# Module –

# High-Level Description

This module implements the learning and diagnostic portions of MtrPos, as well as part of the motor position determination sub function.

# Figures

## Component Diagram

mtrpos2.emf

# Variable Data Dictionary

For details on module input / output variable, refer to the Data Dictionary for the application. Input / output variable names are listed here for reference.

|  |  |  |
| --- | --- | --- |
| Module Inputs | Module Outputs | |
| MotorVelMRF\_MtrRadpS\_f32 | | CorrectedMtrPos\_Rev\_f32 |
|  | | CosTheta1\_Volt\_f32 |
|  | | MechMtrPos\_Rev\_f32 |
|  | | SinTheta1\_Volt\_f32 |
|  | | SysCCorrectedMtrPos\_Rev\_f32 |
|  | | SysCMechMtrPos\_Rev\_f32 |
|  | |  |
|  | |  |

## Module Internal Variables

This section identifies the name, range and resolutions for module specific data created by this module. If there are no range restrictions on the variable, the term “FULL” is placed into the table for legal range.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Variable Name | Resolution | Legal Range  (min) | Legal Range  (max) | Software Segment |
| MtrPos2\_ValidityFltAcc\_Cnt\_M\_u16 | 1 | 0 | 65535 | MTRPOS2\_START\_SEC\_VAR\_CLEARED\_16 |
| MtrPos2\_CorrFltAcc\_Cnt\_M\_u16 | 1 | 0 | 65535 | MTRPOS2\_START\_SEC\_VAR\_CLEARED\_16 |
| MtrPos2\_PrevCorrectedMtrPos\_Rev\_M\_f32 | Single Precision Float | 0 | 0.99998 | MTRPOS2\_START\_SEC\_VAR\_CLEARED\_32 |
|  |  |  |  |  |
|  |  |  |  |  |
| MtrPos2\_PrevSin1RTOffset\_Volts\_M\_f32 | Single Precision Float | 1.7 | 3.8 | MTRPOS2\_START\_SEC\_VAR\_NOINIT\_32 |
| MtrPos2\_PrevCos1RTOffset\_Volts\_M\_f32 | Single Precision Float | 1.7 | 3.8 | MTRPOS2\_START\_SEC\_VAR\_NOINIT\_32 |
| MtrPos2\_Sin1OffsetCorr\_Volts\_M\_f32 | Single Precision Float | -0.5 | 0.5 | MTRPOS2\_START\_SEC\_VAR\_NOINIT\_32 |
| MtrPos2\_Cos1OffsetCorr\_Volts\_M\_f32 | Single Precision Float | -0.5 | 0.5 | MTRPOS2\_START\_SEC\_VAR\_NOINIT\_32 |
| MtrPos2\_Sin1GainCorr\_Uls\_M\_f32 | Single Precision Float | 0.8 | 1.2 | MTRPOS2\_START\_SEC\_VAR\_NOINIT\_32 |
| MtrPos2\_Cos1GainCorr\_Uls\_M\_f32 | Single Precision Float | 0.8 | 1.2 | MTRPOS2\_START\_SEC\_VAR\_NOINIT\_32 |
| MtrPos2\_CosSin1NomRatio\_Uls\_M\_f32 | Single Precision Float | 1 | 10 | MTRPOS2\_START\_SEC\_VAR\_NOINIT\_32 |
| MtrPos2\_SinCos1NomRatio\_Uls\_M\_f32 | Single Precision Float | 1 | 10 | MTRPOS2\_START\_SEC\_VAR\_NOINIT\_32 |
| MtrPos2\_RTToNomHighLmt\_Uls\_M\_f32 | Single Precision Float | 1 | 1.02 | MTRPOS2\_START\_SEC\_VAR\_NOINIT\_32 |
| MtrPos2\_RTToNomLowLmt\_Uls\_M\_f32 | Single Precision Float | 0.98 | 1 | MTRPOS2\_START\_SEC\_VAR\_NOINIT\_32 |
| MtrPos2\_Cos1RTAmpRec\_Uls\_M\_f32 | Single Precision Float | 0.2 | 3.3 | MTRPOS2\_START\_SEC\_VAR\_NOINIT\_32 |
| MtrPos2\_Cos1RTOffset\_Volt\_M\_f32 | Single Precision Float | 1.7 | 3.8 | MTRPOS2\_START\_SEC\_VAR\_NOINIT\_32 |
| MtrPos2\_Sin1RTAmpRec\_Uls\_M\_f32 | Single Precision Float | 0.2 | 3.3 | MTRPOS2\_START\_SEC\_VAR\_NOINIT\_32 |
| MtrPos2\_Sin1RTOffset\_Volt\_M\_f32 | Single Precision Float | 1.7 | 3.8 | MTRPOS2\_START\_SEC\_VAR\_NOINIT\_32 |
| MtrPos2\_Cos1MaxSV\_Volts\_M\_s2p29 | 2-29 | 0.25 | 2.5 | MTRPOS2\_START\_SEC\_VAR\_SAVED\_ZONEH\_32 |
| MtrPos2\_Cos1MinSV\_Volts\_M\_s2p29 | 2-29 | -0.25 | -2.5 | MTRPOS2\_START\_SEC\_VAR\_SAVED\_ZONEH\_32 |
| MtrPos2\_Sin1MaxSV\_Volts\_M\_s2p29 | 2-29 | 0.25 | 2.5 | MTRPOS2\_START\_SEC\_VAR\_SAVED\_ZONEH\_32 |
| MtrPos2\_Sin1MinSV\_Volts\_M\_s2p29 | 2-29 | -0.25 | -2.5 | MTRPOS2\_START\_SEC\_VAR\_SAVED\_ZONEH\_32 |
|  |  |  |  |  |
| MtrPos2\_ValidErr\_VoltsSqrd\_D\_f32 | Single Precision Float | 0 | 4.5 | MTRPOS2\_START\_SEC\_VAR\_CLEARED\_32 |
|  |  |  |  |  |
| MtrPos2\_LearnedParamBfr\_Cnt\_M\_u08 | 1 | 0 | 1 | MTRPOS2\_START\_SEC\_VAR\_CLEARED\_8 |
| MtrPos2\_Sin1RTOffset\_Volts\_M\_u3p13[D\_MTRPOSDBLBUFFSZ\_CNT\_U08] | 2^-13 | 1.7 | 3.8 | MTRPOS2\_START\_SEC\_VAR\_CLEARED\_16 |
| MtrPos2\_Cos1RTOffset\_Volts\_M\_u3p13[D\_MTRPOSDBLBUFFSZ\_CNT\_U08] | 2^-13 | 1.7 | 3.8 | MTRPOS2\_START\_SEC\_VAR\_CLEARED\_16 |
| MtrPos2\_Sin1RTAmpRec\_Uls\_M\_u3p13[D\_MTRPOSDBLBUFFSZ\_CNT\_U08] | 2^-13 | 0.2 | 3.3 | MTRPOS2\_START\_SEC\_VAR\_CLEARED\_16 |
| MtrPos\_Cos1RTAmpRec\_Uls\_M\_u3p13[D\_MTRPOSDBLBUFFSZ\_CNT\_U08] | 2^-13 | 0.2 | 3.3 | MTRPOS2\_START\_SEC\_VAR\_CLEARED\_16 |
| MtrPos2\_ErrorTerm\_Rev\_D\_f32 | Single Precision Float | 0 | 0.5 | MTRPOS2\_START\_SEC\_VAR\_CLEARED\_32 |

