\_LHS

\*Cycle\_O\_LHS

\*Cycle\_--\_LHS

\*Cycle\_I\_LHS

|  |  |  |
| --- | --- | --- |
| **ASIL** | | Choose an item. |
| **Init Default Value** | |  |
| **Encoding Type Name** | |  |
| 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 |  |
| Value 2 | … |
| … | … |
|  |  |
|  |  |
| **Unit** | |  |

Table: Signal Details of ActvButtnLeft\_D\_Stat

Feedback LIN

Place holder for LIN feedback signals 707 program

|  |  |  |
| --- | --- | --- |
| **ASIL** | | Choose an item. |
| **Init Default Value** | |  |
| **Encoding Type Name** | |  |
| 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 |  |
| Value 2 | … |
| … | … |
|  |  |
|  |  |
| **Unit** | |  |

Table: Signal Details of Feedback LIN

RgstrVertlbr\_An\_Rq

Command Position of Vertical Vanes for Righthand Inboard Register

|  |  |  |
| --- | --- | --- |
| **ASIL** | | Choose an item. |
| **Init Default Value** | |  |
| **Encoding Type Name** | |  |
| 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 |  |
| Value 2 | … |
| … | … |
|  |  |
|  |  |
| **Unit** | |  |

Table: Signal Details of RgstrVertlbr\_An\_Rq

RCCM to Actuator HRD

|  |  |  |
| --- | --- | --- |
| **ASIL** | | Choose an item. |
| **Init Default Value** | |  |
| **Encoding Type Name** | |  |
| 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 |  |
| Value 2 | … |
| … | … |
|  |  |
|  |  |
| **Unit** | |  |

Table: Signal Details of RCCM to Actuator HRD

RgstrHzntlIbl\_An\_Actl

Aim Status of Lefthand Inboard Register Horizontal Vanes

|  |  |  |
| --- | --- | --- |
| **ASIL** | | Choose an item. |
| **Init Default Value** | |  |
| **Encoding Type Name** | |  |
| 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 |  |
| Value 2 | … |
| … | … |
|  |  |
|  |  |
| **Unit** | |  |

Table: Signal Details of RgstrHzntlIbl\_An\_Actl

Actuator to Sensor HRD

|  |  |  |
| --- | --- | --- |
| **ASIL** | | Choose an item. |
| **Init Default Value** | |  |
| **Encoding Type Name** | |  |
| 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 |  |
| Value 2 | … |
| … | … |
|  |  |
|  |  |
| **Unit** | |  |

Table: Signal Details of Actuator to Sensor HRD

SensFdbkLHOBHoriz

Sense left hand horizontal outboard register vane positions and forward to RCCM

|  |  |  |
| --- | --- | --- |
| **ASIL** | | Choose an item. |
| **Init Default Value** | |  |
| **Encoding Type Name** | |  |
| 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 |  |
| Value 2 | … |
| … | … |
|  |  |
|  |  |
| **Unit** | |  |

Table: Signal Details of SensFdbkLHOBHoriz

ActCmdLHOBVert

Signal from RCCM to left hand outboard vertical vane register actuator

|  |  |  |
| --- | --- | --- |
| **ASIL** | | Choose an item. |
| **Init Default Value** | |  |
| **Encoding Type Name** | |  |
| 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 |  |
| Value 2 | … |
| … | … |
|  |  |
|  |  |
| **Unit** | |  |

Table: Signal Details of ActCmdLHOBVert

ActCmdRHOBVert

Signal from RCCM to right hand outboard vertical vane register actuator

|  |  |  |
| --- | --- | --- |
| **ASIL** | | Choose an item. |
| **Init Default Value** | |  |
| **Encoding Type Name** | |  |
| 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 |  |
| Value 2 | … |
| … | … |
|  |  |
|  |  |
| **Unit** | |  |

Table: Signal Details of ActCmdRHOBVert

SensedRegPosRHOBhoriz

|  |  |  |
| --- | --- | --- |
| **ASIL** | | Choose an item. |
| **Init Default Value** | |  |
| **Encoding Type Name** | |  |
| 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 |  |
| Value 2 | … |
| … | … |
|  |  |
|  |  |
| **Unit** | |  |

Table: Signal Details of SensedRegPosRHOBhoriz

RgstrVertObl\_An\_Actl

Aim Status of Righthand Inboard Register Vertical Vanes

|  |  |  |
| --- | --- | --- |
| **ASIL** | | Choose an item. |
| **Init Default Value** | |  |
| **Encoding Type Name** | |  |
| 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 |  |
| Value 2 | … |
| … | … |
|  |  |
|  |  |
| **Unit** | |  |

Table: Signal Details of RgstrVertObl\_An\_Actl

ActCmdRHOBHoriz

Signal from RCCM to right hand outboard horizontal vane register actuator

|  |  |  |
| --- | --- | --- |
| **ASIL** | | Choose an item. |
| **Init Default Value** | |  |
| **Encoding Type Name** | |  |
| 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 |  |
| Value 2 | … |
| … | … |
|  |  |
|  |  |
| **Unit** | |  |

Table: Signal Details of ActCmdRHOBHoriz

RgstrHzntlObr\_An\_Actl

Aim Status of Righthand Outboard Register Horizontal Vanes

|  |  |  |
| --- | --- | --- |
| **ASIL** | | Choose an item. |
| **Init Default Value** | |  |
| **Encoding Type Name** | |  |
| 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 |  |
| Value 2 | … |
| … | … |
|  |  |
|  |  |
| **Unit** | |  |

Table: Signal Details of RgstrHzntlObr\_An\_Actl

RgstrHzntlObl\_An\_Rq

Command Position of Horizontal Vanes for Righthand Outboard Register

|  |  |  |
| --- | --- | --- |
| **ASIL** | | Choose an item. |
| **Init Default Value** | |  |
| **Encoding Type Name** | |  |
| 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 |  |
| Value 2 | … |
| … | … |
|  |  |
|  |  |
| **Unit** | |  |

Table: Signal Details of RgstrHzntlObl\_An\_Rq

RgstrVertObr\_An\_Actl

Aim Status of Righthand Outboard Register Vertical Vanes

|  |  |  |
| --- | --- | --- |
| **ASIL** | | Choose an item. |
| **Init Default Value** | |  |
| **Encoding Type Name** | |  |
| 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 |  |
| Value 2 | … |
| … | … |
|  |  |
|  |  |
| **Unit** | |  |

Table: Signal Details of RgstrVertObr\_An\_Actl

SensedRegPosLHOBhoriz

|  |  |  |
| --- | --- | --- |
| **ASIL** | | Choose an item. |
| **Init Default Value** | |  |
| **Encoding Type Name** | |  |
| 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 |  |
| Value 2 | … |
| … | … |
|  |  |
|  |  |
| **Unit** | |  |

Table: Signal Details of SensedRegPosLHOBhoriz

RgstrHzntllbr\_An\_Actl

Aim Status of Righthand Inboard Register Horizontal Vanes

|  |  |  |
| --- | --- | --- |
| **ASIL** | | Choose an item. |
| **Init Default Value** | |  |
| **Encoding Type Name** | |  |
| 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 |  |
| Value 2 | … |
| … | … |
|  |  |
|  |  |
| **Unit** | |  |

Table: Signal Details of RgstrHzntllbr\_An\_Actl

Feedback Sensor to RCCM

|  |  |  |
| --- | --- | --- |
| **ASIL** | | Choose an item. |
| **Init Default Value** | |  |
| **Encoding Type Name** | |  |
| 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 |  |
| Value 2 | … |
| … | … |
|  |  |
|  |  |
| **Unit** | |  |

Table: Signal Details of Feedback Sensor to RCCM

RgstrHzntllbr\_An\_Rq

Command Position of Horizontal Vanes for Righthand Inboard Register

|  |  |  |
| --- | --- | --- |
| **ASIL** | | Choose an item. |
| **Init Default Value** | |  |
| **Encoding Type Name** | |  |
| 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 |  |
| Value 2 | … |
| … | … |
|  |  |
|  |  |
| **Unit** | |  |

Table: Signal Details of RgstrHzntllbr\_An\_Rq

SensedRegPosRHOBvert

|  |  |  |
| --- | --- | --- |
| **ASIL** | | Choose an item. |
| **Init Default Value** | |  |
| **Encoding Type Name** | |  |
| 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 |  |
| Value 2 | … |
| … | … |
|  |  |
|  |  |
| **Unit** | |  |

Table: Signal Details of SensedRegPosRHOBvert

ActCmdLHOBHorizLin

Place holder for LIN signals

|  |  |  |
| --- | --- | --- |
| **ASIL** | | Choose an item. |
| **Init Default Value** | |  |
| **Encoding Type Name** | |  |
| 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 |  |
| Value 2 | … |
| … | … |
|  |  |
|  |  |
| **Unit** | |  |

Table: Signal Details of ActCmdLHOBHorizLin

SensFdbkLHOBVert

Sense left hand vertical outboard register vane positions and forward to RCCM

|  |  |  |
| --- | --- | --- |
| **ASIL** | | Choose an item. |
| **Init Default Value** | |  |
| **Encoding Type Name** | |  |
| 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 |  |
| Value 2 | … |
| … | … |
|  |  |
|  |  |
| **Unit** | |  |

Table: Signal Details of SensFdbkLHOBVert

SensedRegPosLHIBhoriz

|  |  |  |
| --- | --- | --- |
| **ASIL** | | Choose an item. |
| **Init Default Value** | |  |
| **Encoding Type Name** | |  |
| 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 |  |
| Value 2 | … |
| … | … |
|  |  |
|  |  |
| **Unit** | |  |

Table: Signal Details of SensedRegPosLHIBhoriz

ActCmdRHIBVert

Signal from RCCM to right hand inboard vertical vane register actuator

|  |  |  |
| --- | --- | --- |
| **ASIL** | | Choose an item. |
| **Init Default Value** | |  |
| **Encoding Type Name** | |  |
| 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 |  |
| Value 2 | … |
| … | … |
|  |  |
|  |  |
| **Unit** | |  |

Table: Signal Details of ActCmdRHIBVert

ActCmdLHIBVert

Signal from RCCM to left hand inboard vertica vane register actuator

|  |  |  |
| --- | --- | --- |
| **ASIL** | | Choose an item. |
| **Init Default Value** | |  |
| **Encoding Type Name** | |  |
| 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 |  |
| Value 2 | … |
| … | … |
|  |  |
|  |  |
| **Unit** | |  |

Table: Signal Details of ActCmdLHIBVert

ActCmdLHIBHorizLin

Place holder for LIN signals

|  |  |  |
| --- | --- | --- |
| **ASIL** | | Choose an item. |
| **Init Default Value** | |  |
| **Encoding Type Name** | |  |
| 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 |  |
| Value 2 | … |
| … | … |
|  |  |
|  |  |
| **Unit** | |  |

Table: Signal Details of ActCmdLHIBHorizLin

VehStat

Vehicle status signals used to store register positions when vehicle is on

|  |  |  |
| --- | --- | --- |
| **ASIL** | | Choose an item. |
| **Init Default Value** | |  |
| **Encoding Type Name** | |  |
| 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 |  |
| Value 2 | … |
| … | … |
|  |  |
|  |  |
| **Unit** | |  |

Table: Signal Details of VehStat

RgstrVertObl\_An\_Rq

Command Position of Vertical Vanes for Lefthand Outboard Register

|  |  |  |
| --- | --- | --- |
| **ASIL** | | Choose an item. |
| **Init Default Value** | |  |
| **Encoding Type Name** | |  |
| 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 |  |
| Value 2 | … |
| … | … |
|  |  |
|  |  |
| **Unit** | |  |

Table: Signal Details of RgstrVertObl\_An\_Rq

RgstrVertObr\_An\_Rq

Command Position of Vertical Vanes for Righthand Outboard Register

|  |  |  |
| --- | --- | --- |
| **ASIL** | | Choose an item. |
| **Init Default Value** | |  |
| **Encoding Type Name** | |  |
| 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 |  |
| Value 2 | … |
| … | … |
|  |  |
|  |  |
| **Unit** | |  |

Table: Signal Details of RgstrVertObr\_An\_Rq

SensedRegPosLHIBvert

|  |  |  |
| --- | --- | --- |
| **ASIL** | | Choose an item. |
| **Init Default Value** | |  |
| **Encoding Type Name** | |  |
| 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 |  |
| Value 2 | … |
| … | … |
|  |  |
|  |  |
| **Unit** | |  |

Table: Signal Details of SensedRegPosLHIBvert

SensFdbkLHIBVert

Sense left hand vertical inboard register vane positions and forward to RCCM

|  |  |  |
| --- | --- | --- |
| **ASIL** | | Choose an item. |
| **Init Default Value** | |  |
| **Encoding Type Name** | |  |
| 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 |  |
| Value 2 | … |
| … | … |
|  |  |
|  |  |
| **Unit** | |  |

