**Module Design Document**

**For**

**SysFricLrng**

**Oct 04, 2017**

**Prepared By:**

**Matthew Leser**

**Nexteer Automotive,**

**Saginaw, MI, USA**

**Change History**

|  |  |  |  |
| --- | --- | --- | --- |
| **Description** | **Author** | **Version** | **Date** |
| Initial Version | Basavaraja Ganeshappa | 1.0 | 24th Mar 2016 |
| Re base lined by pulling 1.3.1 | Basavaraja Ganeshappa | 2.0 | 25th Jul 2016 |
| Implementation of SF007A v2.0.0 & v2.1.0 | Krishna Anne | 3.0 | 3rd Oct 2016 |
| Updated to design version 2.2.0 | TATA | 4.0 | 05-Dec-16 |
| Updated to design version 2.4.0 | KK | 5.0 | 28-Feb-17 |
| Updated Diagram and added Unit Test Consideration | ML | 6.0 | 04-Oct-17 |

Table of Contents

[1 Introduction 6](#_Toc468713268)

[1.1 Purpose 6](#_Toc468713269)

[2 SysFricLrng High-Level Description 7](#_Toc468713270)

[3 Design details of software module 8](#_Toc468713271)

[3.1 Graphical representation of SysFricLrng 8](#_Toc468713272)

[3.2 Data Flow Diagram 9](#_Toc468713273)

[3.2.1 Component level DFD 9](#_Toc468713274)

[3.2.2 Function level DFD 9](#_Toc468713275)

[4 Constant Data Dictionary 10](#_Toc468713276)

[4.1 Program (fixed) Constants 10](#_Toc468713277)

[4.1.1 Embedded Constants 10](#_Toc468713278)

[5 Software Component Implementation 11](#_Toc468713279)

[5.1 Sub-Module Functions 11](#_Toc468713280)

[5.1.1 Init: SysFricLrngInit1 11](#_Toc468713281)

[5.1.1.1 Design Rationale 11](#_Toc468713282)

[5.1.1.2 Module Outputs 11](#_Toc468713283)

[5.1.2 Per: SysFricLrngPer1 11](#_Toc468713284)

[5.1.2.1 Design Rationale 11](#_Toc468713285)

[5.1.2.2 Store Module Inputs to Local copies 11](#_Toc468713286)

[5.1.2.3 (Processing of function)……… 11](#_Toc468713287)

[5.1.2.4 Store Local copy of outputs into Module Outputs 11](#_Toc468713288)

[5.2 Server Runnables 11](#_Toc468713289)

[5.2.1 Server Runnable Name 11](#_Toc468713290)

[5.2.1.1 Design Rationale 11](#_Toc468713291)

[5.2.1.2 (Processing of function)……… 11](#_Toc468713292)

[5.3 Server Runnables 12](#_Toc468713293)

[5.3.1 Server Runnable Name 12](#_Toc468713294)

[5.3.1.1 Design Rationale 12](#_Toc468713295)

[5.3.1.2 (Processing of function)……… 12](#_Toc468713296)

[5.3.2 Server Runnable Name 12](#_Toc468713297)

[5.3.2.1 Design Rationale 12](#_Toc468713298)

[5.3.2.2 (Processing of function)……… 12](#_Toc468713299)

[5.3.3 Server Runnable Name 12](#_Toc468713300)

[5.3.3.1 Design Rationale 12](#_Toc468713301)

[5.3.3.2 (Processing of function)……… 12](#_Toc468713302)

[5.3.4 Server Runnable Name 12](#_Toc468713303)

[5.3.4.1 Design Rationale 12](#_Toc468713304)

[5.3.4.2 (Processing of function)……… 12](#_Toc468713305)

[5.3.5 Server Runnable Name 13](#_Toc468713306)

[5.3.5.1 Design Rationale 13](#_Toc468713307)

[5.3.5.2 (Processing of function)……… 13](#_Toc468713308)

[5.3.6 Server Runnable Name 13](#_Toc468713309)

[5.3.6.2 (Processing of function)……… 13](#_Toc468713310)

[5.3.7 Server Runnable Name 13](#_Toc468713311)

[5.3.7.2 (Processing of function)……… 13](#_Toc468713312)

[5.4 Interrupt Functions 13](#_Toc468713313)

[5.4.1 Interrupt Function Name 13](#_Toc468713314)