### 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) |
| MtrPosCal\_DataType | BEMFCal\_Rev\_u0p16  R\_BEMFCal\_Rev\_u0p16  Sin1Offset\_Volts\_u3p13  Sin1AmpRec\_Uls\_u3p13  Cos1Offset\_Volts\_u3p13  Cos1AmpRec\_Uls\_u3p13  SinDelta1\_Uls\_s2p13  CosDelta1Rec\_Uls\_u3p13  Sin1OffCorr\_Volts\_s2p13  Sin1GainCorr\_Uls\_u1p15  Cos1OffCorr\_Volts\_s2p13  Cos1GainCorr\_Uls\_u1p15  SinHarTbl\_Cnt\_sm6p13[144]  CosHarTbl\_Cnt\_sm6p13[144] |  | 0  0  2.2  0.25  2.2  0.25  -0.0174524  0.99985  -0.5  0.8  -0.5  0.8  -0.01551  -0.01551 | 1  1  2.8  2.5  2.8  2.5  0.0174524  1  0.5  1.2  0.5  1.2  0.01551  0.01551 |

\*In FDD 06B, Ver 6, range for SinHarTbl and CosHarTbl is given as -1 to 1 which is incorrect as it’s a 8bit signal. FDD needs to be updated. Informed to FDD Owner.

