**Module Design Document**

**For**

**TunSelnMngt**

**May 14, 2018**

**Prepared For:**

**Software Engineering**

**Nexteer Automotive,**

**Saginaw, MI, USA**

**Prepared By:**

**Kevin Smith,**

**Nexteer Automotive,**

**Saginaw, MI, USAChange History**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Description** | **Author** | **Version** | **Date** | **Approved By** |
| Initial Version | K. Smith | 1 | 07-Apr-16 |  |
| Updated program flow diagrams | N. Saxton | 2 | 06-May-16 |  |
| Updates to flow charts for anomaly EA4#6672 corrections | K. Smith | 3 | 29-Aug-16 |  |
| Updated to match version 2.0.0 of the design | K. Smith | 4 | 29-Nov-17 |  |
| Updated to match version 3.0.0 of the design and anomaly EA4#21421 | K. Smith | 5 | 14-May-18 |  |

Table of Contents

[1 Introduction 5](#_Toc450305901)

[1.1 Purpose 5](#_Toc450305902)

[1.2 Scope 5](#_Toc450305903)

[2 TunSelnMngt & High-Level Description 6](#_Toc450305904)

[3 Design details of software module 7](#_Toc450305905)

[3.1 Graphical representation of TunSelnMngt 7](#_Toc450305906)

[3.2 Data Flow Diagram 7](#_Toc450305907)

[3.2.1 Component level DFD 7](#_Toc450305908)

[3.2.2 Function level DFD 7](#_Toc450305909)

[4 Constant Data Dictionary 8](#_Toc450305910)

[4.1 Program (fixed) Constants 8](#_Toc450305911)

[4.1.1 Embedded Constants 8](#_Toc450305912)

[4.2 Variable Data Dictionary 8](#_Toc450305913)

[4.2.1 User Defined Typedef Definition/Declaration 8](#_Toc450305914)

[4.2.1.1 Static Structures 8](#_Toc450305915)

[4.2.1.2 Dynamic Structures and Enums 9](#_Toc450305916)

[4.2.2 User Defined Enumerated Types 10](#_Toc450305917)

[5 Software Component Implementation 11](#_Toc450305918)

[5.1 Sub-Module Functions 11](#_Toc450305919)

[5.1.1 Init: TunSelnMngtInit1 11](#_Toc450305920)

[5.1.2 Per: TunSelnMngtPer1 13](#_Toc450305921)

[5.2 Server Runnables 15](#_Toc450305922)

[5.2.1 CopyCalPageReq 15](#_Toc450305923)

[5.2.2 GetCalPageReq 16](#_Toc450305924)

[5.2.3 GetSegInfoReq 17](#_Toc450305925)

[5.2.4 OnlineTunRamAdrMpg 18](#_Toc450305926)

[5.2.5 SetCalPageReq 19](#_Toc450305927)

[5.3 Interrupt Functions 20](#_Toc450305928)

[5.4 Module Internal (Local) Functions 20](#_Toc450305929)

[5.4.1 SwtCalIdx 20](#_Toc450305930)

[5.4.2 IdxChgMngt 21](#_Toc450305931)

[5.4.3 MemCopy32Bit 23](#_Toc450305932)

[5.4.4 MemCopy8Bit 24](#_Toc450305933)

[5.4.5 SegModAdrInfo 25](#_Toc450305934)

[5.4.6 SegModStdInfo 26](#_Toc450305935)

[5.4.7 SegModAdrMpg 27](#_Toc450305936)

[5.5 GLOBAL Function/Macro Definitions 28](#_Toc450305937)

[6 Known Limitations with Design 29](#_Toc450305938)

[7 UNIT TEST CONSIDERATION 30](#_Toc450305939)

[Appendix A Abbreviations and Acronyms 31](#_Toc450305940)

[Appendix B Glossary 32](#_Toc450305941)

[Appendix C References 33](#_Toc450305942)

# Introduction

## Purpose

This MDD aids in documenting the implementation of ES400A Tuning Selection Management.

