**Module Design Document**

**For**

**Motor Angle 0 Measurement**

**May 04, 2018**

**Prepared For:**

**Software Engineering**

**Nexteer Automotive,**

**Saginaw, MI, USA**

**Prepared By:**

**Software Group,**

**Nexteer Automotive,**

**Saginaw, MI, USAChange History**

|  |  |  |  |
| --- | --- | --- | --- |
| **Description** | **Author** | **Version** | **Date** |
| Initial Version | Avinash James | 1.0 | 06-Jun-2016 |
| Removed local function CalcTurnCntr | Avinash James | 2.0 | 13-Jun-2016 |
| Updated as per Design version 2.0.0 | Krzysztof Byrski | 3.0 | 19-Oct-2017 |
| Updated as per Design version 5.0.0 | Krzysztof Byrski | 4.0 | 25-Apr-2018 |
| Updated the diagram for IO port additions | Avinash James | 5.0 | 27-Apr-2018 |
| Added updates for sensor offset learning | Avinash James | 6.0 | 04-May-2018 |

Table of Contents

[1 Introduction 5](#_Toc512416009)

[1.1 Purpose 5](#_Toc512416010)

[1.2 Scope 5](#_Toc512416011)

[2 MotAg0MeasHigh-Level Description 6](#_Toc512416012)

[3 Design details of software module 7](#_Toc512416013)

[3.1 Graphical representation of MotAg0Meas 7](#_Toc512416014)

[3.2 Data Flow Diagram 7](#_Toc512416015)

[3.2.1 Component level DFD 7](#_Toc512416016)

[3.2.2 Function level DFD 7](#_Toc512416017)

[4 Constant Data Dictionary 8](#_Toc512416018)

[4.1 Program (fixed) Constants 8](#_Toc512416019)

[4.1.1 Embedded Constants 8](#_Toc512416020)

[5 Software Component Implementation 9](#_Toc512416021)

[5.1 Sub-Module Functions 9](#_Toc512416022)

[5.1.1 Init: MotAg0MeasInit1 9](#_Toc512416023)

[5.1.1.1 Design Rationale 9](#_Toc512416024)

[5.1.1.2 Module Outputs 9](#_Toc512416025)

[5.1.1.3 Module Internal 9](#_Toc512416026)

[5.1.2 Per: MotAg0MeasPer1 9](#_Toc512416027)

[5.1.2.1 Design Rationale 9](#_Toc512416028)

[5.1.2.2 Store Module Inputs to Local copies 9](#_Toc512416029)

[5.1.2.3 (Processing of function)……… 9](#_Toc512416030)

[5.1.2.4 Store Local copy of outputs into Module Outputs 9](#_Toc512416031)

[5.1.3 Per: MotAg0MeasPer2 9](#_Toc512416032)

[5.1.3.1 Design Rationale 9](#_Toc512416033)

[5.1.3.2 Store Module Inputs to Local copies 9](#_Toc512416034)

[5.1.3.3 (Processing of function)……… 10](#_Toc512416035)

[5.1.3.4 Store Local copy of outputs into Module Outputs 10](#_Toc512416036)

[5.1.4 Per: MotAg0MeasPer3 10](#_Toc512416037)

[5.1.4.1 Design Rationale 10](#_Toc512416038)

[5.1.4.2 Store Module Inputs to Local copies 10](#_Toc512416039)

[5.1.4.3 (Processing of function)……… 10](#_Toc512416040)

[5.1.4.4 Store Local copy of outputs into Module Outputs 10](#_Toc512416041)

[5.2 Server Runables: MotAg0CoeffTblRead 10](#_Toc512416042)

[5.2.1.1 Design Rationale 10](#_Toc512416043)

[5.2.1.2 Store Module Inputs to Local copies 10](#_Toc512416044)

[5.2.1.3 (Processing of function)……… 10](#_Toc512416045)

[5.2.1.4 Store Local copy of outputs into Module Outputs 10](#_Toc512416046)

[5.3 Server Runables: MotAg0CoeffTblWr 10](#_Toc512416047)

[5.3.1.1 Design Rationale 10](#_Toc512416048)