[5.4.1.1 Design Rationale 13](#_Toc468713315)

[5.4.1.2 (Processing of the ISR function)….. 13](#_Toc468713316)

[5.5 Module Internal (Local) Functions 14](#_Toc468713317)

[5.5.1 Local Function #1 14](#_Toc468713318)

[5.5.1.1 Design Rationale 14](#_Toc468713319)

[5.5.1.2 Processing 14](#_Toc468713320)

[5.5.2 Local Function #2 14](#_Toc468713321)

[5.5.2.1 Design Rationale 14](#_Toc468713322)

[5.5.2.2 Processing 14](#_Toc468713323)

[5.5.3 Local Function #3 15](#_Toc468713324)

[5.5.3.1 Design Rationale 15](#_Toc468713325)

[5.5.3.2 Processing 15](#_Toc468713326)

[5.5.4 Local Function #4 15](#_Toc468713327)

[5.5.4.1 Design Rationale 15](#_Toc468713328)

[5.5.4.2 Processing 15](#_Toc468713329)

[5.5.5 Local Function #5 16](#_Toc468713330)

[5.5.5.1 Design Rationale 16](#_Toc468713331)

[5.5.5.2 Processing 16](#_Toc468713332)

[5.5.6 Local Function #6 16](#_Toc468713333)

[5.5.6.1 Design Rationale 16](#_Toc468713334)

[5.5.6.2 Processing 16](#_Toc468713335)

[5.5.7 Local Function #7 16](#_Toc468713336)

[5.5.7.1 Design Rationale 17](#_Toc468713337)

[5.5.7.2 Processing 17](#_Toc468713338)

[5.5.8 Local Function #8 17](#_Toc468713339)

[5.5.8.1 Design Rationale 17](#_Toc468713340)

[5.5.8.2 Processing 17](#_Toc468713341)

[5.5.8.3 17](#_Toc468713342)

[5.5.9 Local Function #9 17](#_Toc468713343)

[5.5.9.1 Design Rationale 18](#_Toc468713344)

[5.5.9.2 Processing 18](#_Toc468713345)

[5.5.10 Local Function #10 18](#_Toc468713346)

[5.5.10.1 Design Rationale 18](#_Toc468713347)

[5.5.10.2 Processing 18](#_Toc468713348)

[5.5.11 Local Function #11 18](#_Toc468713349)

[5.5.11.1 Design Rationale 18](#_Toc468713350)

[5.5.11.2 Processing 19](#_Toc468713351)

[5.5.12 Local Function #12 19](#_Toc468713352)

[5.5.12.1 Design Rationale 19](#_Toc468713353)

[5.5.12.2 Processing 19](#_Toc468713354)

[5.6 GLOBAL Function/Macro Definitions 19](#_Toc468713355)

[6 Known Limitations with Design 20](#_Toc468713356)

[7 UNIT TEST CONSIDERATION 21](#_Toc468713357)

[Appendix A Abbreviations and Acronyms 22](#_Toc468713358)

[Appendix B Glossary 23](#_Toc468713359)

[Appendix C References 24](#_Toc468713360)