# Constant Data Dictionary

## Calibration Constants

This section lists the calibrations used by the module. For details on calibration constants, refer to the Data Dictionary for the application.

|  |
| --- |
| Constant Name |
| k\_RTOffVelThr\_MtrRadpS\_f32 |
| k\_RTFiltEnThresh\_Uls\_f32 |
| k\_RTOffFiltKn\_Cnt\_u16 |
| k\_RTOffsetLmt\_Volts\_f32 |
| k\_AmpRecVarLmt\_Uls\_f32 |
| k\_RTToNomRatioVar\_Uls\_f32 |
| k\_CorrelationError\_Rev\_f32 |
| k\_MtrPosCorrDiag\_Cnt\_str |
| k\_ValMinError\_VoltsSqrd\_f32 |
| k\_ValMaxError\_VoltsSqrd\_f32 |
| k\_MtrPosValDiag\_Cnt\_str |

## Program(fixed) Constants

### Embedded Constants

All embedded constants whose values are provided in Eng units will be evaluated to the equivalent counts by using the FPM\_InitFixedPoint\_m() macro within the #define statement.

#### Local

|  |  |  |  |
| --- | --- | --- | --- |
| Constant Name | Resolution | Units | Value |
| D\_DEGPERREV\_ULS\_F32 | Single Precision Float | Unitless | 360 |
| D\_OFFSETCORRTHRESH\_VOLTS\_F32 | Single Precision Float | Volts | 0.02 |
| D\_AMPCORRTHRESH\_ULS\_F32 | Single Precision Float | Unitless | 0.02 |
| D\_MAXAMPLITUDE\_VOLTS\_F32 | Single Precision Float | Volts | 2.5 |
| D\_MINAMPLITUDE\_VOLTS\_F32 | Single Precision Float | Volts | 0.25 |
| D\_HALF\_ULS\_F32 | Single Precision Float | Unitless | 0.5 |
| D\_TWO\_ULS\_F32 | Single Precision Float | Unitless | 2 |
| D\_WORDMASK\_CNT\_U16 | 1 | Counts | 0xFFFF |

#### Global

This section lists the global constants used by the module. For details on global constants, refer to the Data Dictionary for the application.

|  |
| --- |
| Constant Name |
| D\_ZERO\_ULS\_F32 |
| D\_ONE\_ULS\_F32 |
| D\_2PI\_ULS\_F32 |
| D\_ZERO\_CNT\_U16 |

### Module specific Lookup Tables Constants

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

# Functions/Macros used by the Sub-Modules

## Library Functions / Macros

The library and functions / Macros that are called by the various sub modules are identified below,

1. FPM\_FixedToFloat\_m
2. FPM\_FloatToFixed\_m
3. Abs\_f32\_m
4. sinf
5. cosf
6. DiagPStep\_m
7. DiagNStep\_m
8. DiagFailed\_m
9. LPF\_SvUpdate\_s16InFixKTrunc\_m
10. Limit\_m

## Data Hiding Functions

1. Rte\_Pim\_MtrPosSnsr\_EOLData
2. Rte\_Call\_NxtrDiagMgr\_GetNTCFailed

## Global Functions/Macros Defined by this Module

None

## Local Functions/Macros Used by this MDD only

### Offset Initialization

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Function Name** | OffsetInit | Type | Min | Max | UTP Tol. |
| **Arguments Passed** | EOLAmpRec\_Uls\_T\_f32 | float32 | 0.25 | 2.5 |  |
|  | EOLOffset\_Volts\_T\_f32 | float32 | 2.2 | 2.8 |  |
|  | StateVarMaxPtr\_Volts\_T\_s2p29 | sint32 \* | 0.25 | 2.5 | 2-13 |
|  | StateVarMinPtr\_Volts\_T\_s2p29 | sint32 \* | -2.5 | -0.25 | 2-13 |
| **Return Value** | RTOffset\_Volts\_T\_f32 | float32 | 2.2 | 2.8 | 2-13 |

#### Note: In FDD, Range on StoredMax\_volts, Stored\_Mins are given as 0.25 to 2.5 and -2.5 to -0.25 respt. And in this local function these signals are compared to D\_MinAmplitude, D\_MaxAmplitude whose const value is 0.25 or 2.5v. 100%coverage for the below IF statement is not achieved with provided ranges. FDD should correct range and is informed to FDD owner. For UnitTesting purpose, special vectors are taken to satisfy IF condition