Table: Signal Details of SensFdbkLHIBVert

Sensed Register Positon

|  |  |  |
| --- | --- | --- |
| **ASIL** | | Choose an item. |
| **Init Default Value** | |  |
| **Encoding Type Name** | |  |
| 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 |  |
| Value 2 | … |
| … | … |
|  |  |
|  |  |
| **Unit** | |  |

Table: Signal Details of Sensed Register Positon

SensFdbkRHOBVert

Sense right hand vertical inboard register vane positions and forward to RCCM

|  |  |  |
| --- | --- | --- |
| **ASIL** | | Choose an item. |
| **Init Default Value** | |  |
| **Encoding Type Name** | |  |
| 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 |  |
| Value 2 | … |
| … | … |
|  |  |
|  |  |
| **Unit** | |  |

Table: Signal Details of SensFdbkRHOBVert

Feedback HRD

Place holder for hardwire feedback from potentiometer to actuator

|  |  |  |
| --- | --- | --- |
| **ASIL** | | Choose an item. |
| **Init Default Value** | |  |
| **Encoding Type Name** | |  |
| 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 |  |
| Value 2 | … |
| … | … |
|  |  |
|  |  |
| **Unit** | |  |

Table: Signal Details of Feedback HRD

RgstrVertIbl\_An\_Actl

Aim Status of Lefthand Inboard Register Vertical Vanes

|  |  |  |
| --- | --- | --- |
| **ASIL** | | Choose an item. |
| **Init Default Value** | |  |
| **Encoding Type Name** | |  |
| 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 |  |
| Value 2 | … |
| … | … |
|  |  |
|  |  |
| **Unit** | |  |

Table: Signal Details of RgstrVertIbl\_An\_Actl

RgstrSetObl\_D\_Rq

Change Aim Command for Lefthand Outboard Register

|  |  |  |
| --- | --- | --- |
| **ASIL** | | Choose an item. |
| **Init Default Value** | |  |
| **Encoding Type Name** | |  |
| 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 |  |
| Value 2 | … |
| … | … |
|  |  |
|  |  |
| **Unit** | |  |

Table: Signal Details of RgstrSetObl\_D\_Rq

SensedRegPosLHOBvert

|  |  |  |
| --- | --- | --- |
| **ASIL** | | Choose an item. |
| **Init Default Value** | |  |
| **Encoding Type Name** | |  |
| 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 |  |
| Value 2 | … |
| … | … |
|  |  |
|  |  |
| **Unit** | |  |

Table: Signal Details of SensedRegPosLHOBvert

SensedRegPosRHIBhoriz

|  |  |  |
| --- | --- | --- |
| **ASIL** | | Choose an item. |
| **Init Default Value** | |  |
| **Encoding Type Name** | |  |
| 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 |  |
| Value 2 | … |
| … | … |
|  |  |
|  |  |
| **Unit** | |  |

Table: Signal Details of SensedRegPosRHIBhoriz

ActCmdRHIBHoriz

Signal from RCCM to right hand inboard horizontal vane register actuator

|  |  |  |
| --- | --- | --- |
| **ASIL** | | Choose an item. |
| **Init Default Value** | |  |
| **Encoding Type Name** | |  |
| 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 |  |
| Value 2 | … |
| … | … |
|  |  |
|  |  |
| **Unit** | |  |

Table: Signal Details of ActCmdRHIBHoriz

ActvButtnLeft\_D\_Rq

Subfeature Selection Commands for Left Side Registers

|  |  |  |
| --- | --- | --- |
| **ASIL** | | Choose an item. |
| **Init Default Value** | |  |
| **Encoding Type Name** | |  |
| 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 |  |
| Value 2 | … |
| … | … |
|  |  |
|  |  |
| **Unit** | |  |

Table: Signal Details of ActvButtnLeft\_D\_Rq

SensFdbkRHOBHoriz

Sense right hand horizontal outboard register vane positions and forward to RCCM

|  |  |  |
| --- | --- | --- |
| **ASIL** | | Choose an item. |
| **Init Default Value** | |  |
| **Encoding Type Name** | |  |
| 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 |  |
| Value 2 | … |
| … | … |
|  |  |
|  |  |
| **Unit** | |  |

Table: Signal Details of SensFdbkRHOBHoriz

RgstrSetIbl\_D\_Rq

Change Aim Command for Lefthand Inboard Register

|  |  |  |
| --- | --- | --- |
| **ASIL** | | Choose an item. |
| **Init Default Value** | |  |
| **Encoding Type Name** | |  |
| 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 |  |
| Value 2 | … |
| … | … |
|  |  |
|  |  |
| **Unit** | |  |

Table: Signal Details of RgstrSetIbl\_D\_Rq

Request HRD

Place holder for hardwire signals

|  |  |  |
| --- | --- | --- |
| **ASIL** | | Choose an item. |
| **Init Default Value** | |  |
| **Encoding Type Name** | |  |
| 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 |  |
| Value 2 | … |
| … | … |
|  |  |
|  |  |
| **Unit** | |  |

Table: Signal Details of Request HRD

RgstrHzntlObl\_An\_Actl

Aim Status of Lefthand Outboard Register Horizontal Vanes

|  |  |  |
| --- | --- | --- |
| **ASIL** | | Choose an item. |
| **Init Default Value** | |  |
| **Encoding Type Name** | |  |
| 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 |  |
| Value 2 | … |
| … | … |
|  |  |
|  |  |
| **Unit** | |  |

Table: Signal Details of RgstrHzntlObl\_An\_Actl

Clmt\_Button\_Stat4

Climate button signal

|  |  |  |
| --- | --- | --- |
| **ASIL** | | Choose an item. |
| **Init Default Value** | |  |
| **Encoding Type Name** | |  |
| 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 |  |
| Value 2 | … |
| … | … |
|  |  |
|  |  |
| **Unit** | |  |

Table: Signal Details of Clmt\_Button\_Stat4

ActvButtnRight\_D\_Rq

Subfeature Selection Commands for Right Side Registers

|  |  |  |
| --- | --- | --- |
| **ASIL** | | Choose an item. |
| **Init Default Value** | |  |
| **Encoding Type Name** | |  |
| 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 |  |
| Value 2 | … |
| … | … |
|  |  |
|  |  |
| **Unit** | |  |

Table: Signal Details of ActvButtnRight\_D\_Rq

APIM to RCCM ActvButtn

|  |  |  |
| --- | --- | --- |
| **ASIL** | | Choose an item. |
| **Init Default Value** | |  |
| **Encoding Type Name** | |  |
| 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 |  |
| Value 2 | … |
| … | … |
|  |  |
|  |  |
| **Unit** | |  |

Table: Signal Details of APIM to RCCM ActvButtn

RgstrHzntlObr\_An\_Rq

Command Position of Horizontal Vanes for Righthand Outboard Register

|  |  |  |
| --- | --- | --- |
| **ASIL** | | Choose an item. |
| **Init Default Value** | |  |
| **Encoding Type Name** | |  |
| 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 |  |
| Value 2 | … |
| … | … |
|  |  |
|  |  |
| **Unit** | |  |

Table: Signal Details of RgstrHzntlObr\_An\_Rq

RCCM to APIM

|  |  |  |
| --- | --- | --- |
| **ASIL** | | Choose an item. |
| **Init Default Value** | |  |
| **Encoding Type Name** | |  |
| 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 |  |
| Value 2 | … |
| … | … |
|  |  |
|  |  |
| **Unit** | |  |

Table: Signal Details of RCCM to APIM

SensedRegPosRHIBvert

|  |  |  |
| --- | --- | --- |
| **ASIL** | | Choose an item. |
| **Init Default Value** | |  |
| **Encoding Type Name** | |  |
| 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 |  |
| Value 2 | … |
| … | … |
|  |  |
|  |  |
| **Unit** | |  |

Table: Signal Details of SensedRegPosRHIBvert

RgstrVertlbl\_An\_Rq

Command Position of Vertical Vanes for Lefthand Inboard Register

|  |  |  |
| --- | --- | --- |
| **ASIL** | | Choose an item. |
| **Init Default Value** | |  |
| **Encoding Type Name** | |  |
| 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 |  |
| Value 2 | … |
| … | … |
|  |  |
|  |  |
| **Unit** | |  |

Table: Signal Details of RgstrVertlbl\_An\_Rq

ActvButtnRight\_D\_Stat

Subfeature Selection Status for Right Side Registers:

\*NonePressed

\*OnBodyLHS

\*OffBodyLHS

\*PresetOneLHS

\*PresetTwoLHS

\*Cycle\_8\_LHS

\*Cycle\_C\_LHS

\*Cycle\_O\_LHS

\*Cycle\_--\_LHS

\*Cycle\_I\_LHS

|  |  |  |
| --- | --- | --- |
| **ASIL** | | Choose an item. |
| **Init Default Value** | |  |
| **Encoding Type Name** | |  |
| 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 |  |
| Value 2 | … |
| … | … |
|  |  |
|  |  |
| **Unit** | |  |

Table: Signal Details of ActvButtnRight\_D\_Stat

SensFdbkRHIBVert

Sense right hand vertical inboard register vane positions and forward to RCCM

|  |  |  |
| --- | --- | --- |
| **ASIL** | | Choose an item. |
| **Init Default Value** | |  |
| **Encoding Type Name** | |  |
| 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 |  |
| Value 2 | … |
| … | … |
|  |  |
|  |  |
| **Unit** | |  |

Table: Signal Details of SensFdbkRHIBVert

ActCmdLHIBVertLin

Place holder for LIN signals

|  |  |  |
| --- | --- | --- |
| **ASIL** | | Choose an item. |
| **Init Default Value** | |  |
| **Encoding Type Name** | |  |
| 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 |  |
| Value 2 | … |
| … | … |
|  |  |
|  |  |
| **Unit** | |  |

Table: Signal Details of ActCmdLHIBVertLin

RgstrHzntlIbl\_An\_Rq

Command Position of Horizontal Vanes for Lefthand Inboard Register

|  |  |  |
| --- | --- | --- |
| **ASIL** | | Choose an item. |
| **Init Default Value** | |  |
| **Encoding Type Name** | |  |
| 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 |  |
| Value 2 | … |
| … | … |
|  |  |
|  |  |
| **Unit** | |  |

Table: Signal Details of RgstrHzntlIbl\_An\_Rq

RgstrSetIbr\_D\_Rq

Change Aim Command for Righthand Intboard Register

|  |  |  |
| --- | --- | --- |
| **ASIL** | | Choose an item. |
| **Init Default Value** | |  |
| **Encoding Type Name** | |  |
| 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 |  |
| Value 2 | … |
| … | … |
|  |  |
|  |  |
| **Unit** | |  |

Table: Signal Details of RgstrSetIbr\_D\_Rq

ActCmdLHOBHoriz

Signal from RCCM to left hand inboard horizontal vane register actuator

|  |  |  |
| --- | --- | --- |
| **ASIL** | | Choose an item. |
| **Init Default Value** | |  |
| **Encoding Type Name** | |  |
| 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 |  |
| Value 2 | … |
| … | … |
|  |  |
|  |  |
| **Unit** | |  |

Table: Signal Details of ActCmdLHOBHoriz

#### GSDB Signals

#### HW I/Os

#### Diagnostic Interfaces

##### DTCs

<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

### Technical Parameters

### Mappings

### Technical Interfaces

#### AIS Interfaces

##### Publisher Interfaces

##### Subscriber Interfaces

#### AUTOSAR Ports

### Messages/APIs

#### CAN Bus “<Bus Name>”

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

#### LIN Bus “<Bus Name>”

#### AUTOSAR Interfaces

#### SOA Service Contracts

<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

SensFdbkLHIBHoriz

Sense left hand horizontal inboard register vane positions and forward to RCCM

**Note:** An encoding is either discrete or continuous. Delete those fields, which are not needed.

|  |  |  |
| --- | --- | --- |
| **Value**  (Continuous Encoding) | Min Value |  |
| Max Value |  |
| Resolution |  |
| Offset |  |
| **Value**  (Discrete  Encoding) |  |  |
| **voltage** | voltage property |
| **Unit** | |  |

Table: Encoding Details of SensFdbkLHIBHoriz

RegisterPositionsSensorFeedback

Signal to provide register position feedback to actuator

**Note:** An encoding is either discrete or continuous. Delete those fields, which are not needed.

|  |  |  |
| --- | --- | --- |
| **Value**  (Continuous Encoding) | Min Value |  |
| Max Value |  |
| Resolution |  |
| Offset |  |
| **Value**  (Discrete  Encoding) |  |  |
| **SensorFeedbck\_LHOB\_Hor** |  |
| **SensorFeedbck\_LHOB\_Ver** |  |
| **SensorFeedbck\_LHIB\_Hor** |  |
| **SensorFeedbck\_LHIB\_Ver** |  |
| **SensorFeedbck\_RHIB\_Hor** |  |
| **SensorFeedbck\_RHIB\_Ver** |  |
| **SensorFeedbck\_RHOB\_Hor** |  |
| **SensorFeedbck\_RHOB\_Ver** |  |
| **Unit** | |  |