## Scope

The following definitions are used throughout this document:

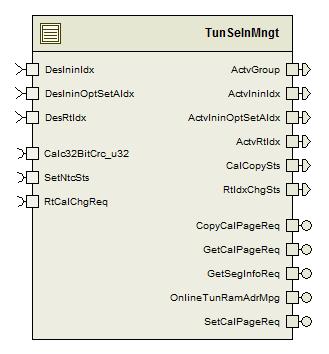
* **Shall**: indicates a mandatory requirement without exception in compliance.
* **Should**: indicates a mandatory requirement; exceptions allowed only with documented justification.
* **May**: indicates an optional action.

# TunSelnMngt & High-Level Description

Tuning Selection Management (TunSelnMngt) provides the ability to change run-time calibrations during operation. The component also provides a RAM buffer for online calibration changes over XCP for tuning processes.

# Design details of software module

## Graphical representation of TunSelnMngt



## Data Flow Diagram

None

### Component level DFD

None

### Function level DFD

None

# Constant Data Dictionary

## Program (fixed) Constants

### Embedded Constants

#### Local Constants

Values between the brackets [] are the ranges that the configurable constants could be defined as for a given integration. These values are generated by Configurator before the software build.

|  |  |  |  |
| --- | --- | --- | --- |
| Constant Name | Resolution | Units | Value |
| MAXININIDXCNT\_CNT\_U08 | Uint8 | Cnt | [0-255] |
| MAXININOPTSETAIDXCNT\_CNT\_U08 | Uint8 | Cnt | [0-255] |
| MAXRTIDXCNT\_CNT\_U08 | Uint8 | Cnt | [0-255] |
| MAXONLINECALCFGCNT\_CNT\_U08 | Uint8 | Cnt | [0-255] |
| ONLINECALGROUPS\_CNT\_U08 | Uint8 | Cnt | [0-255] |
| ONLINECALRAMTBL\_CNT\_U16 | Uint16 | Cnt | [0-65535] |
| PRMPTRTBLSIZEINWORD\_CNT\_U16 | Uint16 | Cnt | [0-65535] |

## Variable Data Dictionary

The following type definitions are found in the private header of this component.

### User Defined Typedef Definition/Declaration

## Static Structures

The following table contains structures are static in their definition. However, internal elements may change based on the configuration of the project but the high level content is the same. These elements are described at the bottom of the table.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Typedef Name | Element Name | User Defined Type | Legal Range  (min) | Legal Range  (max) |
| TunSelnRamTblRec1 | PrmRefTblPtr | Rte\_ParameterRefTabType | N/A | N/A |
|  | PrmTblCrc\_u32 | Uint32 | 0 | 4294967295 |
| TunSelnIdxTblRec1 | SrcIdx\_u16 | Uint16 | 0 | 65535 |
|  | DestIdx\_u16 | Uint16 | 0 | 65535 |
|  | SigIdx\_u08 | Uint8 | 0 | 255 |
| TunSelnOnlineCalIdxTblRec1 | RamStructPtr\_u08 | UInt8\* | 0 | 255 |
|  | StructSize\_u16 | Uint16 | 0 | 65535 |
|  | TblIdx\_u16 | Uint16 | 0 | 65535 |
|  | GroupIdx\_u08 | Uint8 | 0 | 255 |

Rte\_ParameterRefTabType is a structure of void pointers that point to the various calibration component structures. This type is generated by the RTE based on the configuration of the integration.

## Dynamic Structures and Enums

The following table contains structures are dynamic in their definition. The contents contained will vary from project to project. The intent of this table is to document their purpose.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Entry Type | Typedef Name | Example Element(s) | Variable Type | Value | Comment |
| Enum | OnlineCalGroup1 | GroupA | N/A | 0 | This enum contains a number representation for each calibration group that is generated for online calibration from A=0 to Z=26. Groups are defined by a letter suffix. |
|  |  | GroupB | N/A | 1 |
| Struct | <GroupName>Typ  GroupATyp | <Cal Region><Cal Cmp Name><Online Group>  CalRegn01CmnGroupA | RTE Generated | N/A | This structure is generated based on the online groups configured. Each group will generate a structure and contain all calibration software components within that group. |
| Union | OnlineCalTblRec1 | Byte[<RAM Size>] | Unit8 | N/A | This structure combines all the online calibration groups into data type for assignment to the RAM buffer. The ‘byte’ element is used to access the entire buffer on a byte-by-byte basis and ensure that the RAM is properly sized. |
|  |  | <GroupName> | <GroupName>Typ | N/A |