#### Description



### Run Time Filter

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Function Name** | RunTimeFilter | Type | Min | Max | UTP Tol. |
| **Arguments Passed** | SigEst\_Uls\_T\_f32 | float32 | -1 | 1 |  |
|  | SigCorr\_Volts\_T\_f32 | float32 | -3.8 | 3.3 |  |
|  | OffsetDelta\_Volts\_T\_f32 | float32 | -1.1 | 1.6 |  |
|  | StateVarPtr\_Volts\_T\_s2p29 | s2p29\_T | -2.5 | 2.5 |  |
| **Return Value** | Output\_Volts\_T\_f32 | float32 | -2.5 | 2.5 | 2-13 |

#### Description



### Calculate Offset from Min/Max

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Function Name** | CalcOffsetFromMinMax | Type | Min | Max | UTP Tol. |
| **Arguments Passed** | Min\_Volts\_T\_f32 | float32 | -0.25 | -2.5 |  |
|  | Max\_Volts\_T\_f32 | float32 | 0.25 | 2.5 |  |
|  | EOLOffset\_Volts\_T\_f32 | float32 | 2.2 | 2.8 |  |
| **Return Value** | RTOffset\_Volts\_T\_f32 | float32 | 1.7 | 3.8 | 2-13 |

#### Description



### Calculate Sine AmpRec

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Function Name** | CalcSinAmpRec | Type | Min | Max | UTP Tol. |
| **Arguments Passed** | TotSinAmp\_Volts\_T\_f32 | float32 | 0.5 | 5 |  |
|  | EOLSinAmpRec\_Uls\_T\_f32 | float32 | 0.25 | 2.5 |  |
| **Return Value** | SinRTAmpRec\_Uls\_T\_f32 | float32 | 2.2 | 3.3 | 2-13 |

#### Description



### Calculate Cosine AmpRec

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Function Name** | CalcCosAmpRec | Type | Min | Max | UTP Tol. |
| **Arguments Passed** | TotSinAmp\_Volts\_T\_f32 | float32 | 0.5 | 5 |  |
|  | TotCosAmp\_Volts\_T\_f32 | float32 | 0.5 | 5 |  |
|  | SinRTAmpRec\_Uls\_T\_f32 | float32 | 0.2 | 3.3 |  |
|  | SinCosNomRatio\_Uls\_T\_f32 | float32 | 1 | 10 |  |
|  | CosSinNomRatio\_Uls\_T\_f32 | float32 | 1 | 10 |  |
|  | RTToNomLowLmt\_Uls\_T\_f32 | float32 | 0.98 | 1 |  |
|  | RTToNomHighLmt\_Uls\_T\_f32 | float32 | 1 | 1.02 |  |
| **Return Value** | CosRTAmpRec\_Uls\_T\_f32 | float32 | 1.96. | 33.66 | 2-13 |

#### Description



# Software Module Implementation

## Runtime Environment (RTE) Initial Values

This section lists the initial values of data written by this module but controlled by the RTE. After RTE initialization, the data in this table will contain these values.

|  |  |
| --- | --- |
| Data | Value |
| Rte\_InitValue\_CorrectedMtrPos\_Rev\_f32 | 0 |
| Rte\_InitValue\_Cos1RTAmpRec\_Uls\_f32 | 0 |
| Rte\_InitValue\_Cos1RTOffset\_Volt\_f32 | 0 |
| Rte\_InitValue\_DiagCorrectedMtrPos\_Rev\_f32 | 0 |
| Rte\_InitValue\_MechMtrPos\_Rev\_f32 | 0 |
| Rte\_InitValue\_MotorVelMRF\_MtrRadpS\_f32 | 0 |
| Rte\_InitValue\_Sin1RTOffset\_Volt\_f32 | 0 |

## Initialization Functions

### Init: \_Init1

#### Design Rationale

None

#### Calculate Ratios



#### Check Correction Terms



#### Calculate Limits and Initialize RT Offsets



#### Initialize AmpRec



## Periodic Functions

### Per: \_Per1

#### Design Rationale

This function should be run before MtrPos3\_Per1, as the comparisons (in both the main and diverse paths) on the generated signals assume that this order will be followed.