Table: Encoding Details of RegisterPositionsSensorFeedback

Register\_RHIB

Register location passenger inboard

**Note:** An encoding is either discrete or continuous. Delete those fields, which are not needed.

|  |  |  |
| --- | --- | --- |
| **Value**  (Continuous Encoding) | Min Value |  |
| Max Value |  |
| Resolution |  |
| Offset |  |
| **Value**  (Discrete  Encoding) |  |  |
| **NoChange** | Logical signal for register RHIB no change position |
| **ManualChange** | Logical signal for register RHIB manual change position |
| **FeatureChange** | Logical signal for register RHIB Feature change position |
| **Open** | Logical signal for register RHIB open position |
| **Close** | Logical signal for register RHIB close position |
| **Unit** | |  |

Table: Encoding Details of Register\_RHIB

ActiveButtonsLHS

Feedback active button options for the left side of the vehicle

**Note:** An encoding is either discrete or continuous. Delete those fields, which are not needed.

|  |  |  |
| --- | --- | --- |
| **Value**  (Continuous Encoding) | Min Value |  |
| Max Value |  |
| Resolution |  |
| Offset |  |
| **Value**  (Discrete  Encoding) |  |  |
| **NonePressed** | Logical signal active button LHS for none pressed |
| **OnBodyLHS** | Logical signal active button LHS for on body |
| **OffBodyLHS** | Logical signal active button LHS for off body |
| **PresetOneLHS** | Logical signal active button LHS for preset one |
| **PresetTwoLHS** | Logical signal active button LHS for preset two |
| **PresetOneSaveLHS** | Logical signal active button LHS for preset one save |
| **PresetTwoSaveLHS** | Logical signal active button LHS for preset two save |
| **Cycle\_8\_LHS** | Logical signal active button LHS for cycling figure 8 |
| **Cycle\_C\_LHS** | Logical signal active button LHS for close |
| **Cycle\_O\_LHS** | Logical signal active button LHS for open |
| **Cycle\_--\_LHS** | Logical signal active button LHS for cycling horizontal |
| **Cycle\_|\_LHS** | Logical signal active button LHS for cycling vertical |
| **Unit** | |  |

Table: Encoding Details of ActiveButtonsLHS

Request LIN

Place holder for LIN signals 707 program

**Note:** An encoding is either discrete or continuous. Delete those fields, which are not needed.

|  |  |  |
| --- | --- | --- |
| **Value**  (Continuous Encoding) | Min Value |  |
| Max Value |  |
| Resolution |  |
| Offset |  |
| **Value**  (Discrete  Encoding) |  |  |
| **Unit** | |  |

Table: Encoding Details of Request LIN

SensorFeedBack\_RHOB\_Vertical

Register sensor position feedback to HMI for right hand outboard register vertical voltage

**Note:** An encoding is either discrete or continuous. Delete those fields, which are not needed.

|  |  |  |
| --- | --- | --- |
| **Value**  (Continuous Encoding) | Min Value |  |
| Max Value |  |
| Resolution |  |
| Offset |  |
| **Value**  (Discrete  Encoding) |  |  |
| **Unit** | |  |

Table: Encoding Details of SensorFeedBack\_RHOB\_Vertical

SensFdbkRHIBHoriz

Sense right hand horizontal inboard register vane positions and forward to RCCM

**Note:** An encoding is either discrete or continuous. Delete those fields, which are not needed.

|  |  |  |
| --- | --- | --- |
| **Value**  (Continuous Encoding) | Min Value |  |
| Max Value |  |
| Resolution |  |
| Offset |  |
| **Value**  (Discrete  Encoding) |  |  |
| **voltage** | voltage property |
| **Unit** | |  |

Table: Encoding Details of SensFdbkRHIBHoriz

ActCmdLHIBHoriz

Signal from RCCM to left hand inboard horizontal vane register actuator

**Note:** An encoding is either discrete or continuous. Delete those fields, which are not needed.

|  |  |  |
| --- | --- | --- |
| **Value**  (Continuous Encoding) | Min Value |  |
| Max Value |  |
| Resolution |  |
| Offset |  |
| **Value**  (Discrete  Encoding) |  |  |
| **voltage** | voltage property |
| **Unit** | |  |

Table: Encoding Details of ActCmdLHIBHoriz

ActiveButtonsRHS

Feedback for active button options for the right side of the vehicle

**Note:** An encoding is either discrete or continuous. Delete those fields, which are not needed.

|  |  |  |
| --- | --- | --- |
| **Value**  (Continuous Encoding) | Min Value |  |
| Max Value |  |
| Resolution |  |
| Offset |  |
| **Value**  (Discrete  Encoding) |  |  |
| **NonePressed** | Logical signal active button RHS for none pressed |
| **OnBodyLHS** | Logical signal active button RHS for on body |
| **OffBodyLHS** | Logical signal active button RHS for off body |
| **PresetOneLHS** | Logical signal active button RHS for preset one |
| **PresetTwoLHS** | Logical signal active button RHS for preset two |
| **PresetOneSaveLHS** | Logical signal active button RHS for preset one save |
| **PresetTwoSaveLHS** | Logical signal active button RHS for preset two save |
| **Cycle\_8\_LHS** | Logical signal active button RHS for cycling figure 8 |
| **Cycle\_C\_LHS** | Logical signal active button RHS for close |
| **Cycle\_O\_LHS** | Logical signal active button RHS for open |
| **Cycle\_--\_LHS** | Logical signal active button RHS for cycling horizontal |
| **Cycle\_|\_LHS** | Logical signal active button RHS for cycling vertical |
| **Unit** | |  |

Table: Encoding Details of ActiveButtonsRHS

Feedback LIN

Place holder for LIN feedback signals 707 program

**Note:** An encoding is either discrete or continuous. Delete those fields, which are not needed.

|  |  |  |
| --- | --- | --- |
| **Value**  (Continuous Encoding) | Min Value |  |
| Max Value |  |
| Resolution |  |
| Offset |  |
| **Value**  (Discrete  Encoding) |  |  |
| **Unit** | |  |

Table: Encoding Details of Feedback LIN

ActCmdRHOBVert

Signal from RCCM to right hand outboard vertical vane register actuator

**Note:** An encoding is either discrete or continuous. Delete those fields, which are not needed.

|  |  |  |
| --- | --- | --- |
| **Value**  (Continuous Encoding) | Min Value |  |
| Max Value |  |
| Resolution |  |
| Offset |  |
| **Value**  (Discrete  Encoding) |  |  |
| **voltage** | voltage property |
| **Unit** | |  |

Table: Encoding Details of ActCmdRHOBVert

RgstrVertObl\_An\_Actl

Aim Status of Righthand Inboard Register Vertical Vanes

**Note:** An encoding is either discrete or continuous. Delete those fields, which are not needed.

|  |  |  |
| --- | --- | --- |
| **Value**  (Continuous Encoding) | Min Value |  |
| Max Value |  |
| Resolution |  |
| Offset |  |
| **Value**  (Discrete  Encoding) |  |  |
| **unnamed1** |  |
| **Unit** | |  |

Table: Encoding Details of RgstrVertObl\_An\_Actl

Register\_RHOB\_Horizontal

Register airflow direction up or down

**Note:** An encoding is either discrete or continuous. Delete those fields, which are not needed.

|  |  |  |
| --- | --- | --- |
| **Value**  (Continuous Encoding) | Min Value |  |
| Max Value |  |
| Resolution |  |
| Offset |  |
| **Value**  (Discrete  Encoding) |  |  |
| **Percentage** | Logical signal for register RHOB horizontal percentage change position horizontal |
| **Unit** | |  |

Table: Encoding Details of Register\_RHOB\_Horizontal

RgstrHzntlObl\_An\_Rq

Command Position of Horizontal Vanes for Righthand Outboard Register

**Note:** An encoding is either discrete or continuous. Delete those fields, which are not needed.

|  |  |  |
| --- | --- | --- |
| **Value**  (Continuous Encoding) | Min Value |  |
| Max Value |  |
| Resolution |  |
| Offset |  |
| **Value**  (Discrete  Encoding) |  |  |
| **Deg** |  |
| **Unit** | |  |

Table: Encoding Details of RgstrHzntlObl\_An\_Rq

RgstrHzntllbr\_An\_Actl

Aim Status of Righthand Inboard Register Horizontal Vanes

**Note:** An encoding is either discrete or continuous. Delete those fields, which are not needed.

|  |  |  |
| --- | --- | --- |
| **Value**  (Continuous Encoding) | Min Value |  |
| Max Value |  |
| Resolution |  |
| Offset |  |
| **Value**  (Discrete  Encoding) |  |  |
| **Unit** | |  |

Table: Encoding Details of RgstrHzntllbr\_An\_Actl

SensedRegPosRHOBvert

**Note:** An encoding is either discrete or continuous. Delete those fields, which are not needed.

|  |  |  |
| --- | --- | --- |
| **Value**  (Continuous Encoding) | Min Value |  |
| Max Value |  |
| Resolution |  |
| Offset |  |
| **Value**  (Discrete  Encoding) |  |  |
| **Unit** | |  |

Table: Encoding Details of SensedRegPosRHOBvert

Feedback Sensor to RCCM

**Note:** An encoding is either discrete or continuous. Delete those fields, which are not needed.

|  |  |  |
| --- | --- | --- |
| **Value**  (Continuous Encoding) | Min Value |  |
| Max Value |  |
| Resolution |  |
| Offset |  |
| **Value**  (Discrete  Encoding) |  |  |
| **Voltage** |  |
| **Unit** | |  |

Table: Encoding Details of Feedback Sensor to RCCM

RgstrHzntllbr\_An\_Rq

Command Position of Horizontal Vanes for Righthand Inboard Register

**Note:** An encoding is either discrete or continuous. Delete those fields, which are not needed.

|  |  |  |
| --- | --- | --- |
| **Value**  (Continuous Encoding) | Min Value |  |
| Max Value |  |
| Resolution |  |
| Offset |  |
| **Value**  (Discrete  Encoding) |  |  |
| **Unit** | |  |

Table: Encoding Details of RgstrHzntllbr\_An\_Rq

Sensed\_Actuator\_Position

Signal Not Used

**Note:** An encoding is either discrete or continuous. Delete those fields, which are not needed.

|  |  |  |
| --- | --- | --- |
| **Value**  (Continuous Encoding) | Min Value |  |
| Max Value |  |
| Resolution |  |
| Offset |  |
| **Value**  (Discrete  Encoding) |  |  |
| **Unit** | |  |

Table: Encoding Details of Sensed\_Actuator\_Position

ActCmdLHOBHorizLin

Place holder for LIN signals

**Note:** An encoding is either discrete or continuous. Delete those fields, which are not needed.

|  |  |  |
| --- | --- | --- |
| **Value**  (Continuous Encoding) | Min Value |  |
| Max Value |  |
| Resolution |  |
| Offset |  |
| **Value**  (Discrete  Encoding) |  |  |
| **Unit** | |  |

Table: Encoding Details of ActCmdLHOBHorizLin

OpenCloseRequest

Move request to close or open register vertical vanes

**Note:** An encoding is either discrete or continuous. Delete those fields, which are not needed.

|  |  |  |
| --- | --- | --- |
| **Value**  (Continuous Encoding) | Min Value |  |
| Max Value |  |
| Resolution |  |
| Offset |  |
| **Value**  (Discrete  Encoding) |  |  |
| **NoChange** |  |
| **Open** |  |
| **Close** |  |
| **Unit** | |  |

Table: Encoding Details of OpenCloseRequest

SensFdbkLHOBVert

Sense left hand vertical outboard register vane positions and forward to RCCM

**Note:** An encoding is either discrete or continuous. Delete those fields, which are not needed.

|  |  |  |
| --- | --- | --- |
| **Value**  (Continuous Encoding) | Min Value |  |
| Max Value |  |
| Resolution |  |
| Offset |  |
| **Value**  (Discrete  Encoding) |  |  |
| **voltage** | voltage property |
| **Unit** | |  |