### User Defined Enumerated Types

|  |  |  |
| --- | --- | --- |
| Enum Name | Element Name | Value |
| GetSegModeSegInfo1 | GETSEGMODSEGINFO\_ADR | 0 |
|  | GETSEGMODSEGINFO\_LEN | 1 |
| GetSegModeMpgIdx1 | GETSEGMODMPGIDX\_SRCADR | 0 |
|  | GETSEGMODMPGIDX\_DESTADR | 1 |
|  | GETSEGMODMPGIDX\_LEN | 2 |

# Software Component Implementation

## Sub-Module Functions

### Init: Init1

#### Design Rationale

The initialization function creates two copies of the RTE generated flash table in to RAM. A CRC is performed on each of the copies to ensure that they match. The function also adjusts the tables to load the appropriate initialization and run-time calibrations indexes that are required by the application.

#### Processing





#### Module Outputs

None

### Per: Per1

#### Design Rationale

#### Processing





#### Module Outputs

None

## Server Runnables

### CopyCalPageReq

#### Design Rationale

This function is called by the XCP master and queues the copy of the calibrations contained in the selected group, or segment, into the RAM buffer. The actual copy is performed by TunSelnMngt’s main periodic function, but this runnable logs the current state of TunSelnMngt and marks the copy in progress.

#### (Processing of function)………



#### Module Outputs

None

### GetCalPageReq

#### Design Rationale

This function is called by the XCP master and returns the page with the request XCP and ECU access for a given segment.

#### (Processing of function)………



#### Module Outputs

None

### GetSegInfoReq

#### Design Rationale

This function is called by the XCP master and returns the requested information for the provided segment.

#### (Processing of function)………



#### Module Outputs

None

### OnlineTunRamAdrMpg

#### Design Rationale

This function is called by the XCP master. During tuning, tools such as eTool and CANape will read calibration values from their flash addresses because that is how the A2L file is defined. However, when the calibrations are access from RAM, the user does not always know the exact address the calibration is located. This function calculates the RAM address for a given calibration to the XCP function for reading and writing.

#### (Processing of function)………



#### Module Outputs

None

### SetCalPageReq

#### Design Rationale

This function is called by the XCP master. This function will set the status of the calibration page for a given segment.

#### (Processing of function)………



#### Module Outputs

None

## Interrupt Functions

None

## Module Internal (Local) Functions

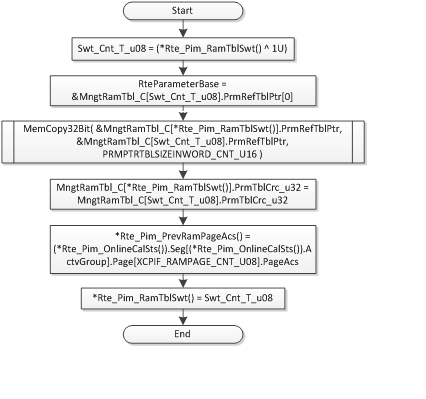
### SwtCalIdx

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Function Name** | SwtCalIdx | Type | Min | Max |
| **Arguments Passed** | N/A |  |  |  |
| **Return Value** | N/A |  |  |  |

#### Design Rationale

This function manages the RAM buffer access and switches the calibration index between the two copies in RAM.

#### Processing



### IdxChgMngt

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Function Name** | IdxChgMngt | Type | Min | Max |
| **Arguments Passed** | SeldIdx\_Cnt\_T\_u08 | Uint8 | 0 | 255 |
|  | GendCalTblSize\_Cnt\_T\_u08 | Uint8 | 0 | 255 |
|  | GendCalTbl\_T\_rec | TunSelnIdxTblRec1\* | 0 | 4294967295 |
| **Return Value** | IdxFound\_Cnt\_T\_logl | Boolean | FALSE | TRUE |

#### Design Rationale

#### Processing







### MemCopy32Bit

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Function Name** | MemCopy32Bit | Type | Min | Max |
| **Arguments Passed** | Des\_Arg | Void | 0 | 4294967295 |
|  | Src\_Arg | Void | 0 | 4294967295 |
|  | Len\_Arg | Uint16 | 0 | 65535 |
| **Return Value** | N/A |  |  |  |

#### Design Rationale

The 32-bit mem copy function is used to move calibration pointers from flash to RAM. The void pointers are internally assigned to a uint32 pointer before the processing loop begins.