[5.3.1.2 Store Module Inputs to Local copies 10](#_Toc512416049)

[5.3.1.3 (Processing of function)……… 10](#_Toc512416050)

[5.3.1.4 Store Local copy of outputs into Module Outputs 11](#_Toc512416051)

[5.4 Interrupt Functions 11](#_Toc512416052)

[5.5 Module Internal (Local) Functions 12](#_Toc512416053)

[5.5.1 ProcessErrorRegAndDieRevCtr 12](#_Toc512416054)

[5.5.2 SPI\_AnglePolarityAdjust 12](#_Toc512416055)

[5.5.3 SPIvsENCA 12](#_Toc512416056)

[5.5.4 CalcCorrnTbl 13](#_Toc512416057)

[5.5.5 MotAgFaultProcessing 13](#_Toc512416058)

[5.5.6 CalcNtcPrm 14](#_Toc512416059)

[5.5.7 SetMotAg0FltNtc 14](#_Toc512416060)

[5.5.8 OffsetCalculation 15](#_Toc512416061)

[5.5.9 CalculateMotAgTurnCntr 15](#_Toc512416062)

[5.5.10 SPI\_AngleRawProcess 16](#_Toc512416063)

[5.5.11 CompensateMechMtrPos 16](#_Toc512416064)

[5.6 GLOBAL Function/Macro Definitions 17](#_Toc512416065)

[6 Known Limitations with Design 18](#_Toc512416066)

[7 UNIT TEST CONSIDERATION 19](#_Toc512416067)

[Appendix A Abbreviations and Acronyms 20](#_Toc512416068)

[Appendix B Glossary 21](#_Toc512416069)

[Appendix C References 22](#_Toc512416070)

# Introduction

## Purpose

This document defines the module level design for the Sensor Offset and Correction Component. Major part of design has been captured in the FDD and any design rationale that has not been identified in the FDD and has been used to implement the component has been documented in the MDD

## 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.

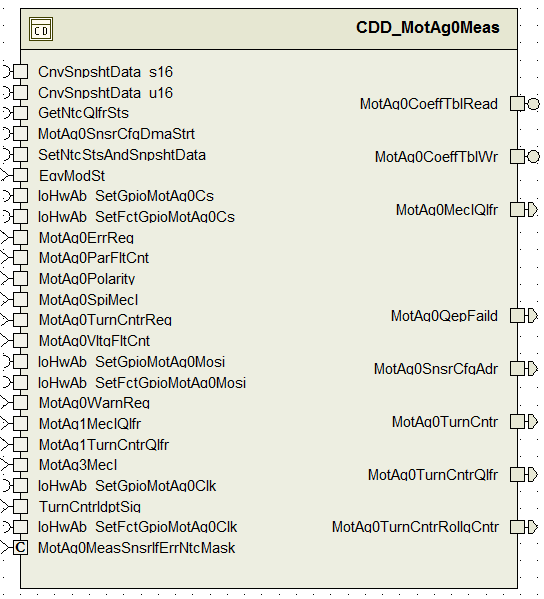
# MotAg0MeasHigh-Level Description

The CDD\_MotAg0Meas component is the complex driver for the motor angle 1 measurement subsystem. This function initializes the registers for CSIH3 SPI channel for communicating with the motor angle 1 measurement sensor board. The SPI transmission is triggered periodically by DMA component. This function receives the RAW sensor data at 62.5uS rate. The component contains two source files, both described in this MDD: CDD\_MotAg0Meas.c contains the RTE runnables and services; CDD\_MotAg0Meas\_MotCtrl.c contains the motor control runnable.

# Design details of software module

See FDD.

## Graphical representation of MotAg0Meas



## Data Flow Diagram

See FDD.

### Component level DFD

See FDD.

### Function level DFD

See FDD.