Table: Encoding Details of SensFdbkLHOBVert

EM\_DesiredPositionsRequest

Manual move register requests

**Note:** An encoding is either discrete or continuous. Delete those fields, which are not needed.

|  |  |  |
| --- | --- | --- |
| **Value**  (Continuous Encoding) | Min Value |  |
| Max Value |  |
| Resolution |  |
| Offset |  |
| **Value**  (Discrete  Encoding) |  |  |
| **register\_LHOB\_Horz** |  |
| **register\_LHOB\_Vert** |  |
| **register\_LHIB\_Vert** |  |
| **register\_LHIB\_Horz** |  |
| **register\_RHOB\_Horz** |  |
| **register\_RHOB\_Vert** |  |
| **register\_RHIB\_Vert** |  |
| **register\_RHIB\_Horz** |  |
| **requesterType** |  |
| **Unit** | |  |

Table: Encoding Details of EM\_DesiredPositionsRequest

ActCmdLHIBHorizLin

Place holder for LIN signals

**Note:** An encoding is either discrete or continuous. Delete those fields, which are not needed.

|  |  |  |
| --- | --- | --- |
| **Value**  (Continuous Encoding) | Min Value |  |
| Max Value |  |
| Resolution |  |
| Offset |  |
| **Value**  (Discrete  Encoding) |  |  |
| **Unit** | |  |

Table: Encoding Details of ActCmdLHIBHorizLin

SensorFeedBack\_LHIB\_Vertical

Register sensor position feedback to HMI for Left hand inboard register voltage

**Note:** An encoding is either discrete or continuous. Delete those fields, which are not needed.

|  |  |  |
| --- | --- | --- |
| **Value**  (Continuous Encoding) | Min Value |  |
| Max Value |  |
| Resolution |  |
| Offset |  |
| **Value**  (Discrete  Encoding) |  |  |
| **Unit** | |  |

Table: Encoding Details of SensorFeedBack\_LHIB\_Vertical

Register\_RHIB\_Horizontal

Register airflow direction up or down

**Note:** An encoding is either discrete or continuous. Delete those fields, which are not needed.

|  |  |  |
| --- | --- | --- |
| **Value**  (Continuous Encoding) | Min Value |  |
| Max Value |  |
| Resolution |  |
| Offset |  |
| **Value**  (Discrete  Encoding) |  |  |
| **Percentage** | Logical signal for register RHIB horizontal percentage change position horizontal |
| **Unit** | |  |

Table: Encoding Details of Register\_RHIB\_Horizontal

ActCmdLHIBVert

Signal from RCCM to left hand inboard vertica vane register actuator

**Note:** An encoding is either discrete or continuous. Delete those fields, which are not needed.

|  |  |  |
| --- | --- | --- |
| **Value**  (Continuous Encoding) | Min Value |  |
| Max Value |  |
| Resolution |  |
| Offset |  |
| **Value**  (Discrete  Encoding) |  |  |
| **voltage** | voltage property |
| **Unit** | |  |

Table: Encoding Details of ActCmdLHIBVert

HMIFeedback\_RHIB\_Vertical

Register position feedback to HMI

**Note:** An encoding is either discrete or continuous. Delete those fields, which are not needed.

|  |  |  |
| --- | --- | --- |
| **Value**  (Continuous Encoding) | Min Value |  |
| Max Value |  |
| Resolution |  |
| Offset |  |
| **Value**  (Discrete  Encoding) |  |  |
| **Unit** | |  |

Table: Encoding Details of HMIFeedback\_RHIB\_Vertical

SensorFeedback\_RHIB\_Horizontal

Register sensor position feedback to HMI for right hand inboard register horizontal voltage

**Note:** An encoding is either discrete or continuous. Delete those fields, which are not needed.

|  |  |  |
| --- | --- | --- |
| **Value**  (Continuous Encoding) | Min Value |  |
| Max Value |  |
| Resolution |  |
| Offset |  |
| **Value**  (Discrete  Encoding) |  |  |
| **Unit** | |  |

Table: Encoding Details of SensorFeedback\_RHIB\_Horizontal

SensedRegPosLHIBvert

**Note:** An encoding is either discrete or continuous. Delete those fields, which are not needed.

|  |  |  |
| --- | --- | --- |
| **Value**  (Continuous Encoding) | Min Value |  |
| Max Value |  |
| Resolution |  |
| Offset |  |
| **Value**  (Discrete  Encoding) |  |  |
| **Unit** | |  |

Table: Encoding Details of SensedRegPosLHIBvert

RgstrVertObr\_An\_Rq

Command Position of Vertical Vanes for Righthand Outboard Register

**Note:** An encoding is either discrete or continuous. Delete those fields, which are not needed.

|  |  |  |
| --- | --- | --- |
| **Value**  (Continuous Encoding) | Min Value |  |
| Max Value |  |
| Resolution |  |
| Offset |  |
| **Value**  (Discrete  Encoding) |  |  |
| **unnamed1** |  |
| **Unit** | |  |

Table: Encoding Details of RgstrVertObr\_An\_Rq

SensFdbkLHIBVert

Sense left hand vertical inboard register vane positions and forward to RCCM

**Note:** An encoding is either discrete or continuous. Delete those fields, which are not needed.

|  |  |  |
| --- | --- | --- |
| **Value**  (Continuous Encoding) | Min Value |  |
| Max Value |  |
| Resolution |  |
| Offset |  |
| **Value**  (Discrete  Encoding) |  |  |
| **voltage** | voltage property |
| **Unit** | |  |

Table: Encoding Details of SensFdbkLHIBVert

Sensed Register Positon

**Note:** An encoding is either discrete or continuous. Delete those fields, which are not needed.

|  |  |  |
| --- | --- | --- |
| **Value**  (Continuous Encoding) | Min Value |  |
| Max Value |  |
| Resolution |  |
| Offset |  |
| **Value**  (Discrete  Encoding) |  |  |
| **Sensed Register Pos** |  |
| **Unit** | |  |

Table: Encoding Details of Sensed Register Positon

SensFdbkRHOBVert

Sense right hand vertical inboard register vane positions and forward to RCCM

**Note:** An encoding is either discrete or continuous. Delete those fields, which are not needed.

|  |  |  |
| --- | --- | --- |
| **Value**  (Continuous Encoding) | Min Value |  |
| Max Value |  |
| Resolution |  |
| Offset |  |
| **Value**  (Discrete  Encoding) |  |  |
| **voltage** | voltage property |
| **Unit** | |  |

Table: Encoding Details of SensFdbkRHOBVert

SensorFeedback\_LHIB\_Horizontal

Register sensor position feedback to HMI for left hand inboard register voltage

**Note:** An encoding is either discrete or continuous. Delete those fields, which are not needed.

|  |  |  |
| --- | --- | --- |
| **Value**  (Continuous Encoding) | Min Value |  |
| Max Value |  |
| Resolution |  |
| Offset |  |
| **Value**  (Discrete  Encoding) |  |  |
| **Unit** | |  |

Table: Encoding Details of SensorFeedback\_LHIB\_Horizontal

userInput

User input from HMI screen display

**Note:** An encoding is either discrete or continuous. Delete those fields, which are not needed.

|  |  |  |
| --- | --- | --- |
| **Value**  (Continuous Encoding) | Min Value |  |
| Max Value |  |
| Resolution |  |
| Offset |  |
| **Value**  (Discrete  Encoding) |  |  |
| **HMI Touch Input** | Logical signal that provides user input from HMI Touch screen |
| **Unit** | |  |

Table: Encoding Details of userInput

RgstrVertIbl\_An\_Actl

Aim Status of Lefthand Inboard Register Vertical Vanes

**Note:** An encoding is either discrete or continuous. Delete those fields, which are not needed.

|  |  |  |
| --- | --- | --- |
| **Value**  (Continuous Encoding) | Min Value |  |
| Max Value |  |
| Resolution |  |
| Offset |  |
| **Value**  (Discrete  Encoding) |  |  |
| **unnamed1** |  |
| **Unit** | |  |

Table: Encoding Details of RgstrVertIbl\_An\_Actl

Feedback HRD

Place holder for hardwire feedback from potentiometer to actuator

**Note:** An encoding is either discrete or continuous. Delete those fields, which are not needed.

|  |  |  |
| --- | --- | --- |
| **Value**  (Continuous Encoding) | Min Value |  |
| Max Value |  |
| Resolution |  |
| Offset |  |
| **Value**  (Discrete  Encoding) |  |  |
| **Unit** | |  |

Table: Encoding Details of Feedback HRD

RgstrSetObl\_D\_Rq

Change Aim Command for Lefthand Outboard Register

**Note:** An encoding is either discrete or continuous. Delete those fields, which are not needed.

|  |  |  |
| --- | --- | --- |
| **Value**  (Continuous Encoding) | Min Value |  |
| Max Value |  |
| Resolution |  |
| Offset |  |
| **Value**  (Discrete  Encoding) |  |  |
| **0x0** |  |
| **0x1** |  |
| **0x2** |  |
| **0x3** |  |
| **0x4** |  |
| **0x5** |  |
| **0x6** |  |
| **0x7** |  |
| **Unit** | |  |

Table: Encoding Details of RgstrSetObl\_D\_Rq

ActCmdRHIBHoriz

Signal from RCCM to right hand inboard horizontal vane register actuator

**Note:** An encoding is either discrete or continuous. Delete those fields, which are not needed.

|  |  |  |
| --- | --- | --- |
| **Value**  (Continuous Encoding) | Min Value |  |
| Max Value |  |
| Resolution |  |
| Offset |  |
| **Value**  (Discrete  Encoding) |  |  |
| **voltage** | voltage property |
| **Unit** | |  |

Table: Encoding Details of ActCmdRHIBHoriz

Register\_LHIB\_Horizontal

Register up or down airflow direction

**Note:** An encoding is either discrete or continuous. Delete those fields, which are not needed.

|  |  |  |
| --- | --- | --- |
| **Value**  (Continuous Encoding) | Min Value |  |
| Max Value |  |
| Resolution |  |
| Offset |  |
| **Value**  (Discrete  Encoding) |  |  |
| **Percentage** | Logical signal for register LHIB horizontal percentage change position horizontal |
| **Unit** | |  |

Table: Encoding Details of Register\_LHIB\_Horizontal

SensFdbkRHOBHoriz

Sense right hand horizontal outboard register vane positions and forward to RCCM

**Note:** An encoding is either discrete or continuous. Delete those fields, which are not needed.

|  |  |  |
| --- | --- | --- |
| **Value**  (Continuous Encoding) | Min Value |  |
| Max Value |  |
| Resolution |  |
| Offset |  |
| **Value**  (Discrete  Encoding) |  |  |
| **voltage** | voltage property |
| **Unit** | |  |

Table: Encoding Details of SensFdbkRHOBHoriz

HMIFeedback\_LHIB\_Vertical

Register position feedback to HMI

**Note:** An encoding is either discrete or continuous. Delete those fields, which are not needed.

|  |  |  |
| --- | --- | --- |
| **Value**  (Continuous Encoding) | Min Value |  |
| Max Value |  |
| Resolution |  |
| Offset |  |
| **Value**  (Discrete  Encoding) |  |  |
| **Unit** | |  |

Table: Encoding Details of HMIFeedback\_LHIB\_Vertical

APIM to RCCM ActvButtn

**Note:** An encoding is either discrete or continuous. Delete those fields, which are not needed.

|  |  |  |
| --- | --- | --- |
| **Value**  (Continuous Encoding) | Min Value |  |
| Max Value |  |
| Resolution |  |
| Offset |  |
| **Value**  (Discrete  Encoding) |  |  |
| **ActvButtnLeft\_D\_Rq** |  |
| **ActButtnRight\_D\_Rq** |  |
| **Unit** | |  |

Table: Encoding Details of APIM to RCCM ActvButtn

HMIFeedback\_RHOB

Register position feedback to HMI

**Note:** An encoding is either discrete or continuous. Delete those fields, which are not needed.

|  |  |  |
| --- | --- | --- |
| **Value**  (Continuous Encoding) | Min Value |  |
| Max Value |  |
| Resolution |  |
| Offset |  |
| **Value**  (Discrete  Encoding) |  |  |
| **Position** | Logical signal HMI feedback RHOB position |
| **Close** | Logical signal HMI feedback RHOB close |
| **Unit** | |  |

Table: Encoding Details of HMIFeedback\_RHOB

SensedRegPosRHIBvert

**Note:** An encoding is either discrete or continuous. Delete those fields, which are not needed.

|  |  |  |
| --- | --- | --- |
| **Value**  (Continuous Encoding) | Min Value |  |
| Max Value |  |
| Resolution |  |
| Offset |  |
| **Value**  (Discrete  Encoding) |  |  |
| **Unit** | |  |

Table: Encoding Details of SensedRegPosRHIBvert

RgstrVertlbl\_An\_Rq

Command Position of Vertical Vanes for Lefthand Inboard Register

**Note:** An encoding is either discrete or continuous. Delete those fields, which are not needed.

|  |  |  |
| --- | --- | --- |
| **Value**  (Continuous Encoding) | Min Value |  |
| Max Value |  |
| Resolution |  |
| Offset |  |
| **Value**  (Discrete  Encoding) |  |  |
| **unnamed1** |  |
| **Unit** | |  |

