# Module -- CmMtrCurr

# High-Level Description

The Current Measurement function is responsible for measuring the motor phase currents used as feedback by the Motor Control FDD. Two motor phase currents are measured using a shunt resistor and a differential amplifier circuitry, and along with the motor position are transformed into direct (D) and quadrature (Q) axes currents using the combined Clarke/Park transform

UNIT Test Notes:

Unit test should be done with enabling one of the six predefned macro MTRCURRPHASEBC, MTRCURRPHASECB, MTRCRRPHASECA, MTRCURRPHASEAB, MTRCURRPHASEAC, MTRCURRPHASEBA. Hence it should have six UTP results (one for each Macro enabled).

# Figures

## Component diagram

# 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 | |
| ADCMtrCurr1\_Volt\_f32 | | MtrCurrQax\_Amps\_f32 |
| ADCMtrCurr2\_Volt\_f32 | | MtrCurrDax\_Amps\_f32 |
| MtrVel\_MtrRadpS\_f32 | | CurrentGainSvc\_Cnt\_lgc |
| FiltCntrlTemp\_DegC\_f32 | | ComOffset\_Cnt\_u16 |
| MtrCurrAngle\_Rev\_f32 | | ElecPosDelayComp\_Rad\_f32 |
| VehSpd\_Kph\_f32 | | CorrMtrCurrPosition\_Rev\_f32 |
| VhSpdValid\_Cnt\_lgc | | MtrCurrK1\_Amps\_f32 |
| Vecu\_Volt\_f32 | | MtrCurrK2\_Amps\_f32 |
| MtrCurr1TempOffset\_Volt\_f32 | | MtrCurr1\_Volts\_f32 |
| MtrCurr2TempOffset\_Volt\_f32 | | MtrCurr2\_Volts\_f32 |
| Phs1Curr\_Cnt\_u16 | | MtrCurrQax\_Amps\_f32 |
| Phs2Curr\_Cnt\_u16 | | MtrCurrDax\_Amps\_f32 |
| MtrElecPol\_Cnt\_s08 | | CurrentGainSvc\_Cnt\_lgc |
| DCPhsBComp\_Cnt\_u16 | |  |
| DCPhsCComp\_Cnt\_u16 | |  |
| DCPhsCComp\_Cnt\_u16 | |  |
| DCPhsBComp\_Cnt\_u16 | |  |
| DCPhsAComp\_Cnt\_u16 | |  |
| DCPhsBComp\_Cnt\_u16 | |  |
| DCPhsBComp\_Cnt\_u16 | |  |
| DCPhsAComp\_Cnt\_u16 | |  |
| DCPhsAComp\_Cnt\_u16 | |  |
| DCPhsCComp\_Cnt\_u16 | |  |
| DCPhsCComp\_Cnt\_u16 | |  |
| DCPhsAComp\_Cnt\_u16 | |  |
| ADC2OffsetComp\_Cnt\_u8p8 | |  |