#### Program Flow Start

Rte\_Call\_MtrPos2\_Per1\_CP0\_CheckpointReached()

#### Store Module Inputs to Local copies

CDD\_Read\_MtrPos\_ActvDblBuf\_Cnt\_u08(&IdleDblBuf\_Cnt\_T\_u08)

IdleDblBuf\_Cnt\_T\_u08 = (IdleDblBuf\_Cnt\_T\_u08 & 1u)^1u

CorrectedMtrPos\_Rev\_T\_f32 = FPM\_FixedToFloat\_m(MtrPos\_CorrectedMtrPos\_Rev\_M\_u0p16[IdleDblBuf\_Cnt\_T\_u08], u0p16\_T)

Cos1Scaled\_Volt\_T\_f32 = FPM\_FixedToFloat\_m(MtrPos\_Cos1Scaled\_Volts\_M\_u3p13[IdleDblBuf\_Cnt\_T\_u08], u3p13\_T)

DiagCorrectedMtrPos\_Rev\_T\_f32 = MtrPos3\_DiagCorrectedMtrPos\_Rev\_M\_f32

MechMtrPos\_Rev\_T\_f32 = \_FixedToFloat\_m(MtrPos\_MechMtrPos\_Rev\_M\_u0p16[IdleDblBuf\_Cnt\_T\_u08], u0p16\_T)

MotorVelMRF\_MtrRadpS\_T\_f32 = Rte\_IRead\_MtrPos2\_Per1\_MotorVelMRF\_MtrRadpS\_f32()

Sin1Scaled\_Volt\_T\_f32 = FPM\_FixedToFloat\_m(MtrPos\_Sin1Scaled\_Volts\_M\_u3p13[IdleDblBuf\_Cnt\_T\_u08], u3p13\_T)

Rte\_Call\_NxtrDiagMgr\_GetNTCFailed(NTC\_Num\_PriMSB\_SinCosCorr, &MtrPosFault1\_Cnt\_T\_lgc)

Rte\_Call\_NxtrDiagMgr\_GetNTCFailed(NTC\_Num\_PriVsSec\_SinCosCorr, &MtrPosFault2\_Cnt\_T\_lgc)

Sin1Offset\_Volts\_T\_f32 = FPM\_FixedToFloat\_m(MtrPos\_EOLDataPtr\_Cnt\_M\_Str->Sin1Offset\_Volts\_u3p13, u3p13\_T)

Cos1Offset\_Volts\_T\_f32 = FPM\_FixedToFloat\_m(MtrPos\_EOLDataPtr\_Cnt\_M\_Str->Cos1Offset\_Volts\_u3p13, u3p13\_T)

Sin1AmpRec\_Uls\_T\_f32 = FPM\_FixedToFloat\_m(MtrPos\_EOLDataPtr\_Cnt\_M\_Str->Sin1AmpRec\_Uls\_u3p13, u3p13\_T)

#### Motor Position Measurement

#### Run Time Learning



#### Back EMF Diagnostic



#### Correlation Diagnostic



#### Validity Diagnostic



#### Store Local copy of outputs into Module Outputs

MtrPos2\_PrevCorrectedMtrPos\_Rev\_M\_f32 = CorrectedMtrPos\_Rev\_T\_f32

MtrPos2\_ErrorTerm\_Rev\_D\_f32 = CorrelationDelta\_Rev\_T\_f32

MtrPosValidErr\_VoltsSqrd\_D\_f32 = ValidErr\_VoltsSqrd\_T\_f32

Rte\_IWrite\_MtrPos2\_Per1\_CorrectedMtrPos\_Rev\_f32(CorrectedMtrPos\_Rev\_T\_f32)

Rte\_IWrite\_MtrPos2\_Per1\_SysCCorrectedMtrPos\_Rev\_f32(CorrectedMtrPos\_Rev\_T\_f32)

Rte\_IWrite\_MtrPos2\_Per1\_MechMtrPos\_Rev\_f32(MechMtrPos\_Rev\_T\_f32)

Rte\_IWrite\_MtrPos2\_Per1\_SysCMechMtrPos\_Rev\_f32(MechMtrPos\_Rev\_T\_f32)

Rte\_IWrite\_MtrPos2\_Per1\_CosTheta1\_Volt\_f32(FPM\_FixedToFloat\_m(MtrPos\_CosTheta1\_Volts\_G\_s2p13, s2p13\_T))

Rte\_IWrite\_MtrPos2\_Per1\_SinTheta1\_Volt\_f32(FPM\_FixedToFloat\_m(MtrPos\_SinTheta1\_Volts\_G\_s2p13, s2p13\_T))

IdleDblBuf\_Cnt\_T\_u08 = (MtrPos2\_LearnedParamBfr\_Cnt\_M\_u08 & 1)^1

MtrPos2\_Cos1RTAmpRec\_Uls\_M\_u3p13[IdleDblBuf\_Cnt\_T\_u08] = FPM\_FloatToFixed\_m(MtrPos2\_Cos1RTAmpRec\_Uls\_M\_f32, u3p13\_T)