Table: Encoding Details of RgstrVertlbl\_An\_Rq

SensFdbkRHIBVert

Sense right hand vertical inboard register vane positions and forward to RCCM

**Note:** An encoding is either discrete or continuous. Delete those fields, which are not needed.

|  |  |  |
| --- | --- | --- |
| **Value**  (Continuous Encoding) | Min Value |  |
| Max Value |  |
| Resolution |  |
| Offset |  |
| **Value**  (Discrete  Encoding) |  |  |
| **voltage** | voltage property |
| **Unit** | |  |

Table: Encoding Details of SensFdbkRHIBVert

ActCmdLHIBVertLin

Place holder for LIN signals

**Note:** An encoding is either discrete or continuous. Delete those fields, which are not needed.

|  |  |  |
| --- | --- | --- |
| **Value**  (Continuous Encoding) | Min Value |  |
| Max Value |  |
| Resolution |  |
| Offset |  |
| **Value**  (Discrete  Encoding) |  |  |
| **Unit** | |  |

Table: Encoding Details of ActCmdLHIBVertLin

Active\_ButtonFeedbck

**Note:** An encoding is either discrete or continuous. Delete those fields, which are not needed.

|  |  |  |
| --- | --- | --- |
| **Value**  (Continuous Encoding) | Min Value |  |
| Max Value |  |
| Resolution |  |
| Offset |  |
| **Value**  (Discrete  Encoding) |  |  |
| **Active Buttons LHS** |  |
| **Active Buttons RHS** |  |
| **Unit** | |  |

Table: Encoding Details of Active\_ButtonFeedbck

ActuatorCommand\_LHIB\_Horizontal

Left hand inboard register voltage for up down movement

**Note:** An encoding is either discrete or continuous. Delete those fields, which are not needed.

|  |  |  |
| --- | --- | --- |
| **Value**  (Continuous Encoding) | Min Value |  |
| Max Value |  |
| Resolution |  |
| Offset |  |
| **Value**  (Discrete  Encoding) |  |  |
| **Voltage** | Logical signal for LHIB horizontal percentage change |
| **Unit** | |  |

Table: Encoding Details of ActuatorCommand\_LHIB\_Horizontal

Register\_LHOB\_Horizontal

Register up or down airflow direction

**Note:** An encoding is either discrete or continuous. Delete those fields, which are not needed.

|  |  |  |
| --- | --- | --- |
| **Value**  (Continuous Encoding) | Min Value |  |
| Max Value |  |
| Resolution |  |
| Offset |  |
| **Value**  (Discrete  Encoding) |  |  |
| **Degrees** | Logical signal for register LHOB horizontal percentage change position horizontal |
| **Unit** | |  |

Table: Encoding Details of Register\_LHOB\_Horizontal

RgstrSetIbr\_D\_Rq

Change Aim Command for Righthand Intboard Register

**Note:** An encoding is either discrete or continuous. Delete those fields, which are not needed.

|  |  |  |
| --- | --- | --- |
| **Value**  (Continuous Encoding) | Min Value |  |
| Max Value |  |
| Resolution |  |
| Offset |  |
| **Value**  (Discrete  Encoding) |  |  |
| **0x0** |  |
| **0x1** |  |
| **0x2** |  |
| **0x3** |  |
| **0x4** |  |
| **0x5** |  |
| **0x6** |  |
| **0x7** |  |
| **Unit** | |  |

Table: Encoding Details of RgstrSetIbr\_D\_Rq

ActuatorCommand\_LHOB\_Horizontal

Left hand inboard register voltage for up down movement

**Note:** An encoding is either discrete or continuous. Delete those fields, which are not needed.

|  |  |  |
| --- | --- | --- |
| **Value**  (Continuous Encoding) | Min Value |  |
| Max Value |  |
| Resolution |  |
| Offset |  |
| **Value**  (Discrete  Encoding) |  |  |
| **Voltage** | Logical signal for LHOB horizontal percentage change |
| **Unit** | |  |

Table: Encoding Details of ActuatorCommand\_LHOB\_Horizontal

ActCmdLHOBHoriz

Signal from RCCM to left hand inboard horizontal vane register actuator

**Note:** An encoding is either discrete or continuous. Delete those fields, which are not needed.

|  |  |  |
| --- | --- | --- |
| **Value**  (Continuous Encoding) | Min Value |  |
| Max Value |  |
| Resolution |  |
| Offset |  |
| **Value**  (Discrete  Encoding) |  |  |
| **voltage** | voltage property |
| **Unit** | |  |

Table: Encoding Details of ActCmdLHOBHoriz

RgstrVertllbr\_An\_Actl

Aim Status of Righthand Inboard Register Vertical Vanes

**Note:** An encoding is either discrete or continuous. Delete those fields, which are not needed.

|  |  |  |
| --- | --- | --- |
| **Value**  (Continuous Encoding) | Min Value |  |
| Max Value |  |
| Resolution |  |
| Offset |  |
| **Value**  (Discrete  Encoding) |  |  |
| **RotationAngle** |  |
| **SignalRange** | Signal Range 8 bit (0-255) for angle |
| **Unit** | |  |

Table: Encoding Details of RgstrVertllbr\_An\_Actl

Register\_LHOB

Register location driver outboard

**Note:** An encoding is either discrete or continuous. Delete those fields, which are not needed.

|  |  |  |
| --- | --- | --- |
| **Value**  (Continuous Encoding) | Min Value |  |
| Max Value |  |
| Resolution |  |
| Offset |  |
| **Value**  (Discrete  Encoding) |  |  |
| **Register\_LHOB\_Horizontal** |  |
| **Register\_LHOB\_Vertical** |  |
| **NoChange** | Logical signal for register LHOB no changel position |
| **Open** | Logical signal for register LHOB open position |
| **Close** | Logical signal for register LHOB close position |
| **Unit** | |  |

Table: Encoding Details of Register\_LHOB

RgstrSetObr\_D\_Rq

Change Aim Command for Righthand Intboard Register

**Note:** An encoding is either discrete or continuous. Delete those fields, which are not needed.

|  |  |  |
| --- | --- | --- |
| **Value**  (Continuous Encoding) | Min Value |  |
| Max Value |  |
| Resolution |  |
| Offset |  |
| **Value**  (Discrete  Encoding) |  |  |
| **0x0** |  |
| **0x1** |  |
| **0x2** |  |
| **0x3** |  |
| **0x4** |  |
| **0x5** |  |
| **0x6** |  |
| **0x7** |  |
| **Unit** | |  |

Table: Encoding Details of RgstrSetObr\_D\_Rq

SensorFeedBack\_LHOB\_Vertical

Register sensor position feedback to HMI for Left hand inboard register vertical voltage

**Note:** An encoding is either discrete or continuous. Delete those fields, which are not needed.

|  |  |  |
| --- | --- | --- |
| **Value**  (Continuous Encoding) | Min Value |  |
| Max Value |  |
| Resolution |  |
| Offset |  |
| **Value**  (Discrete  Encoding) |  |  |
| **Unit** | |  |

Table: Encoding Details of SensorFeedBack\_LHOB\_Vertical

APIM to RCCM Rgstr Set

**Note:** An encoding is either discrete or continuous. Delete those fields, which are not needed.

|  |  |  |
| --- | --- | --- |
| **Value**  (Continuous Encoding) | Min Value |  |
| Max Value |  |
| Resolution |  |
| Offset |  |
| **Value**  (Discrete  Encoding) |  |  |
| **Register Set IBR** |  |
| **Register Set OBR** |  |
| **Register Set OBL** |  |
| **Register Set IBL** |  |
| **Unit** | |  |

Table: Encoding Details of APIM to RCCM Rgstr Set

RCCM to Actuator LIN

Signals sent from RCCM to register actuator

**Note:** An encoding is either discrete or continuous. Delete those fields, which are not needed.

|  |  |  |
| --- | --- | --- |
| **Value**  (Continuous Encoding) | Min Value |  |
| Max Value |  |
| Resolution |  |
| Offset |  |
| **Value**  (Discrete  Encoding) |  |  |
| **ActCmdLHOBHoriz** | Implementation signal for ActCmdLHOBHoriz |
| **ActCmdLHOBVert** | Implementation signal for ActCmdLHOBVert |
| **ActCmdLHIBHoriz** | Implementation signal for ActCmdLHIBHoriz |
| **ActCmdLHIBVert** | Implementation signal for ActCmdLHIBVert |
| **ActCmdRHIBHoriz** | Implementation signal for ActCmdRHIBHoriz |
| **ActCmdRHIBVert** | Implementation signal for ActCmdRHIBVert |
| **ActCmdRHOBHoriz** | Implementation signal for ActCmdRHOBHoriz |
| **ActCmdRHOBVert** | Implementation signal for ActCmdRHOBVert |
| **LIN** |  |
| **Hardwire** |  |
| **Unit** | |  |

Table: Encoding Details of RCCM to Actuator LIN

ActuatorCommand\_RHOB\_Vertical

Right hand outboard register voltage for left right movement

**Note:** An encoding is either discrete or continuous. Delete those fields, which are not needed.

|  |  |  |
| --- | --- | --- |
| **Value**  (Continuous Encoding) | Min Value |  |
| Max Value |  |
| Resolution |  |
| Offset |  |
| **Value**  (Discrete  Encoding) |  |  |
| **Voltage** | Logical signal for RHOB vertical percentage change |
| **Unit** | |  |

Table: Encoding Details of ActuatorCommand\_RHOB\_Vertical

ActvButtnLeft\_D\_Stat

Subfeature Selection Status for Left Side Registers:

\*NonePressed

\*OnBodyLHS

\*OffBodyLHS

\*PresetOneLHS

\*PresetTwoLHS

\*Cycle\_8\_LHS

\*Cycle\_C\_LHS

\*Cycle\_O\_LHS

\*Cycle\_--\_LHS

\*Cycle\_I\_LHS

**Note:** An encoding is either discrete or continuous. Delete those fields, which are not needed.

|  |  |  |
| --- | --- | --- |
| **Value**  (Continuous Encoding) | Min Value |  |
| Max Value |  |
| Resolution |  |
| Offset |  |
| **Value**  (Discrete  Encoding) |  |  |
| **0x0 nonePressed** | Button pressed discrete signal for none pressed LHS selection |
| **0x1 onBodyLHS** | Button pressed discrete signal for on body LHS selection |
| **0x2 offBodyLHS** | Button pressed discrete signal for off body LHS selection |
| **0x3 presetOneLHS** | Button pressed discrete signal for preset one LHS selection |
| **0x4 presetTwoLHS** | Button pressed discrete signal for preset two LHS selection |
| **0x6 presetTwoSaveLHS** | Button pressed discrete signal for preset two saved LHS selection |
| **0x5 presetOneSaveLHS** | Button pressed discrete signal for preset one saved LHS selection |
| **0x5 cycle\_8\_LHS** | Button pressed discrete signal for cycling figure 8 vanes LHS selection |
| **0x6 cycle\_C\_LHS** | Button pressed discrete signal for closed vanes LHS selection |
| **0x7 cycle\_O\_LHS** | Button pressed discrete signal for open vanes LHS selection |
| **0x8 cycle\_--\_LHS** | Button pressed discrete signal for cycling horizontal vanes LHS selection |
| **0x9 cycle\_|\_LHS** | Button pressed discrete signal for cycling vertical vanes LHS selection |
| **Unit** | |  |

Table: Encoding Details of ActvButtnLeft\_D\_Stat

RgstrVertlbr\_An\_Rq

Command Position of Vertical Vanes for Righthand Inboard Register

**Note:** An encoding is either discrete or continuous. Delete those fields, which are not needed.

|  |  |  |
| --- | --- | --- |
| **Value**  (Continuous Encoding) | Min Value |  |
| Max Value |  |
| Resolution |  |
| Offset |  |
| **Value**  (Discrete  Encoding) |  |  |
| **unnamed1** |  |
| **Unit** | |  |

Table: Encoding Details of RgstrVertlbr\_An\_Rq

HMIFeedback\_LHOB

Register position feedback to HMI

**Note:** An encoding is either discrete or continuous. Delete those fields, which are not needed.

|  |  |  |
| --- | --- | --- |
| **Value**  (Continuous Encoding) | Min Value |  |
| Max Value |  |
| Resolution |  |
| Offset |  |
| **Value**  (Discrete  Encoding) |  |  |
| **Position** | Logical signal HMI feedback LHOB position |
| **Close** | Logical signal HMI feedback LHOB close |
| **Unit** | |  |

