**Module Design Document**

**For**

**CF14 PSA Torque Arbitrator**

**VERSION: 1.0**

**DATE: 10-Mar-2014**

**Prepared By:**

**Sankardu Varadapureddi,**

**Nexteer Automotive,**

**Saginaw, MI, USA**

**Location:** The official version of this document is stored in the Nexteer Configuration Management System.

**Revision History**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Sl. No.** | **Description** | **Author** | **Version** | **Date** |
| 1 | Initial Version | Sankardu Varadapureddi | 1.0 | 10-Mar-2015 |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

**Table of Contents**

[1 Abbrevations And Acronyms 5](#_Toc414443241)

[2 References 6](#_Toc414443242)

[3 PSA State Handler High-Level Description 7](#_Toc414443243)

[4 Design details of software module 8](#_Toc414443244)

[4.1 Graphical representation of PSA Torque Arbitrator 8](#_Toc414443245)

[4.2 Data Flow Diagram 8](#_Toc414443246)

[4.2.1 Module level DFD 8](#_Toc414443247)

[4.2.2 Sub-Module level DFD 8](#_Toc414443248)

[4.3 COMPONENT FLOW DIAGRAM 8](#_Toc414443249)

[5 Variable Data Dictionary 9](#_Toc414443250)

[5.1 User defined typedef definition/declaration 9](#_Toc414443251)

[5.2 Variable definition for enumerated types 9](#_Toc414443252)

[6 Constant Data Dictionary 10](#_Toc414443253)

[6.1 Program(fixed) Constants 10](#_Toc414443254)

[6.1.1 Embedded Constants 10](#_Toc414443255)

[6.1.1.1 Local 10](#_Toc414443256)

[6.1.1.2 Global 10](#_Toc414443257)

[6.1.2 Module specific Lookup Tables Constants 10](#_Toc414443258)

[7 Software Module Implementation 11](#_Toc414443259)

[7.1 Sub-Module Functions 11](#_Toc414443260)

[7.1.1 Initialization Functions 11](#_Toc414443261)

[7.1.1.1 Init: PSATA\_Init1 11](#_Toc414443262)

[7.1.1.1.1 Design Rationale 11](#_Toc414443263)

[7.1.1.1.2 Module Outputs 11](#_Toc414443264)

[7.1.1.1.3 Module Internal 11](#_Toc414443265)

[7.1.2 PERIODIC FUNCTIONS 11](#_Toc414443266)

[7.1.2.1 Per: PSATA\_per1 11](#_Toc414443267)

[7.1.2.1.1 Design Rationale 11](#_Toc414443268)

[7.1.2.1.2 Store Module Inputs to Local copies 11](#_Toc414443269)

[7.1.2.1.3 (Processing of function)……… 11](#_Toc414443270)

[7.1.2.1.4 Store Local copy of outputs into Module Outputs 11](#_Toc414443271)

[7.1.3 Interrupt Functions 12](#_Toc414443272)

[7.1.4 Serial Communication Functions 13](#_Toc414443273)

[7.1.5 Local Function/Macro Definitions 13](#_Toc414443274)

[7.1.5.1 Local Function #1 13](#_Toc414443275)

[7.1.5.1.1 Description 13](#_Toc414443276)

[7.1.5.2 Local Function #2 13](#_Toc414443277)

[7.1.5.2.1 Description 13](#_Toc414443278)

[7.1.6 GLObAL Function/Macro Definitions 13](#_Toc414443279)

[7.1.7 Tranisition FUNCTIONS 13](#_Toc414443280)

[8 Known Limitations With Design 14](#_Toc414443281)

[9 UNIT TEST CONSIDERATION 15](#_Toc414443282)

[10 Appendix 16](#_Toc414443283)

# Abbrevations And Acronyms

|  |  |
| --- | --- |
| Abbreviation | Description |
| DFD | Design functional diagram |
| MDD | Module design Document |
| FDD | Functional Design Document |
| CF | Customer Function |

# References

This section lists the title & version of all the documents that are referred for development of this document

|  |  |  |
| --- | --- | --- |
| Sr. No. | Title | Version |
| 1 | MDD Guidelines | 1.4 |
| 2 | Software Naming Conventions | 1.2 |
| 3 | Software Design and Coding standards | 2.1 |
| 4 | CF14 PSA State handler FDD | 1.1.0 |
| 5 | Data Dictionary.xlsm | 1.0 |

# PSA State Handler High-Level Description

*The PSA Torque Arbitrator will be equipped for EPS Systems with functions including Ramping and smoothing of PosServo command and Safety function. The safety function will monitor the PSA State Handler and PSA Torque Arbitrator.*

# Design details of software module

## Graphical representation of PSA Torque Arbitrator

*Refer FDD*

## Data Flow Diagram

*Refer FDD*

## Module level DFD

*Refer FDD*

## Sub-Module level DFD

*Refer FDD*

## COMPONENT FLOW DIAGRAM

*Refer FDD*

# Variable Data Dictionary

## User defined typedef definition/declaration

*<This section documents any user types uniquely used for the module.>*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Typedef Name | Element Name | User Defined Type | Legal Range  (min) | Legal Range  (max) |
| None |  |  |  |  |
|  |  |  |  |  |

## Variable definition for enumerated types

|  |  |  |
| --- | --- | --- |
| Enum Name | Element Name | Value |
| None |  |  |