MtrPos2\_Cos1RTOffset\_Volts\_M\_u3p13[IdleDblBuf\_Cnt\_T\_u08] = FPM\_FloatToFixed\_m(MtrPos2\_Cos1RTOffset\_Volt\_M\_f32, u3p13\_T)

MtrPos2\_Sin1RTAmpRec\_Uls\_M\_u3p13[IdleDblBuf\_Cnt\_T\_u08] = FPM\_FloatToFixed\_m(MtrPos2\_Sin1RTAmpRec\_Uls\_M\_f32, u3p13\_T)

MtrPos2\_Sin1RTOffset\_Volts\_M\_u3p13[IdleDblBuf\_Cnt\_T\_u08] = FPM\_FloatToFixed\_m(MtrPos2\_Sin1RTOffset\_Volt\_M\_f32, u3p13\_T)

MtrPos2\_LearnedParamBfr\_Cnt\_M\_u08 = IdleDblBuf\_Cnt\_T\_u08

#### Program Flow End

Rte\_Call\_MtrPos2\_Per1\_CP1\_CheckpointReached()

## Fault Recovery Functions

None

## Shutdown Functions

None

## Interrupt Functions

None

## Serial Communication Functions

### SCom: \_SCom\_ReadEOLMtrCals

|  |  |
| --- | --- |
| **Arguments Passed** | Type |
| MtrCalDataPtr | MtrPosCal\_DataType \* |

#### Design Rationale

None

#### Program Flow Start

N/A

#### Store Module Inputs to Local copies

None

#### Processing of function



#### Store Local copy of outputs into Module Outputs

None

#### Program Flow End

N/A

### SCom: \_SCom\_SetEOLMtrCals

|  |  |
| --- | --- |
| **Arguments Passed** | Type |
| MtrCalDataPtr | MtrPosCal\_DataType \* |

#### Design Rationale

None

#### Program Flow Start

N/A

#### Store Module Inputs to Local copies

None

#### Processing of function



#### Store Local copy of outputs into Module Outputs

Rte\_IWrite\_MtrPos2\_SCom\_SetEOLMtrCals\_EOLBEMF\_Rev\_f32(FPM\_FixedToFloat\_m(Rte\_Pim\_MtrPosSnsr\_EOLData()->BEMFCal\_Rev\_u0p16, u0p16\_T))

#### Program Flow End

N/A

# Execution Requirements

## Execution Sequence of the Module

See section 6.3.1.1 for details on task ordering.

## Execution Rates for sub-modules called by the Scheduler

This table serves as reference for the Scheduler design

|  |  |  |
| --- | --- | --- |
| Function Name | Calling Frequency | System State(s) in which the function is called |
| MtrPos2\_Init1 | On Event | On Init |
| MtrPos2\_Per1 | 2 ms | ALL |

## Execution Requirements for Serial Communication Functions

|  |  |
| --- | --- |
| Function Name | Sub-Module called by (Serial Comm Function Name) |
| MtrPos2\_SCom\_ReadEOLMtrCals |  |
| MtrPos2\_SCom\_SetEOLMtrCals |  |

# Memory Map Definition Requirements

## Sub Modules (Functions)

This table identifies the software segments for functions identified in this module.

|  |  |
| --- | --- |
| Name of Sub Module | Software Segment |
| MtrPos2\_Init1 | RTE\_START\_SEC\_SA\_MTRPOS2\_APPL\_CODE |
| MtrPos2\_Per1 | RTE\_START\_SEC\_SA\_MTRPOS2\_APPL\_CODE |

## Local Functions

This table identifies the software segments for local functions identified in this module.

|  |  |
| --- | --- |
| Name of Sub Module | Software Segment |
| OffsetInit |  |
| RunTimeFilter |  |
| CalcOffsetFromMinMax |  |
| CalcSinAmpRec |  |
| CalcCosAmpRec |  |

# Known Issues / Limitations With Design

1. INLINE functions defined in GlobalMacro.h are not unit tested.

# Revision Control Log

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Item #** | **Rev #** | **Change Description** | **Date** | **Author Initials** |
| 1 | 1.0 | Initial Version | 23-Oct-12 | OT |
| 2 | 2.0 | UTP updates | 18-Dec-12 | OT |
| 3 | 3.0 | MDD Catchup for SRC Ver 4 | 13-June-13 | NRAR |