Table: Encoding Details of HMIFeedback\_LHOB

Register\_RHOB

Register location passenger outboard

**Note:** An encoding is either discrete or continuous. Delete those fields, which are not needed.

|  |  |  |
| --- | --- | --- |
| **Value**  (Continuous Encoding) | Min Value |  |
| Max Value |  |
| Resolution |  |
| Offset |  |
| **Value**  (Discrete  Encoding) |  |  |
| **NoChange** | Logical signal for register RHOB no change position |
| **ManualChange** | Logical signal for register RHOB manual change position |
| **FeatureChange** | Logical signal for register RHOB feature change position |
| **Open** | Logical signal for register RHOB open position |
| **Close** | Logical signal for register RHOB close position |
| **Unit** | |  |

Table: Encoding Details of Register\_RHOB

RgstrHzntlIbl\_An\_Actl

Aim Status of Lefthand Inboard Register Horizontal Vanes

**Note:** An encoding is either discrete or continuous. Delete those fields, which are not needed.

|  |  |  |
| --- | --- | --- |
| **Value**  (Continuous Encoding) | Min Value |  |
| Max Value |  |
| Resolution |  |
| Offset |  |
| **Value**  (Discrete  Encoding) |  |  |
| **Unit** | |  |

Table: Encoding Details of RgstrHzntlIbl\_An\_Actl

RCCM to Actuator HRD

**Note:** An encoding is either discrete or continuous. Delete those fields, which are not needed.

|  |  |  |
| --- | --- | --- |
| **Value**  (Continuous Encoding) | Min Value |  |
| Max Value |  |
| Resolution |  |
| Offset |  |
| **Value**  (Discrete  Encoding) |  |  |
| **ActCmdLHOBHoriz** |  |
| **Unit** | |  |

Table: Encoding Details of RCCM to Actuator HRD

Actuate Register Command

Signal requesting actuation of register position changes

**Note:** An encoding is either discrete or continuous. Delete those fields, which are not needed.

|  |  |  |
| --- | --- | --- |
| **Value**  (Continuous Encoding) | Min Value |  |
| Max Value |  |
| Resolution |  |
| Offset |  |
| **Value**  (Discrete  Encoding) |  |  |
| **ActuatorCmnd\_LHOB\_Hor** |  |
| **actuatorCommand\_LHOB\_Horizontal** |  |
| **ActuatorCmnd\_LHOB\_Ver** |  |
| **ActuatorCmnd\_LHIB\_Hor** |  |
| **ActuatorCmnd\_LHIB\_Ver** |  |
| **ActuatorCmnd\_RHIB\_Hor** |  |
| **ActuatorCmnd\_RHIB\_Ver** |  |
| **ActuatorCmnd\_RHOB\_Hor** |  |
| **ActuatorCmnd\_RHOB\_Ver** |  |
| **Unit** | |  |

Table: Encoding Details of Actuate Register Command

ActuatorCommand\_RHIB\_Horizontal

Right hand inboard register voltage for up down movement

**Note:** An encoding is either discrete or continuous. Delete those fields, which are not needed.

|  |  |  |
| --- | --- | --- |
| **Value**  (Continuous Encoding) | Min Value |  |
| Max Value |  |
| Resolution |  |
| Offset |  |
| **Value**  (Discrete  Encoding) |  |  |
| **Voltage** | Logical signal for RHIB horizontal percentage change |
| **Unit** | |  |

Table: Encoding Details of ActuatorCommand\_RHIB\_Horizontal

Actuator to Sensor HRD

**Note:** An encoding is either discrete or continuous. Delete those fields, which are not needed.

|  |  |  |
| --- | --- | --- |
| **Value**  (Continuous Encoding) | Min Value |  |
| Max Value |  |
| Resolution |  |
| Offset |  |
| **Value**  (Discrete  Encoding) |  |  |
| **SensedRegPosLHOBHoriz** |  |
| **SensedRegPosLHOBVert** |  |
| **SensedRegPosLHIBHorz** |  |
| **SensedRegPosLHIBVert** |  |
| **SensedRegPosRHOBHoriz** |  |
| **SensedRegPosRHOBVert** |  |
| **SensedRegPosRHIBHoriz** |  |
| **SensedRegPosRHIBVert** |  |
| **Unit** | |  |

Table: Encoding Details of Actuator to Sensor HRD

SensFdbkLHOBHoriz

Sense left hand horizontal outboard register vane positions and forward to RCCM

**Note:** An encoding is either discrete or continuous. Delete those fields, which are not needed.

|  |  |  |
| --- | --- | --- |
| **Value**  (Continuous Encoding) | Min Value |  |
| Max Value |  |
| Resolution |  |
| Offset |  |
| **Value**  (Discrete  Encoding) |  |  |
| **voltage** | voltage property |
| **Unit** | |  |

Table: Encoding Details of SensFdbkLHOBHoriz

ActCmdLHOBVert

Signal from RCCM to left hand outboard vertical vane register actuator

**Note:** An encoding is either discrete or continuous. Delete those fields, which are not needed.

|  |  |  |
| --- | --- | --- |
| **Value**  (Continuous Encoding) | Min Value |  |
| Max Value |  |
| Resolution |  |
| Offset |  |
| **Value**  (Discrete  Encoding) |  |  |
| **voltage** | voltage property |
| **Unit** | |  |

Table: Encoding Details of ActCmdLHOBVert

SensedRegPosRHOBhoriz

**Note:** An encoding is either discrete or continuous. Delete those fields, which are not needed.

|  |  |  |
| --- | --- | --- |
| **Value**  (Continuous Encoding) | Min Value |  |
| Max Value |  |
| Resolution |  |
| Offset |  |
| **Value**  (Discrete  Encoding) |  |  |
| **Unit** | |  |

Table: Encoding Details of SensedRegPosRHOBhoriz

HMIFeedback\_LHOB\_Horizontal

Register position feedback to HMI

**Note:** An encoding is either discrete or continuous. Delete those fields, which are not needed.

|  |  |  |
| --- | --- | --- |
| **Value**  (Continuous Encoding) | Min Value |  |
| Max Value |  |
| Resolution |  |
| Offset |  |
| **Value**  (Discrete  Encoding) |  |  |
| **Unit** | |  |

Table: Encoding Details of HMIFeedback\_LHOB\_Horizontal

ActCmdRHOBHoriz

Signal from RCCM to right hand outboard horizontal vane register actuator

**Note:** An encoding is either discrete or continuous. Delete those fields, which are not needed.

|  |  |  |
| --- | --- | --- |
| **Value**  (Continuous Encoding) | Min Value |  |
| Max Value |  |
| Resolution |  |
| Offset |  |
| **Value**  (Discrete  Encoding) |  |  |
| **voltage** | voltage property |
| **Unit** | |  |

Table: Encoding Details of ActCmdRHOBHoriz

RgstrHzntlObr\_An\_Actl

Aim Status of Righthand Outboard Register Horizontal Vanes

**Note:** An encoding is either discrete or continuous. Delete those fields, which are not needed.

|  |  |  |
| --- | --- | --- |
| **Value**  (Continuous Encoding) | Min Value |  |
| Max Value |  |
| Resolution |  |
| Offset |  |
| **Value**  (Discrete  Encoding) |  |  |
| **Unit** | |  |

Table: Encoding Details of RgstrHzntlObr\_An\_Actl

SensedRegPosLHOBhoriz

**Note:** An encoding is either discrete or continuous. Delete those fields, which are not needed.

|  |  |  |
| --- | --- | --- |
| **Value**  (Continuous Encoding) | Min Value |  |
| Max Value |  |
| Resolution |  |
| Offset |  |
| **Value**  (Discrete  Encoding) |  |  |
| **Unit** | |  |

Table: Encoding Details of SensedRegPosLHOBhoriz

RgstrVertObr\_An\_Actl

Aim Status of Righthand Outboard Register Vertical Vanes

**Note:** An encoding is either discrete or continuous. Delete those fields, which are not needed.

|  |  |  |
| --- | --- | --- |
| **Value**  (Continuous Encoding) | Min Value |  |
| Max Value |  |
| Resolution |  |
| Offset |  |
| **Value**  (Discrete  Encoding) |  |  |
| **unnamed1** |  |
| **Unit** | |  |

Table: Encoding Details of RgstrVertObr\_An\_Actl

HMIFeedback\_RHOB\_Vertical

Register position feedback to HMI

**Note:** An encoding is either discrete or continuous. Delete those fields, which are not needed.

|  |  |  |
| --- | --- | --- |
| **Value**  (Continuous Encoding) | Min Value |  |
| Max Value |  |
| Resolution |  |
| Offset |  |
| **Value**  (Discrete  Encoding) |  |  |
| **Unit** | |  |

Table: Encoding Details of HMIFeedback\_RHOB\_Vertical

ActuatorCommand\_RHIB\_Vertical

Right hand inboard register voltage for left right movement

**Note:** An encoding is either discrete or continuous. Delete those fields, which are not needed.

|  |  |  |
| --- | --- | --- |
| **Value**  (Continuous Encoding) | Min Value |  |
| Max Value |  |
| Resolution |  |
| Offset |  |
| **Value**  (Discrete  Encoding) |  |  |
| **Voltage** | Logical signal for RHIB vertical percentage change |
| **Unit** | |  |

Table: Encoding Details of ActuatorCommand\_RHIB\_Vertical

Requester Type

Signal to indicate if register move request is from another feature or user

**Note:** An encoding is either discrete or continuous. Delete those fields, which are not needed.

|  |  |  |
| --- | --- | --- |
| **Value**  (Continuous Encoding) | Min Value |  |
| Max Value |  |
| Resolution |  |
| Offset |  |
| **Value**  (Discrete  Encoding) |  |  |
| **Unsupervised** |  |
| **Supervised** |  |
| **Unit** | |  |

Table: Encoding Details of Requester Type

Actuator to RCCM LIN

Signals sent from register actuator to RCCM

**Note:** An encoding is either discrete or continuous. Delete those fields, which are not needed.

|  |  |  |
| --- | --- | --- |
| **Value**  (Continuous Encoding) | Min Value |  |
| Max Value |  |
| Resolution |  |
| Offset |  |
| **Value**  (Discrete  Encoding) |  |  |
| **SensFdbkLHOBHoriz** | Implementation signal for SensFdbkLHOBHoriz |
| **SensFdbkLHOBVert** | Implementation signal for SensFdbkLHOBVert |
| **SensFdbkLHIBHoriz** | Implementation signal for SensFdbkLHIBHoriz |
| **SensFdbkLHIBVert** | Implementation signal for SensFdbkLHIBVert |
| **SensFdbkRHIBHoriz** | Implementation signal for SensFdbkRHIBHoriz |
| **SensFdbkRHIBVert** | Implementation signal for SensFdbkRHIBVert |
| **SensFdbkRHOBHoriz** | Implementation signal for SensFdbkRHOBHoriz |
| **SensFdbkRHOBVert** | Implementation signal for SensFdbkRHOBVert |
| **Unit** | |  |

Table: Encoding Details of Actuator to RCCM LIN

HMIFeedback\_RHIB\_Horizontal

Register position feedback to HMI

**Note:** An encoding is either discrete or continuous. Delete those fields, which are not needed.

|  |  |  |
| --- | --- | --- |
| **Value**  (Continuous Encoding) | Min Value |  |
| Max Value |  |
| Resolution |  |
| Offset |  |
| **Value**  (Discrete  Encoding) |  |  |
| **Unit** | |  |

Table: Encoding Details of HMIFeedback\_RHIB\_Horizontal

SensedRegPosLHIBhoriz

**Note:** An encoding is either discrete or continuous. Delete those fields, which are not needed.

|  |  |  |
| --- | --- | --- |
| **Value**  (Continuous Encoding) | Min Value |  |
| Max Value |  |
| Resolution |  |
| Offset |  |
| **Value**  (Discrete  Encoding) |  |  |
| **Unit** | |  |

Table: Encoding Details of SensedRegPosLHIBhoriz

ActCmdRHIBVert

Signal from RCCM to right hand inboard vertical vane register actuator

**Note:** An encoding is either discrete or continuous. Delete those fields, which are not needed.

|  |  |  |
| --- | --- | --- |
| **Value**  (Continuous Encoding) | Min Value |  |
| Max Value |  |
| Resolution |  |
| Offset |  |
| **Value**  (Discrete  Encoding) |  |  |
| **voltage** | voltage property |
| **Unit** | |  |

Table: Encoding Details of ActCmdRHIBVert

VehStat

Vehicle status signals used to store register positions when vehicle is on

**Note:** An encoding is either discrete or continuous. Delete those fields, which are not needed.

|  |  |  |
| --- | --- | --- |
| **Value**  (Continuous Encoding) | Min Value |  |
| Max Value |  |
| Resolution |  |
| Offset |  |
| **Value**  (Discrete  Encoding) |  |  |
| **unnamed1** |  |
| **ON** | Implementation signal for vehicle status=on |
| **OFF** | Implementation signal for vehicle status=OFF |
| **ACCESSORY** | Implementation signal for vehicle status=ACCESSORY |
| **Unit** | |  |