# Constant Data Dictionary

## Program(fixed) Constants

## Embedded Constants

## Local

|  |  |  |  |
| --- | --- | --- | --- |
| Constant Name | Resolution | Units | Value |
| D\_NTCHIGH\_CNT\_LGC | 1 | CNT | 1 |
| D\_NTCLOW\_CNT\_LGC | 1 | CNT | 0 |
| D\_POSSRVONTCENABLE\_MTRNM\_F32 | single preicision float | MTRNM | 0.0F |
| D\_POSSRVOTRANSITION\_ULS\_F32 | single preicision float | ULS | 0.01999999955F |
| D\_SMOOTHINGHIGH\_CNT\_F32 | 1 | CNT | 1 |
| D\_SMOOTHINGLOW\_CNT\_F32 | 1 | CNT | 0 |

## Global

|  |
| --- |
| Constant Name |
| D\_ZERO\_CNT\_U8 |
| D\_ZERO\_ULS\_F32 |
| D\_TRUE\_CNT\_LGC |
| D\_2MS\_SEC\_F32 |
| D\_MTRTRQCMDHILMT\_MTRNM\_F32 |
| D\_MTRTRQCMDLOLMT\_MTRNM\_F32 |

## Module specific Lookup Tables Constants

|  |  |  |  |
| --- | --- | --- | --- |
| Constant Name | Resolution | Value | Software Segment |
| None |  |  |  |

# Software Module Implementation

## Sub-Module Functions

## Initialization Functions

## Init: PSATA\_Init1

## Design Rationale

*Low pass filter to calucalte the init co-efficient ‘PSATA\_FilterdTrqSV\_HwNm\_M\_Str’.*

*Also trigger DIAG manager to make ‘NTC\_Num\_SigPath5CrossChk’ ‘test not ready status’ to cleared. This is done by setting NTC status to ‘Passed’.*

## Module Outputs

None

## Module Internal

PSATA\_FilterdTrqSV\_HwNm\_M\_Str

## PERIODIC FUNCTIONS

## Per: PSATA\_per1

## Design Rationale

*Design follows implemenetation in FDD.*

*As per FDD owner,*

1. *‘NTC\_Num\_PosServFltMode’ and ‘NTC\_Num\_SigPath5CrossChk’ are different names used in different documents for the same NTC ‘196u’. Note that ‘NTC\_Num\_PosServFltMode’ is ignitial latched.*
2. *In block ‘PSATA\_Per1’, there is a switch block based on 'PosSrvoNTC\_Cnt\_lgc >=1 'condition. ‘>’ condition will never be TRUE since 'PosSrvoNTC\_Cnt\_lgc' gets either ‘1’ or ‘0’ only. [Here option of ''PosSrvoNTC\_Cnt\_lgc ~=0' had been considered in the FDD design phase but if there is toggling in the value then in design it is not considered that option due to previous experiences.]*

## Store Module Inputs to Local copies

HwTorque\_HwNm\_T\_f32 = Rte\_IRead\_PSATA\_Per1\_HwTorque\_HwNm\_f32();

PosSrvoCmd\_MtrNm\_T\_f32 = Rte\_IRead\_PSATA\_Per1\_PosSrvoCmd\_MtrNm\_f32();

PosSrvoEnable\_Cnt\_T\_lgc = Rte\_IRead\_PSATA\_Per1\_PosSrvoEnable\_Cnt\_lgc();

VehicleSpeed\_Kph\_T\_f32 = Rte\_IRead\_PSATA\_Per1\_VehicleSpeed\_Kph\_f32();

## (Processing of function)………

*Refer to FDD (Block ‘PSATA\_Per1’)*

## Store Local copy of outputs into Module Outputs

Rte\_IWrite\_PSATA\_Per1\_OpTrqOv\_MtrNm\_f32(OpTrqOv\_MtrNm\_T\_f32);

Rte\_IWrite\_PSATA\_Per1\_PosSrvoNTC\_Cnt\_lgc(PSATA\_PosSrvoNTC\_Cnt\_M\_lgc);

## Interrupt Functions

*None*

## Serial Communication Functions

None

## Local Function/Macro Definitions

## Local Function #1

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Function Name** | Cal\_PosSrvoNTC | Type | Min | Max |
| **Arguments Passed** | PosSrvoCmd\_MtrNm\_T\_f32 | Float32 | -8.800000191 | 8.800000191 |
|  | PosSrvoEnable\_Cnt\_T\_lgc | Boolean | FALSE | TRUE |
|  | VehicleSpeed\_Kph\_T\_f32 | Float32 | 0 | 511 |
| **Return Value** | N/A |  |  |  |

## Description

This function monitors 'State Handler' and 'PosServo' for errors. Sets 'PosSrvoNTC\_Cnt\_lgc' signal accordingly.

Note: This implementation corresponds to lower half of 'PSATA\_Per1' block.

## Local Function #2

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Function Name** | Cal\_PosServoSmoothingFactor | Type | Min | Max |
| **Arguments Passed** | HwTorque\_HwNm\_T\_f32 | Float32 | -10 | 10 |
|  | PosSrvoEnable\_Cnt\_T\_lgc | Boolean | FALSE | TRUE |
| **Return Value** | N/A |  |  |  |

## Description

Implementation of 'PosServoSmoothing' block. PosServoCmd changes instantly from zero when disabled to non-zero when enabled, and vice-versa. This routine calculates a scale factor for the PosServoCmd to smoothly ramp it in and out. First, it produces a linear scale factor, then feeds the linear factor into a lookup table to non-linearize it.This produces softer transitions when scale factor is near zero or near unity. The scale factor can decrease more rapidly when driver hand wheel torque is present.

## GLObAL Function/Macro Definitions

None

## Tranisition FUNCTIONS

None

# Known Limitations With Design

# UNIT TEST CONSIDERATION

FDD describes functionality and structural breakdown of this componenet. Data dictionary contains all attributes of varibales and calibrations used.

# Appendix