# Constant Data Dictionary

## Program (fixed) Constants

### Embedded Constants

#### Local Constants

|  |  |  |  |
| --- | --- | --- | --- |
| Constant Name | Resolution | Units | Value |
| MOTAG0TURNCNTRLOLIM\_CNT\_F32 | Single point | Cnt | -256.09375 |
| MOTAG0TURNCNTRHILIM\_CNT\_F32 | Single point | Cnt | 255.96875 |
| MOTCTRLMOTAG0WARNREGLOLIM\_CNT\_U32 | 1 | Cnt | 0 |
| MOTCTRLMOTAG0WARNREGHILIM\_CNT\_U32 | 1 | Cnt | 67108863 |
| MOTCTRLMOTAG0ERRREGLOLIM\_CNT\_U32 | 1 | Cnt | 0 |
| MOTCTRLMOTAG0ERRREGHILIM\_CNT\_U32 | 1 | Cnt | 67108863 |
| MOTCTRLMOTAG0TURNCNTRREGLOLIM\_CNT\_U32 | 1 | Cnt | 0 |
| MOTCTRLMOTAG0TURNCNTRREGHILIM\_CNT\_U32 | 1 | Cnt | 67108863 |

**\* Also see FDD – CM620B\_MotAg0Meas\_DataDict.m file**

# Software Component Implementation

## Sub-Module Functions

The sub-module functions are grouped based on similar functionality that needs to be executed in a given “State” of the system (refer States and Modes). For a given module, the MDD will identify the type and number of sub-modules required. The sub-module types are described below.

### Init: MotAg0MeasInit1

## Design Rationale

All register initialization that is allowed at the register level (see Register/Field column of the CM620B\_MotAg0Meas\_RegisterConfiguration.xlsm spreadsheet in the FDD) is done at the register level to save execution time as compared to the read/modify/writes that would be needed to initialize at the field level. Field level initialization done only where required by the spreadsheet.

## Module Outputs

See FDD: MotAg0MeasInit1 model block.

## Module Internal

See FDD: MotAg0MeasInit1 model block for Per Instance Memory.

### Per: MotAg0MeasPer1

## Design Rationale

For run time efficiency in the motor control loop the **Compensate MechMtrPos** block is implemented in a optimized way in the code by letting a uint16 variable be overflown

## Store Module Inputs to Local copies

See FDD: MotAg0MeasPer1 model block

## (Processing of function)………

See FDD: MotAg0MeasPer1 model block.

## Store Local copy of outputs into Module Outputs

See FDD: MotAg0MeasPer1 model block.

### Per: MotAg0MeasPer2

## Design Rationale

None

## Store Module Inputs to Local copies

See FDD: MotAg0MeasPer2 model block

## (Processing of function)………

See FDD: MotAg0MeasPer2 model block.

## Store Local copy of outputs into Module Outputs

See FDD: MotAg0MeasPer2 model block.

### Per: MotAg0MeasPer3

## Design Rationale

None

## Store Module Inputs to Local copies

See FDD: MotAg0MeasPer3 model block

## (Processing of function)………

See FDD: MotAg0MeasPer3 model block.

## Store Local copy of outputs into Module Outputs

See FDD: MotAg0MeasPer3 model block.