Table: Encoding Details of VehStat

Register\_RHOB\_Vertical

Register airflow direction left or right

**Note:** An encoding is either discrete or continuous. Delete those fields, which are not needed.

|  |  |  |
| --- | --- | --- |
| **Value**  (Continuous Encoding) | Min Value |  |
| Max Value |  |
| Resolution |  |
| Offset |  |
| **Value**  (Discrete  Encoding) |  |  |
| **Percentage** | Logical signal for register RHOB horizontal percentage change position vertical |
| **Unit** | |  |

Table: Encoding Details of Register\_RHOB\_Vertical

userFeedback

User feedback sent to the HMI visual output

**Note:** An encoding is either discrete or continuous. Delete those fields, which are not needed.

|  |  |  |
| --- | --- | --- |
| **Value**  (Continuous Encoding) | Min Value |  |
| Max Value |  |
| Resolution |  |
| Offset |  |
| **Value**  (Discrete  Encoding) |  |  |
| **HMI Visual Output** | logical signal for HMI visual output |
| **Unit** | |  |

Table: Encoding Details of userFeedback

RgstrVertObl\_An\_Rq

Command Position of Vertical Vanes for Lefthand Outboard Register

**Note:** An encoding is either discrete or continuous. Delete those fields, which are not needed.

|  |  |  |
| --- | --- | --- |
| **Value**  (Continuous Encoding) | Min Value |  |
| Max Value |  |
| Resolution |  |
| Offset |  |
| **Value**  (Discrete  Encoding) |  |  |
| **unnamed1** |  |
| **Unit** | |  |

Table: Encoding Details of RgstrVertObl\_An\_Rq

SensorFeedback\_RHOB\_Horizontal

Register sensor position feedback to HMI for right hand outboard register horizontal voltage

**Note:** An encoding is either discrete or continuous. Delete those fields, which are not needed.

|  |  |  |
| --- | --- | --- |
| **Value**  (Continuous Encoding) | Min Value |  |
| Max Value |  |
| Resolution |  |
| Offset |  |
| **Value**  (Discrete  Encoding) |  |  |
| **Unit** | |  |

Table: Encoding Details of SensorFeedback\_RHOB\_Horizontal

Register\_Location

Register location in IP

**Note:** An encoding is either discrete or continuous. Delete those fields, which are not needed.

|  |  |  |
| --- | --- | --- |
| **Value**  (Continuous Encoding) | Min Value |  |
| Max Value |  |
| Resolution |  |
| Offset |  |
| **Value**  (Discrete  Encoding) |  |  |
| **Register\_LHOB** |  |
| **Register\_LHIB** |  |
| **Register\_RHIB** |  |
| **Register\_RHOB** |  |
| **Unit** | |  |

Table: Encoding Details of Register\_Location

RCCM to APIM ActvButtn

Signals sent from RCCM to APIM

**Note:** An encoding is either discrete or continuous. Delete those fields, which are not needed.

|  |  |  |
| --- | --- | --- |
| **Value**  (Continuous Encoding) | Min Value |  |
| Max Value |  |
| Resolution |  |
| Offset |  |
| **Value**  (Discrete  Encoding) |  |  |
| **ActiveButtonLHS\_D\_Stat** | Implementation signal from RCCM to APIM for ActiveButtonsLHS\_D\_Stat |
| **ActiveButtonRHS\_D\_Stat** | Implementation signal from RCCM to APIM for ActiveButtonsRHS\_D\_Stat |
| **Unit** | |  |

Table: Encoding Details of RCCM to APIM ActvButtn

ActuatorCommand\_RHOB\_Horizontal

Right hand outboard register voltage for up down movement

**Note:** An encoding is either discrete or continuous. Delete those fields, which are not needed.

|  |  |  |
| --- | --- | --- |
| **Value**  (Continuous Encoding) | Min Value |  |
| Max Value |  |
| Resolution |  |
| Offset |  |
| **Value**  (Discrete  Encoding) |  |  |
| **Voltage** | Logical signal for RHOB horizontal percentage change |
| **Unit** | |  |

Table: Encoding Details of ActuatorCommand\_RHOB\_Horizontal

SensedRegPosRHIBhoriz

**Note:** An encoding is either discrete or continuous. Delete those fields, which are not needed.

|  |  |  |
| --- | --- | --- |
| **Value**  (Continuous Encoding) | Min Value |  |
| Max Value |  |
| Resolution |  |
| Offset |  |
| **Value**  (Discrete  Encoding) |  |  |
| **Unit** | |  |

Table: Encoding Details of SensedRegPosRHIBhoriz

SensedRegPosLHOBvert

**Note:** An encoding is either discrete or continuous. Delete those fields, which are not needed.

|  |  |  |
| --- | --- | --- |
| **Value**  (Continuous Encoding) | Min Value |  |
| Max Value |  |
| Resolution |  |
| Offset |  |
| **Value**  (Discrete  Encoding) |  |  |
| **Unit** | |  |

Table: Encoding Details of SensedRegPosLHOBvert

HMIFeedback\_RHOB\_Horizontal

Register position feedback to HMI

**Note:** An encoding is either discrete or continuous. Delete those fields, which are not needed.

|  |  |  |
| --- | --- | --- |
| **Value**  (Continuous Encoding) | Min Value |  |
| Max Value |  |
| Resolution |  |
| Offset |  |
| **Value**  (Discrete  Encoding) |  |  |
| **Unit** | |  |

Table: Encoding Details of HMIFeedback\_RHOB\_Horizontal

HMIFeedback\_LHIB\_Horizontal

Register position feedback to HMI

**Note:** An encoding is either discrete or continuous. Delete those fields, which are not needed.

|  |  |  |
| --- | --- | --- |
| **Value**  (Continuous Encoding) | Min Value |  |
| Max Value |  |
| Resolution |  |
| Offset |  |
| **Value**  (Discrete  Encoding) |  |  |
| **Unit** | |  |

Table: Encoding Details of HMIFeedback\_LHIB\_Horizontal

HMIFeedback\_RHIB

Register position feedback to HMI

**Note:** An encoding is either discrete or continuous. Delete those fields, which are not needed.

|  |  |  |
| --- | --- | --- |
| **Value**  (Continuous Encoding) | Min Value |  |
| Max Value |  |
| Resolution |  |
| Offset |  |
| **Value**  (Discrete  Encoding) |  |  |
| **Position** | Logical signal HMI feedback RHIB position |
| **Close** | Logical signal HMI feedback RHIB close |
| **Unit** | |  |

Table: Encoding Details of HMIFeedback\_RHIB

ActvButtnLeft\_D\_Rq

Subfeature Selection Commands for Left Side Registers

**Note:** An encoding is either discrete or continuous. Delete those fields, which are not needed.

|  |  |  |
| --- | --- | --- |
| **Value**  (Continuous Encoding) | Min Value |  |
| Max Value |  |
| Resolution |  |
| Offset |  |
| **Value**  (Discrete  Encoding) |  |  |
| **0x0 = nonePressed (Default)** |  |
| **0x1 = onBodyRHS** |  |
| **0x2 = offBodyRHS** |  |
| **0x3 = presetOneLHS** |  |
| **0x4 = presetTwoRHS** |  |
| **0x5** |  |
| **0x6** |  |
| **0x7** |  |
| **0x8** |  |
| **0x9** |  |
| **0x10** |  |
| **0x11 = cycle\_|\_RHS** |  |
| **0xC - 0xF** |  |
| **Unit** | |  |

Table: Encoding Details of ActvButtnLeft\_D\_Rq

ActuatorCommand\_LHOB\_Vertical

Left hand outboard register voltage for up down movement

**Note:** An encoding is either discrete or continuous. Delete those fields, which are not needed.

|  |  |  |
| --- | --- | --- |
| **Value**  (Continuous Encoding) | Min Value |  |
| Max Value |  |
| Resolution |  |
| Offset |  |
| **Value**  (Discrete  Encoding) |  |  |
| **Voltage** | Logical signal for LHOB vertical percentage change |
| **Unit** | |  |

Table: Encoding Details of ActuatorCommand\_LHOB\_Vertical

Register\_LHIB\_Vertical

Register vane left or right airflow direction

**Note:** An encoding is either discrete or continuous. Delete those fields, which are not needed.

|  |  |  |
| --- | --- | --- |
| **Value**  (Continuous Encoding) | Min Value |  |
| Max Value |  |
| Resolution |  |
| Offset |  |
| **Value**  (Discrete  Encoding) |  |  |
| **Percentage** | Logical signal for register LHIB horizontal percentage change position vertical |
| **Unit** | |  |

Table: Encoding Details of Register\_LHIB\_Vertical

SensorFeedback\_LHOB\_Horizontal

Register sensor position feedback to HMI for Left hand outboard register horizontal voltage

**Note:** An encoding is either discrete or continuous. Delete those fields, which are not needed.

|  |  |  |
| --- | --- | --- |
| **Value**  (Continuous Encoding) | Min Value |  |
| Max Value |  |
| Resolution |  |
| Offset |  |
| **Value**  (Discrete  Encoding) |  |  |
| **Unit** | |  |

Table: Encoding Details of SensorFeedback\_LHOB\_Horizontal

RgstrSetIbl\_D\_Rq

Change Aim Command for Lefthand Inboard Register

**Note:** An encoding is either discrete or continuous. Delete those fields, which are not needed.

|  |  |  |
| --- | --- | --- |
| **Value**  (Continuous Encoding) | Min Value |  |
| Max Value |  |
| Resolution |  |
| Offset |  |
| **Value**  (Discrete  Encoding) |  |  |
| **0x0** |  |
| **0x1** |  |
| **0x2** |  |
| **0x3** |  |
| **0x4** |  |
| **0x5** |  |
| **0x6** |  |
| **0x7** |  |
| **Unit** | |  |

Table: Encoding Details of RgstrSetIbl\_D\_Rq

Request HRD

Place holder for hardwire signals

**Note:** An encoding is either discrete or continuous. Delete those fields, which are not needed.

|  |  |  |
| --- | --- | --- |
| **Value**  (Continuous Encoding) | Min Value |  |
| Max Value |  |
| Resolution |  |
| Offset |  |
| **Value**  (Discrete  Encoding) |  |  |
| **Unit** | |  |

Table: Encoding Details of Request HRD

RgstrHzntlObl\_An\_Actl

Aim Status of Lefthand Outboard Register Horizontal Vanes

**Note:** An encoding is either discrete or continuous. Delete those fields, which are not needed.

|  |  |  |
| --- | --- | --- |
| **Value**  (Continuous Encoding) | Min Value |  |
| Max Value |  |
| Resolution |  |
| Offset |  |
| **Value**  (Discrete  Encoding) |  |  |
| **unnamed1** |  |
| **Unit** | |  |

Table: Encoding Details of RgstrHzntlObl\_An\_Actl

SensorFeedBack\_RHIB\_Vertical

Register sensor position feedback to HMI for right hand inboard register vertical voltage

**Note:** An encoding is either discrete or continuous. Delete those fields, which are not needed.

|  |  |  |
| --- | --- | --- |
| **Value**  (Continuous Encoding) | Min Value |  |
| Max Value |  |
| Resolution |  |
| Offset |  |
| **Value**  (Discrete  Encoding) |  |  |
| **Unit** | |  |

Table: Encoding Details of SensorFeedBack\_RHIB\_Vertical

Register\_RHIB\_Vertical

Register airflow direction left or right

**Note:** An encoding is either discrete or continuous. Delete those fields, which are not needed.

|  |  |  |
| --- | --- | --- |
| **Value**  (Continuous Encoding) | Min Value |  |
| Max Value |  |
| Resolution |  |
| Offset |  |
| **Value**  (Discrete  Encoding) |  |  |
| **Percentage** | Logical signal for register RHIB horizontal percentage change position vertical |
| **Unit** | |  |

Table: Encoding Details of Register\_RHIB\_Vertical

Register\_LHIB

Register location driver inboard

**Note:** An encoding is either discrete or continuous. Delete those fields, which are not needed.

|  |  |  |
| --- | --- | --- |
| **Value**  (Continuous Encoding) | Min Value |  |
| Max Value |  |
| Resolution |  |
| Offset |  |
| **Value**  (Discrete  Encoding) |  |  |
| **NoChange** | Logical signal for register LHIB no changel position |
| **ManualChange** | Logical signal for register LHIB no change position |
| **FeatureChange** | Logical signal for register LHIB Feature change position |
| **Open** | Logical signal for register LHIB open position |
| **Close** | Logical signal for register LHIB close position |
| **Unit** | |  |