# Introduction

## Purpose

MDD for System Friction Learning

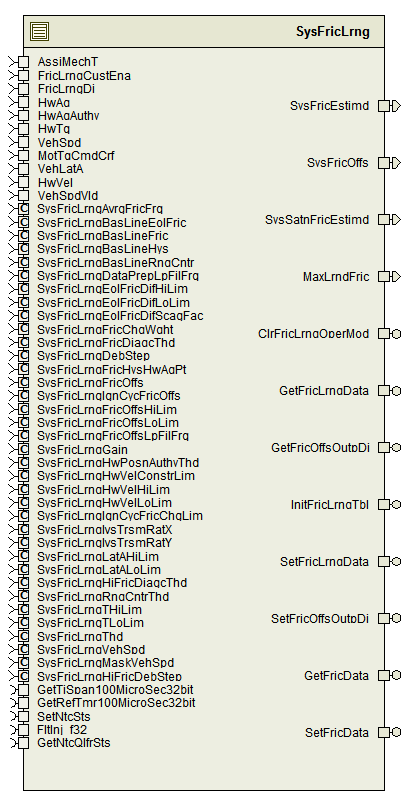
# SysFricLrng High-Level Description

*Refer FDD*

# Design details of software module

*Refer FDD*

## Graphical representation of SysFricLrng



## Data Flow Diagram

*Refer FDD*

### Component level DFD

*Refer FDD*

### Function level DFD

*Refer FDD*

# Constant Data Dictionary

## Program (fixed) Constants

### Embedded Constants

#### Local Constants

|  |  |  |  |
| --- | --- | --- | --- |
| Constant Name | Resolution | Units | Value |
| INDEX0\_CNT\_U08 | 1 | CNT | 0U |
| INDEX1\_CNT\_U08 | 1 | CNT | 1U |
| INDEX2\_CNT\_U08 | 1 | CNT | 2U |
| INDEX3\_CNT\_U08 | 1 | CNT | 3U |
| SYSSATNFRICESTIMDMIN\_HWNWMTR\_F32 | 1 | HwNwMtr | 0.0F |
| SYSSATNFRICESTIMDMAX\_HWNWMTR\_F32 2 | 1 | HwNwMtr | 20.0F |
| SYSFRICESTIMDMIN\_HWNWMTR\_F32 | 1 | HwNwMtr | 0.0F |
| SYSFRICESTIMDMAX\_HWNWMTR\_F32 | 1 | HwNwMtr | 20.0F |
| SYSFRICOFFSMIN\_HWNWMTR\_F32 | 1 | HwNwMtr | -5.0F |
| SYSFRICOFFSMAX\_HWNWMTR\_F32 | 1 | HwNwMtr | 5.0F |

For rest of the constants, please refer Data Dictionary

# Software Component Implementation

The detailed design of the function is provided in the FDD.

## Sub-Module Functions

## Init: SysFricLrngInit1

## Design Rationale

*In MDD, filters are initialized inside the for loop using switch case but in code filters are initialized one by one without any conditions.*

*In model, filters are initialized twice as it is not possible to use a variable for the filter initialization in the model. This is redundancy is not present in the code as variables are used for initializing the filters.*

## Module Outputs

*Refer FDD*

## Per: SysFricLrngPer1

## Design Rationale

*Refer FDD*

## Store Module Inputs to Local copies

*Refer FDD*

## (Processing of function)………

*Refer FDD*

## Store Local copy of outputs into Module Outputs

*Refer FDD*

## Server Runnables

## Server Runnable Name

*ClrFricLrngOperMod*

## Design Rationale

*Refer FDD*

## (Processing of function)………

*On server invocation call*

## Server Runnables

## Server Runnable Name

*GetFricLrngData*

## Design Rationale

*Refer FDD*

## (Processing of function)………

*On server invocation call*

## Server Runnable Name

*GetFricOffsOutpDi*

## Design Rationale

*Refer FDD*

## (Processing of function)………

*On server invocation call*

## Server Runnable Name

*InitFricLrngTbl*

## Design Rationale

*Refer FDD*

## (Processing of function)………

*On server invocation call*

## Server Runnable Name

*SetFricLrngDatal*

## Design Rationale

*Refer FDD*

## (Processing of function)………

*On server invocation call*

## Server Runnable Name

*SetFricOffsOutpDi*

## Design Rationale

*Refer FDD*

## (Processing of function)………

*On server invocation call*

## Server Runnable Name

#### *GetFricData*

#### Design Rationale

*Refer FDD*

To avoid calculating array indexing for updating PIMs Rte\_Pim\_FricLrngData()->Hys and Rte\_Pim\_FricLrngData()->RngCntr, performed casting the array argument back to it's actual type (similar to what we do with cal arrays) so we can use normal indexing.

## (Processing of function)………

*On server invocation call*

## Server Runnable Name

#### *SetFricData*

#### Design Rationale

*Refer FDD*

To avoid calculating array indexing for updating from PIMs Rte\_Pim\_FricLrngData()->Hys and Rte\_Pim\_FricLrngData()->RngCntr, performed casting the array argument back to it's actual type (similar to what we do with cal arrays) so we can use normal indexing.