#### Processing



### MemCopy8Bit

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Function Name** | MemCopy8Bit | Type | Min | Max |
| **Arguments Passed** | Des\_Arg | Void | 0 | 4294967295 |
|  | Src\_Arg | Void | 0 | 4294967295 |
|  | Len\_Arg | Uint16 | 0 | 65535 |
| **Return Value** | N/A |  |  |  |

#### Design Rationale

The 8-bit mem copy function is used to move calibration segments into the RAM space. Since the length of the segments is not guaranteed to be a 32-bit even address, 8-bit was selected to ensure that only the bytes required to be moved are performed. The void pointers are internally assigned to a uint8 pointer before the processing loop begins.

#### Processing



### SegModAdrInfo

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Function Name** | SegModAdrInfo | Type | Min | Max |
| **Arguments Passed** | Seg\_Arg | UInt8 | 0 | 255 |
|  | SegInfo\_Arg | GetSegModSegInfo1 | 0 | 1 |
|  | Resp\_Arg | Uint8\* | 0 | 255 |
|  | RespLen\_Arg | Uint8\* | 8 | 8 |
| **Return Value** | Rtn\_Cnt\_T\_u08 | Uint8 | 0 | 255 |

#### Design Rationale

#### Processing



### SegModStdInfo

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Function Name** | SegModStdInfo | Type | Min | Max |
| **Arguments Passed** | Seg\_Arg | Uint8 | 0 | 255 |
|  | Resp\_Arg | Uint8\* | 0 | 255 |
|  | RespLen\_Arg | UInt8\* | 6 | 6 |
| **Return Value** | <Hard Coded> | UInt8 | 1 | 1 |

#### Design Rationale

#### Processing



### SegModAdrMpg

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Function Name** | SegModAdrMpg | Type | Min | Max |
| **Arguments Passed** | Seg\_Arg | Uint8 | 0 | 255 |
|  | MpgIdxInfo\_Arg | GetSegModMpgIdx1 | 0 | 2 |
|  | MpgIdx\_Arg | Uint8 | 0 | 255 |
|  | Resp\_Arg | UInt8\* | 0 | 4294967295 |
|  | RespLen\_Arg | Uint8\* | 8 | 8 |
| **Return Value** | Rtn\_Cnt\_T\_u08 | Uint8 | 0 | 255 |

#### Design Rationale

#### Processing



### TunSelnMngt\_ChkXcpWrAcs

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Function Name** | TunSelnMngt\_ChkXcpWrAcs | Type | Min | Max |
| **Arguments Passed** | ReqAdr\_Cnt\_T\_u32 | Uint32 | 0 | 4294967296 |
| **Return Value** | Rtn\_Cnt\_T\_logl | Boolean | FALSE | TRUE |

#### Design Rationale

This function is generated based on the configured access regions in Configurator. Below is just a high level overview of the function. The conditions in the if statements are logically OR’d together to form the tests.

#### Processing



## GLOBAL Function/Macro Definitions

None

# Known Limitations with Design

1. Since the design relies heavily on the integration project to generate the RTE and calibrations, the component cannot simply have those settings available at this level. Attached are the settings used to generate the files in the local folder for testing with the compiler and also configurator settings.

|  |  |
| --- | --- |
| File | Use |
|  | Import into Configurator |
|  | Place contents at the end of the Rte\_Type.h file. Note: This file is based on the Configurator settings in this table. |

# UNIT TEST CONSIDERATION

None

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)) | v1.3.0 R4.0 Rev 2 |
| 2 | MDD Guideline | EA4 01.00.01 |
| 3 | [Software Naming Conventions.doc](http://misagweb01.nexteer.com/eRoomReq/Files/erooms8/NextGeneration/0_fc55f/Software%20Naming%20Conventions%2003x(In%20Work).doc) | 1.0 |
| 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 |