Table: Encoding Details of Register\_LHIB

ActuatorCommand\_LHIB\_Vertical

Register left hand inboard voltage for vertical movement

**Note:** An encoding is either discrete or continuous. Delete those fields, which are not needed.

|  |  |  |
| --- | --- | --- |
| **Value**  (Continuous Encoding) | Min Value |  |
| Max Value |  |
| Resolution |  |
| Offset |  |
| **Value**  (Discrete  Encoding) |  |  |
| **Voltage** | Logical signal for LHIB vertical percentage change |
| **Unit** | |  |

Table: Encoding Details of ActuatorCommand\_LHIB\_Vertical

EM\_PredefinedPositionsRequest

Move register commands based on register postion settings or button options

**Note:** An encoding is either discrete or continuous. Delete those fields, which are not needed.

|  |  |  |
| --- | --- | --- |
| **Value**  (Continuous Encoding) | Min Value |  |
| Max Value |  |
| Resolution |  |
| Offset |  |
| **Value**  (Discrete  Encoding) |  |  |
| **ButtonPressedRHS** |  |
| **ButtonPressedLHS** |  |
| **requesterType** |  |
| **Unit** | |  |

Table: Encoding Details of EM\_PredefinedPositionsRequest

APIM to RCCM

Signals sent from APIM to RCCM

**Note:** An encoding is either discrete or continuous. Delete those fields, which are not needed.

|  |  |  |
| --- | --- | --- |
| **Value**  (Continuous Encoding) | Min Value |  |
| Max Value |  |
| Resolution |  |
| Offset |  |
| **Value**  (Discrete  Encoding) |  |  |
| **RgstrHzntlIbl\_An\_Rq** |  |
| **RgstrHzntIbr\_An\_Rq** |  |
| **RgstrHzntlObl\_An\_Rq** |  |
| **RgstrHzntlObr\_An\_Rq** |  |
| **RgstrVertIbl\_An\_Rq** |  |
| **RgstrVertbr\_An\_Rq** |  |
| **RgstrVertObl\_An\_Rq** |  |
| **RgstrVertObr\_An\_Rq** |  |
| **Apim to Rccm Buttons** |  |
| **APIM to RCCM RGSTR SET** |  |
| **Unit** | |  |

Table: Encoding Details of APIM to RCCM

Clmt\_Button\_Stat4

Climate button signal

**Note:** An encoding is either discrete or continuous. Delete those fields, which are not needed.

|  |  |  |
| --- | --- | --- |
| **Value**  (Continuous Encoding) | Min Value |  |
| Max Value |  |
| Resolution |  |
| Offset |  |
| **Value**  (Discrete  Encoding) |  |  |
| **None\_Pressed** | Implementation signal for climate button none pressed |
| **Front\_Power\_Pressed** | Implementation signal for climate button front power pressed |
| **Unit** | |  |

Table: Encoding Details of Clmt\_Button\_Stat4

ActvButtnRight\_D\_Rq

Subfeature Selection Commands for Right Side Registers

**Note:** An encoding is either discrete or continuous. Delete those fields, which are not needed.

|  |  |  |
| --- | --- | --- |
| **Value**  (Continuous Encoding) | Min Value |  |
| Max Value |  |
| Resolution |  |
| Offset |  |
| **Value**  (Discrete  Encoding) |  |  |
| **0x0 = nonePressed (Default)** |  |
| **0x1 = onBodyRHS** |  |
| **0x2 = offBodyRHS** |  |
| **0x3 = presetOneLHS** |  |
| **0x4 = presetTwoRHS** |  |
| **0x5** |  |
| **0x6** |  |
| **0x7** |  |
| **0x8** |  |
| **0x9** |  |
| **0x10** |  |
| **0x11 = cycle\_|\_RHS** |  |
| **0xC - 0xF** |  |
| **Unit** | |  |

Table: Encoding Details of ActvButtnRight\_D\_Rq

RegHomePositions

Signal for register home positions

**Note:** An encoding is either discrete or continuous. Delete those fields, which are not needed.

|  |  |  |
| --- | --- | --- |
| **Value**  (Continuous Encoding) | Min Value |  |
| Max Value |  |
| Resolution |  |
| Offset |  |
| **Value**  (Discrete  Encoding) |  |  |
| **Reg\_LHIB\_Horz\_Home** |  |
| **Reg\_LHIB\_Vert\_Home** |  |
| **Reg\_LHOB\_Horz\_Home** |  |
| **Reg\_LHOB\_Vert\_Home** |  |
| **Reg\_RHIB\_Horz\_Home** |  |
| **Reg\_RHIB\_Vert\_Home** |  |
| **Reg\_RHOB\_Horz\_Home** |  |
| **Reg\_RHOB\_Vert\_Home** |  |
| **Reg\_LHIB\_Horz\_Angle\_Percentage** |  |
| **Reg\_LHIB\_Vert\_Angle\_Percentage** |  |
| **Reg\_LHOB\_Horz\_Angle\_Percentage** |  |
| **Reg\_LHOB\_Vert\_Angle\_Percentage** |  |
| **Reg\_RHIB\_Horz\_Angle\_Percentage** |  |
| **Reg\_RHOB\_Vert\_Angle\_Percentage** |  |
| **Reg\_RHIB\_Vert\_Angle\_Percentage** |  |
| **Reg\_RHOB\_Horz\_Angle\_Percentage** |  |
| **Unit** | |  |

Table: Encoding Details of RegHomePositions

Register\_LHOB\_Vertical

Register airflow left or right airflow direction

**Note:** An encoding is either discrete or continuous. Delete those fields, which are not needed.

|  |  |  |
| --- | --- | --- |
| **Value**  (Continuous Encoding) | Min Value |  |
| Max Value |  |
| Resolution |  |
| Offset |  |
| **Value**  (Discrete  Encoding) |  |  |
| **Percentage** | Logical signal for register LHOB horizontal percentage change position vertical |
| **Unit** | |  |

Table: Encoding Details of Register\_LHOB\_Vertical

RgstrHzntlObr\_An\_Rq

Command Position of Horizontal Vanes for Righthand Outboard Register

**Note:** An encoding is either discrete or continuous. Delete those fields, which are not needed.

|  |  |  |
| --- | --- | --- |
| **Value**  (Continuous Encoding) | Min Value |  |
| Max Value |  |
| Resolution |  |
| Offset |  |
| **Value**  (Discrete  Encoding) |  |  |
| **Unit** | |  |

Table: Encoding Details of RgstrHzntlObr\_An\_Rq

RCCM to APIM

**Note:** An encoding is either discrete or continuous. Delete those fields, which are not needed.

|  |  |  |
| --- | --- | --- |
| **Value**  (Continuous Encoding) | Min Value |  |
| Max Value |  |
| Resolution |  |
| Offset |  |
| **Value**  (Discrete  Encoding) |  |  |
| **Register Horz IBL** |  |
| **Register Horz IBR** |  |
| **Register Horz OBL** |  |
| **Register Horz OBR** |  |
| **Register Vert IBL** |  |
| **Register Vert IBR** |  |
| **Register Vert OBL** |  |
| **Register Vert OBR** |  |
| **Active Buttons** |  |
| **Unit** | |  |

Table: Encoding Details of RCCM to APIM

vehicleStatus

Vehicle on off status used by register move commands

**Note:** An encoding is either discrete or continuous. Delete those fields, which are not needed.

|  |  |  |
| --- | --- | --- |
| **Value**  (Continuous Encoding) | Min Value |  |
| Max Value |  |
| Resolution |  |
| Offset |  |
| **Value**  (Discrete  Encoding) |  |  |
| **ON\_TO\_OFF** | vehicle status on |
| **OFF\_TO\_ON** | vehicle status off |
| **Accessory** | vehicle status accessory |
| **Unit** | |  |

Table: Encoding Details of vehicleStatus

ActvButtnRight\_D\_Stat

Subfeature Selection Status for Right Side Registers:

\*NonePressed

\*OnBodyLHS

\*OffBodyLHS

\*PresetOneLHS

\*PresetTwoLHS

\*Cycle\_8\_LHS

\*Cycle\_C\_LHS

\*Cycle\_O\_LHS

\*Cycle\_--\_LHS

\*Cycle\_I\_LHS

**Note:** An encoding is either discrete or continuous. Delete those fields, which are not needed.

|  |  |  |
| --- | --- | --- |
| **Value**  (Continuous Encoding) | Min Value |  |
| Max Value |  |
| Resolution |  |
| Offset |  |
| **Value**  (Discrete  Encoding) |  |  |
| **0x0 nonePressed** | Button pressed discrete signal for none pressed RHS selection |
| **0x1 onBodyRHS** | Button pressed discrete signal for on body RHS selection |
| **0x2 offBodyRHS** | Button pressed discrete signal for off body RHS selection |
| **0x3 presetOneRHS** | Button pressed discrete signal for preset one RHS selection |
| **0x4 presetTwoRHS** | Button pressed discrete signal for preset two RHS selection |
| **0x6 presetTwoSaveRHS** | Button pressed discrete signal for preset one saved RHS selection |
| **0x5 cycle\_8\_RHS** | Button pressed discrete signal cycling in figure 8 RHS selection |
| **0x5 presetOneSaveRHS** | Button pressed discrete signal for preset one saved RHS selection |
| **0x6 cycle\_C\_RHS** | Button pressed discrete signal for closed vanes RHS selection |
| **0x7 cycle\_O\_RHS** | Button pressed discrete signal open vanes RHS selection |
| **0x8 cycle\_--\_RHS** | Button pressed discrete signal for cycling vanes in horizontal direction RHS selection |
| **0x9 cycle\_|\_RHS** | Button pressed discrete signal for cycling vanes in vertical direction RHS selection |
| **Unit** | |  |

Table: Encoding Details of ActvButtnRight\_D\_Stat

RegisterPositionHMIFeedback

Register sensor position feedback to HMI

**Note:** An encoding is either discrete or continuous. Delete those fields, which are not needed.

|  |  |  |
| --- | --- | --- |
| **Value**  (Continuous Encoding) | Min Value |  |
| Max Value |  |
| Resolution |  |
| Offset |  |
| **Value**  (Discrete  Encoding) |  |  |
| **unnamed1** |  |
| **HMIFeedbck\_LHOB\_Horz** |  |
| **HMIFeedbck\_LHOB\_Vert** |  |
| **HMIFeedbck\_LHIB\_Horz** |  |
| **HMIFeedbck\_LHIB\_Vert** |  |
| **HMIFeedbck\_RHIB\_Horz** |  |
| **HMIFeedbck\_RHIB\_Vert** |  |
| **HMIFeedbck\_RHOB\_Horz** |  |
| **HMIFeedbck\_RHOB\_Vert** |  |
| **Unit** | |  |

Table: Encoding Details of RegisterPositionHMIFeedback

RgstrHzntlIbl\_An\_Rq

Command Position of Horizontal Vanes for Lefthand Inboard Register

**Note:** An encoding is either discrete or continuous. Delete those fields, which are not needed.

|  |  |  |
| --- | --- | --- |
| **Value**  (Continuous Encoding) | Min Value |  |
| Max Value |  |
| Resolution |  |
| Offset |  |
| **Value**  (Discrete  Encoding) |  |  |
| **Unit** | |  |

Table: Encoding Details of RgstrHzntlIbl\_An\_Rq

MoveAirFlowCommand

Move register commands from user input or another feature

**Note:** An encoding is either discrete or continuous. Delete those fields, which are not needed.

|  |  |  |
| --- | --- | --- |
| **Value**  (Continuous Encoding) | Min Value |  |
| Max Value |  |
| Resolution |  |
| Offset |  |
| **Value**  (Discrete  Encoding) |  |  |
| **userInput** | Logical signal for move airflow command for user input to move registers |
| **Unit** | |  |

Table: Encoding Details of MoveAirFlowCommand

HMIFeedback\_LHIB

Register position feedback to HMI

**Note:** An encoding is either discrete or continuous. Delete those fields, which are not needed.

|  |  |  |
| --- | --- | --- |
| **Value**  (Continuous Encoding) | Min Value |  |
| Max Value |  |
| Resolution |  |
| Offset |  |
| **Value**  (Discrete  Encoding) |  |  |
| **Position** | Logical signal HMI feedback LHIB position |
| **Close** | Logical signal HMI feedback LHIB close |
| **Unit** | |  |

Table: Encoding Details of HMIFeedback\_LHIB

HMIFeedback\_LHOB\_Vertical

Register position feedback to HMI

**Note:** An encoding is either discrete or continuous. Delete those fields, which are not needed.

|  |  |  |
| --- | --- | --- |
| **Value**  (Continuous Encoding) | Min Value |  |
| Max Value |  |
| Resolution |  |
| Offset |  |
| **Value**  (Discrete  Encoding) |  |  |
| **Unit** | |  |

Table: Encoding Details of HMIFeedback\_LHOB\_Vertical

### Technology State Machines

Document ends here.