## (Processing of function)………

*On server invocation call*

## Server Runnable Name

*FricLrngShtDwn*

## Design Rationale

*Refer FDD*

## (Processing of function)………

## Interrupt Functions

*None*

## Interrupt Function Name

*None*

## Design Rationale

*NA*

## (Processing of the ISR function)…..

*NA*

## Module Internal (Local) Functions

## Local Function #1

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Function Name** | FricLearning | Type | Min | Max |
| **Arguments Passed** | SelHwAg\_HwDeg\_T\_f32 | Float32 | -1440.0 | 1440.0 |
| SelColTq\_HwNwtMtr\_T\_f32 | Float32 | -10 | 10 |
| VehSpdIdx\_Cnt\_T\_u16 | Uint16 | 0 | 3 |
| HwVelDir\_Cnt\_T\_u08 | Uint8 | 0 | 1 |
|  | LrngEna\_Cnt\_T\_Logl | Boolean | FALSE | TRUE |
| **Return Value** | NA | NA | NA | NA |

## Design Rationale

## Processing

Refer to ‘FricLearning’ subsystem in FDD.

Following per instance data is updated.

|  |
| --- |
| \*Rte\_Pim\_RawAvrg() (Min:0, Max:20) |
| Rte\_Pim\_SatnAvrgFric()[VehSpdIdx\_Cnt\_T\_u16] (Min:0, Max:20) |

Also writes the outputs SysFricEstimd and SysSatnFricEstimd

## Local Function #2

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Function Name** | RunningAndCalibrationModes | Type | Min | Max |
| **Arguments Passed** | \*FricOffs\_HwNwtMtr\_T\_f32 | Float32 | -5.0 | +5.0 |
|  | \*LrngEna\_Cnt\_T\_Logl | Boolean | FALSE | TRUE |
| **Return Value** | None | NA | NA | NA |

## Design Rationale

## Processing

Following PIMs are updated; refer to ‘RunningAndCalibrationModes’ subsystem in the FDD. FricOffs\_HwNwtMtr\_T\_f32 is the output of this function

|  |
| --- |
| Rte\_Pim\_FricLrngData()->FricOffs (Min:-5, Max:5) |
| \*Rte\_Pim\_RawAvrg() (Min:0, Max:20) |
| Rte\_Pim\_SatnAvrgFric()[VehSpdIdx\_Cnt\_T\_u16] (Min:0, Max:20) |

Also updates the input argument, \*FricOffs\_HwNwtMtr\_T\_f32.

## Local Function #3

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Function Name** | RawAvrgCalc | Type | Min | Max |
| **Arguments Passed** | VehSpdIdx\_Cnt\_T\_u16 | Uint16 | 0 | 5 |
| DeltaIdxOffsDec\_Cnt\_T\_u16 | Uint16 | 0 | 12 |
| DeltaIdxOffsInc\_Cnt\_T\_u16 | Uint16 | 0 | 13 |
| TotalCounter\_Cnt\_T\_u32 | Uint32 | 0 | 65535 |
|  | LrngEna\_Cnt\_T\_Logl | Boolean | FALSE | TRUE |
| **Return Value** | NA | NA | NA | NA |

## Design Rationale

## Processing

Refer to ‘Raw Average Calculation’ subsystem in FDD.

Following per instance data is updated.

|  |
| --- |
| \*Rte\_Pim\_RawAvrg() (Min:0, Max:20) |
| Rte\_Pim\_SatnAvrgFric()[VehSpdIdx\_Cnt\_T\_u16] (Min:0, Max:20) |

## Local Function #4

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Function Name** | PhiCalc | Type | Min | Max |
| **Arguments Passed** | SelHwAg\_HwDeg\_T\_f32 | Float32 | -1440 | 1440 |
| Gate\_Cnt\_T\_u16 | Uint16 | 0 | 65535 |
| DeltaIdxOffs\_Cnt\_T\_u16 | Uint16 | 0 | 10 |
| SelColTq\_HwNwtMtr\_T\_f32 | Float32 | -10 | 10 |
| **Return Value** | NA | NA | NA | NA |

## Design Rationale

## Processing

Refer to ‘Raw Average Calculation’ subsystem in FDD.