## Server Runables: MotAg0CoeffTblRead

## Design Rationale

None

## Store Module Inputs to Local copies

None

## (Processing of function)………

*See* MotAg0CoeffTblRead *block in the FDD*

## Store Local copy of outputs into Module Outputs

None

## Server Runables: MotAg0CoeffTblWr

## Design Rationale

None

## Store Module Inputs to Local copies

None

## (Processing of function)………

*See* MotAg0CoeffTblWr *block in the FDD*

## Store Local copy of outputs into Module Outputs

*See* MotAg0MeasMotAg0CoeffTblWr*block in the FDD*

## Server Runables: MotAg0CfgLoPwrMod

## Design Rationale

None

## Store Module Inputs to Local copies

None

## (Processing of function)………

*See* MotAg0CfgLoPwrMod *block in the FDD*

## Store Local copy of outputs into Module Outputs

*See* MotAg0CfgLoPwrMod *block in the FDD*

## Interrupt Functions

None

## Module Internal (Local) Functions

### ProcessErrorRegAndDieRevCtr

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Function Name** | ProcessErrorRegAndDieRevCtr | Type | Min | Max |
| **Arguments Passed** | MotAgErrReg\_Cnt\_T\_u32 | uint32 | 0 | 67108863 |
|  | MotAgTurnCntrReg\_Cnt\_T\_u32 | uint32 | 0 | 67108863 |
|  | MotAgWarnReg\_Cnt\_T\_u32 | uint32 | 0 | 67108863 |
| **Return Value** | MotAg0Err\_Cnt\_T\_u16 | uint16 | 0 | 65535 |
|  | MotAg0Warn\_Cnt\_T\_u16 | uint16 | 0 | 65535 |
|  | TurnCntrParFltCnt\_Cnt\_T\_u16 | uint16 | 0 | 65535 |
|  | TurnCntr\_Cnt\_T\_s16 | sint16 | -2048 | 2048 |
|  | MotAgTurnCntrRollgCntr\_Cnt\_T\_u08 | uint8 | 0 | 65535 |

#### Design Rationale

Implementation of "Process ErrorReg and DieRevCtr" Simulink block



### SPIvsENCA

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Function Name** | SPIvsENCA | Type | Min | Max |
| **Arguments Passed** | MotAgSpiMecl\_MotRev\_T\_u0p16 | u0p16 | 0 | 65535 |
|  | MotAgEncaMecl\_MotRev\_T\_u0p16 | u0p16 | 0 | 65535 |
| **Return Value** | MotAgSyncErr\_Cnt\_T\_u08 | uint8 | 0 | 1 |

#### Design Rationale

Implementation of "SPI vs ENCA" Simulink block

### CalcCorrnTbl

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Function Name** | CalcCorrnTbl | Type | Min | Max |
| **Arguments Passed** | None | N/A | N/A | N/A |
| **Return Value** | N/A | N/A | N/A | N/A |

#### Design Rationale

See “Calculate Correction Table” block in the Simulink model of the design

### MotAgFaultProcessing

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Function Name** | MotAgFaultProcessing | Type | Min | Max |
| **Arguments Passed** | MotAgErr\_Cnt\_T\_u16 | uint16 | 0 | 65535 |
|  | MotAgWarn\_Cnt\_T\_u16 | uint16 | 0 | 65535 |
|  | TurnCntrParFltCnt\_Cnt\_T\_u16 | uint16 | 0 | 65535 |
|  | MotAgVltgFltCnt\_Cnt\_T\_u16 | uint16 | 0 | 65535 |
|  | MotAgParFltCnt\_Cnt\_T\_u16 | uint16 | 0 | 65535 |
|  | MotAgSyncErr\_Cnt\_T\_u16 | uint8 | 0 | 1 |
|  | MotAg3Mecl\_MotRev\_T\_u0p16 | u0p16 | 0 | 65535 |
|  | MotAgSpiMecl\_MotRev\_T\_u0p16 | u0p16 | 0 | 65535 |
|  | TurnCntr\_Cnt\_T\_s16 | sint16 | -2048 | 2048 |
| **Return Value** | MotAgQlfr\_Cnt\_T\_enum | SigQlfr1 | 0 | 2 |
|  | MotAgTurnCntrQlfr\_Cnt\_T\_enum | SigQlfr1 | 0 | 2 |
|  | MotAgQepFaild\_Cnt\_T\_logl | boolean | FALSE | TRUE |

#### Design Rationale

Implementation of "MotAg Fault Processing" Simulink block

### CalcNtcPrm

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Function Name** | CalcNtcPrm | Type | Min | Max |
| **Arguments Passed** | MotAgErr\_Cnt\_T\_u16 | uint16 | 0 | 65535 |
|  | MotAgWarn\_Cnt\_T\_u16 | uint16 | 0 | 65535 |
|  | TurnCntrParFltCnt\_Cnt\_T\_u16 | uint16 | 0 | 65535 |
|  | MotAgVltgFltCnt\_Cnt\_T\_u16 | uint16 | 0 | 65535 |
|  | MotAgParFltCnt\_Cnt\_T\_u16 | uint16 | 0 | 65535 |
| **Return Value** | MotAgNtcParm\_Cnt\_T\_u08 | uint8 | 0 | 127 |
|  | TurnCtrNtcParm\_Cnt\_T\_u08 | uint8 | 0 | 255 |
|  | TurnCtrVltgNtcParm\_Cnt\_T\_u08 | uint8 | 0 | 1 |