Following per instance data is updated.

|  |
| --- |
| Rte\_Pim\_FricLrngData()->Hys[DeltaIdxOffs\_Cnt\_T\_u16][Gate\_Cnt\_T\_u16 + 1U] (Min:-127, Max:127) |
| Rte\_Pim\_FricLrngData()->Hys[DeltaIdxOffs\_Cnt\_T\_u16][Gate\_Cnt\_T\_u16] (Min:-127, Max:127) |

## Local Function #5

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Function Name** | RangeCounterManager | Type | Min | Max |
| **Arguments Passed** | DeltaIdxOffs\_Cnt\_T\_u16 | Uint16 | 0 | 10 |
| DeltaIdxOffsDec\_Cnt\_T\_u16 | Uint16 | 0 | 12 |
| DeltaIdxOffsInc\_Cnt\_T\_u16 | Uint16 | 0 | 13 |
| Gate\_Cnt\_T\_u16 | Uint16 | 0 | 65535 |
| **Return Value** | NA | NA | NA | NA |

## Design Rationale

## Processing

Refer to ‘Range counter manager’ subsystem in FDD.

Following per instance data is updated.

|  |
| --- |
| \*Rte\_Pim\_ RngCntrThdExcdd() (Min:0, Max:1) |
| Rte\_Pim\_FricLrngData->RngCntr (:,:) (Min:0, Max:65535) |

## Local Function #6

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Function Name** | NTCSetReset | Type | Min | Max |
| **Arguments Passed** | MaxRawAvrgFric\_Cnt\_T\_f32 | Float32 | -127 | 254 |
| **Return Value** | NA | NA | NA | NA |

## Design Rationale

## Processing

Refer to ‘NTC\_Pass’ and ‘NTC\_Fail’ subsystem in FDD

Sets or resets the NTCNR\_0X0A2

## Local Function #7

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Function Name** | ClearingMode | Type | Min | Max |
| **Arguments Passed** | none | NA | NA | NA |
| **Return Value** | none | NA | NA | NA |

## Design Rationale

## Processing

Refer to ‘Clearing Mode’ subsystem in FDD.

Following per instance data is updated.

|  |
| --- |
| \*Rte\_Pim\_FricOffs()(Min:-5, Max:5) |

## Local Function #8

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Function Name** | ResettingMode | Type | Min | Max |
| **Arguments Passed** | \*FricOffs\_HwNwtMtr\_T\_f32 | NA | NA | NA |
| **Return Value** | None | NA | NA | NA |

## Design Rationale

## Processing

## 

Refer to ‘ResettingMode’ subsystem in FDD.

Following per instance data is updated. Also updates the input argument ‘\*FricOffs\_HwNwtMtr\_T\_f32’.

|  |
| --- |
| Rte\_Pim\_FricLrngData()->RngCntr(;) |
| Rte\_Pim\_AvrgFricLpFil**X**()->FilSt (X: 1 to 4) |
| Rte\_Pim\_FricLrngData()->Hys(;) |
| Rte\_Pim\_FricOffs()(Min:-5, Max:5) |
| Rte\_Pim\_VehBasLineFric()[] (Min:-0, Max:127)  Rte\_Pim\_RawAvrgFric()[] (Min:--127, Max:254)  Rte\_Pim\_FilAvrgFric()[] (Min:--10 , Max: 10)  Rte\_Pim\_SatnAvrgFric()[](Min:--127, Max:254)  Rte\_Pim\_FricLrngData()->VehLrndFric[] (0-127) |

## Local Function #9

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Function Name** | HwAngConstraint | Type | Min | Max |
| **Arguments Passed** | FilHwAg\_HwDeg\_T\_f32 | Float32 | -1440 | 1440 |
|  | \*HwAgOK\_Cnt\_T\_Logl | boolean | 0 | 1 |
|  | \*SelHwAg\_HwDeg\_T\_f32 | Float32 | -1440 | 1440 |
| **Return Value** | NA | NA | NA | NA |

## Design Rationale

 IDXSELN2\_ULS\_U08 is not used in the code because it is not required instead IDXSELN1\_ULS\_U08 serves the purpose.

## Processing

Refer to ‘HwAngConstraint‘ subsystem in FDD. Updates the input arguments, \*HwAgOK\_Cnt\_T\_Logl and \*SelHwAg\_HwDeg\_T\_f32

## Local Function #10

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Function Name** | HwVelConstraint | Type | Min | Max |
| **Arguments Passed** | HwVel\_HwRadPerSec\_T\_f32 | Float32 | -42 | 42 |
| HwVelOK\_Cnt\_T\_Logl | Boolean | 0 | 1 |
| HwVelDir\_Cnt\_T\_u08 | Uint8 | 0 | 1 |
| **Return Value** | NA | NA | NA | NA |

## Design Rationale

## Processing

Refer to ‘HwVelConstraint’ subsystem in FDD.

## Local Function #11

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Function Name** | VehSpdConstraint | Type | Min | Max |
| **Arguments Passed** | VehSpd\_Kph\_T\_f32 | Float32 | 0 | 511 |
|  | \*VehSpdOK\_Cnt\_T\_Logl | Boolean | 0 | 1 |
| \*VehSpdIdx\_Cnt\_T\_u16 | Uint16 | 0 | 5 |
| **Return Value** | None | NA | NA | NA |

## Design Rationale

*Code is optimized due to limitation with the model; hence code completely won’t match the model. There won’t be any impact on the functionality.*

*In the model as it is not possible to break the for loop until the loop iterator reaches the configured constant threshold value, index corresponding to the position in ‘SysFricLrngVehSpd’ which breaches the conditions mentioned in ‘VehSpdIdxCalcn’ subsystem is calculated by successively adding the index value after multiplying it with either the condition true or false based on whether the vehicle speed value breaches the threshold mentioned in the FDD. In code as it is possible to exit the for loop as soon as a value in ‘VehSpdIdxCalcn’ breaches thresholds as mentioned in FDD, no such successive addition of loop counter is required.*

## Processing

Refer to ‘VehSpdConstraint’ subsystem in FDD.

## Local Function #12

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Function Name** | ColTqconstraint | Type | Min | Max |
| **Arguments Passed** | FilColTq\_HwNwtMtr\_T\_f32 | Float32 | -10 | 10 |
| \*SelColTq\_HwNwtMtr\_T\_f32 | Boolean | -10 | 10 |
| **Return Value** | NA | NA | NA | NA |

## Design Rationale

## Processing

Refer to ‘ColTqconstraint’ subsystem in FDD. Updates the \*SelColTq\_HwNwtMtr\_T\_f32.

## GLOBAL Function/Macro Definitions

NA

# Known Limitations with Design

None

# UNIT TEST CONSIDERATION

1. In model, one based indexing is used but in code 0 based indexing is used.
2. In the NVM block needs area of Developer tool, the options of "Restore at Startup" and "Store at Shutdown" are disabled as the newer version (3.13.22 SP2) of this tool throws warnings while doing a DCF check.
3. There will be a source model mismatch that occurs because of a logic change that happened for a PSR. There is limiting that occurs in the new Non Rte Server Runnable. These limits were switched for the PSR but this change was not brought in for the design. An ICR has been submitted to fix the design to match the implementation, EA4#15920.

Abbreviations and Acronyms

| **Abbreviation or Acronym** | **Description** |
| --- | --- |
|  |  |
|  |  |

Glossary

**Note**: Terms and definitions from the source “Nexteer Automotive” take precedence over all other definitions of the same term. Terms and definitions from the source “Nexteer Automotive” are formulated from multiple sources, including the following:

* ISO 9000
* ISO/IEC 12207
* ISO/IEC 15504
* Automotive SPICE® Process Reference Model (PRM)
* Automotive SPICE® Process Assessment Model (PAM)
* ISO/IEC 15288
* ISO 26262
* IEEE Standards
* SWEBOK
* PMBOK
* Existing Nexteer Automotive documentation

| **Term** | **Definition** | **Source** |
| --- | --- | --- |
| MDD | Module Design Document |  |
| DFD | Data Flow Diagram |  |

References

| **Ref. #** | **Title** | **Version** |
| --- | --- | --- |
| 1 | AUTOSAR Specification of Memory Mapping (Link:[AUTOSAR\_SWS\_MemoryMapping.pdf](http://www.autosar.org/download/R4.0/AUTOSAR_SWS_MemoryMapping.pdf)) | Process 4.02.01 |
| 2 | MDD Guideline | Process 4.02.01 |
| 3 | [Software Naming Conventions.doc](http://misagweb01.nexteer.com/eRoomReq/Files/erooms8/NextGeneration/0_fc55f/Software%20Naming%20Conventions%2003x(In%20Work).doc) | 2.0 |
| 4 | [Software Design and Coding Standards.doc](http://eroom1.nexteer.com/eRoomReq/Files/erooms8/NextGeneration/0_1a67a9/Software%20Design%20and%20Coding%20Standards.doc) | 2.1 |
| 5 | FDD- SF007A\_SysFricLrng\_Design | See Synergy sub project version |