#### Design Rationale

See “Determine NTC Parameters” block in the Simulink model of the design

### SetMotAg0FltNtc

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Function Name** | SetMotAg0FltNtc | Type | Min | Max |
| **Arguments Passed** | MotAgNtcParm\_Cnt\_T\_u08 | uint8 | 0 | 127 |
|  | TurnCntrNtcParm\_Cnt\_T\_u08 | uint8 | 0 | 255 |
|  | TurnCntrVltgNtcParm\_Cnt\_T\_u08 | uint8 | 0 | 1 |
|  | MotAgQepMecl\_MotRev\_T\_u0p16 | u0p16 | 0 | 65535 |
|  | MotAgSPIMecl\_\_MotRev\_T\_u0p16 | u0p16 | 0 | 65535 |
|  | TurnCntr\_Cnt\_T\_s16 | sint16 | -2048 | 2048 |
| **Return Value** | N/A | N/A | N/A | N/A |

#### Design Rationale

See “MotAg0PtrclFlt Processing, MotAg0TurnCntrFlt Processing & MotAg0TurnCntrVltgFlt Processing” block in the Simulink model of the design



### 

### CalculateMotAgTurnCntr

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Function Name** | CalculateMotAgTurnCntr | Type | Min | Max |
| **Arguments Passed** | MotAgSpiMecl\_MotRev\_T\_u0p16 | u0p16 | 0 | 65535 |
|  | TurnCntr\_Cnt\_T\_s16 | sint16 | -2048 | 2048 |
|  | MotAgPolarity\_Cnt\_T\_s08 | sint8 | -1 | 1 |
| **Return Value** | MotAgTurnCntr\_Cnt\_T\_f32 | float32 | -256.09375 | 255.96875 |

#### Design Rationale

Implementation of "Calculate MotAgTurnCntr" Simulink block

### SPI\_AngleRawProcess

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Function Name** | SPI\_AngleRawProcess | Type | Min | Max |
| **Arguments Passed** | RawAngReg\_Cnt\_T\_u32 | uint32 | 0 | 67108863 |
| **Return Value** | N/A | N/A | N/A | N/A |

#### Design Rationale

Implementation of "SPI\_AngleRawProcess" Simulink block

### CompensateMechMtrPos

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Function Name** | CompensateMechMtrPos | Type | Min | Max |
| **Arguments Passed** | MotAgRawMecl\_Cnt\_T\_u0p16 | u0p16 | 0 | 65535 |
| **Return Value** | MotAgMecl\_MotRev\_T\_u0p16 | u0p16 | 0 | 65535 |

#### Design Rationale

Implementation of "Compensate MechMtrPos" Simulink block

## GLOBAL Function/Macro Definitions

* MotAg0MeasPer1
* MotAg0CfgLoPwrMod

# Known Limitations with Design

None

# UNIT TEST CONSIDERATION

Following variables are used as rolling counters, so the overflow of these variables is intentional.

Rte\_Pim\_MotAg0VltgFltCntPrev

Rte\_Pim\_MotAg0ParFltCntPrev

Rte\_Pim\_MotAg0MeclRollgCntrPrev

Rte\_Pim\_MotAg0TurnCntrParFltCntPrev

There is an unreachable code in the SinCos\_f32() function, because the input of this function is always positive. This is ok as the SinCos\_f32() is a library function.

For the config param MOTAG0MEAS\_DIQEPIF\_CNT\_LOGL, please use the file CDD\_MotAg0Meas\_Cfg.h in the include folder.

Abbreviations and Acronyms

| **Abbreviation or Acronym** | **Description** |
| --- | --- |
| FDD | Functional Design Document. (See references) |

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](https://www.autosar.org/fileadmin/files/standards/classic/4-0/software-architecture/implementation-integration/standard/AUTOSAR_SWS_MemoryMapping.pdf)) | v1.4.0 R4.0 Rev 3 |
| 2 | MDD Guideline EA4 | 01.00.01 |
| 3 | EA4 Software Naming Conventions | 01.01.00 |
| 4 | Software Design and Coding Standards | 2.1 |
| 5 | CM620B\_MotAg0Meas\_Design | See Synergy Sub Project Version |