## 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 |
| CmMtrCurr\_CorrMtrCurr1\_Amp\_D\_f32 | See Data Dictionary | See Data Dictionary | See Data Dictionary | CMMTRCURR\_START\_SEC\_VAR\_CLEARED\_32 |
| CmMtrCurr\_CorrMtrCurr2\_Amp\_D\_f32 | See Data Dictionary | See Data Dictionary | See Data Dictionary | CMMTRCURR\_START\_SEC\_VAR\_CLEARED\_32 |
| CmMtrCurr\_CurrVectPosition\_Rev\_D\_f32 | See Data Dictionary | See Data Dictionary | See Data Dictionary | CMMTRCURR\_START\_SEC\_VAR\_CLEARED\_32 |
| CmMtrCurr\_VectPosCosTheta\_Uls\_D\_f32 | See Data Dictionary | See Data Dictionary | See Data Dictionary | CMMTRCURR\_START\_SEC\_VAR\_CLEARED\_32 |
| CmMtrCurr\_VectPosSinTheta\_Uls\_D\_f32 | See Data Dictionary | See Data Dictionary | See Data Dictionary | CMMTRCURR\_START\_SEC\_VAR\_CLEARED\_32 |
| CmMtrCurr\_CurrCorrDiag\_Amp\_D\_f32 | See Data Dictionary | See Data Dictionary | See Data Dictionary | CMMTRCURR\_START\_SEC\_VAR\_CLEARED\_32 |
| CmMtrCurr\_FiltCurrCorrDiag\_Amp\_D\_f32 | See Data Dictionary | See Data Dictionary | See Data Dictionary | CMMTRCURR\_START\_SEC\_VAR\_CLEARED\_32 |
| CmMtrCurr\_ CurrentGainSvc\_Cnt\_M\_lgc | See Data Dictionary | See Data Dictionary | See Data Dictionary | CMMTRCURR\_START\_SEC\_VAR\_CLEARED\_BOOLEAN |
| CmMtrCurr\_CurrCorrDiagKSV\_M\_str | See Data Dictionary | See Data Dictionary | See Data Dictionary | CMMTRCURR\_START\_SEC\_VAR\_CLEARED\_UNSPECIFIED |
| CurrCorrDiagKSV\_M\_str.SV\_Uls\_f32 | See Data Dictionary | See Data Dictionary | See Data Dictionary | CMMTRCURR\_START\_SEC\_VAR\_CLEARED\_UNSPECIFIED |
| CurrCorrDiagKSV\_M\_str.K\_Uls\_f32 | See Data Dictionary | See Data Dictionary | See Data Dictionary | CMMTRCURR\_START\_SEC\_VAR\_CLEARED\_UNSPECIFIED |
| CmMtrCurr\_MtrCurr1LpFltrSV\_Volt\_M\_u3p29 | See Data Dictionary | See Data Dictionary | See Data Dictionary | CMMTRCURR\_START\_SEC\_VAR\_CLEARED\_32 |
| CmMtrCurr\_MtrCurr2LpFltrSV\_Volt\_M\_u3p29 | See Data Dictionary | See Data Dictionary | See Data Dictionary | CMMTRCURR\_START\_SEC\_VAR\_CLEARED\_32 |
| CmMtrCurr\_FiltMtrCurr1\_Volt\_M\_f32 | See Data Dictionary | See Data Dictionary | See Data Dictionary | CMMTRCURR\_START\_SEC\_VAR\_CLEARED\_32 |
| CmMtrCurr\_FiltMtrCurr2\_Volt\_M\_f32 | See Data Dictionary | See Data Dictionary | See Data Dictionary | CMMTRCURR\_START\_SEC\_VAR\_CLEARED\_32 |
| CmMtrCurr\_MtrCurr1SumHi\_Volt\_M\_f32 | See Data Dictionary | See Data Dictionary | See Data Dictionary | CMMTRCURR\_START\_SEC\_VAR\_CLEARED\_32 |
| CmMtrCurr\_MtrCurr2SumHi\_Volt\_M\_f32 | See Data Dictionary | See Data Dictionary | See Data Dictionary | CMMTRCURR\_START\_SEC\_VAR\_CLEARED\_32 |
| CmMtrCurr\_MtrCurr1SumLo\_Volt\_M\_f32 | See Data Dictionary | See Data Dictionary | See Data Dictionary | CMMTRCURR\_START\_SEC\_VAR\_CLEARED\_32 |
| CmMtrCurr\_MtrCurr2SumLo\_Volt\_M\_f32 | See Data Dictionary | See Data Dictionary | See Data Dictionary | CMMTRCURR\_START\_SEC\_VAR\_CLEARED\_32 |
| CmMtrCurr\_MtrCurr1SumZero\_Volt\_M\_f32 | See Data Dictionary | See Data Dictionary | See Data Dictionary | CMMTRCURR\_START\_SEC\_VAR\_CLEARED\_32 |
| CmMtrCurr\_MtrCurr2SumZero\_Volt\_M\_f32 | See Data Dictionary | See Data Dictionary | See Data Dictionary | CMMTRCURR\_START\_SEC\_VAR\_CLEARED\_32 |
| CmMtrCurr\_VecuSum\_Volt\_M\_f32 | See Data Dictionary | See Data Dictionary | See Data Dictionary | CMMTRCURR\_START\_SEC\_VAR\_CLEARED\_32 |
| CmMtrCurr\_MtrCurr1OffsetHi\_Volt\_M\_f32 | See Data Dictionary | See Data Dictionary | See Data Dictionary | CMMTRCURR\_START\_SEC\_VAR\_CLEARED\_32 |
| CmMtrCurr\_MtrCurr2OffsetHi\_Volt\_M\_f32 | See Data Dictionary | See Data Dictionary | See Data Dictionary | CMMTRCURR\_START\_SEC\_VAR\_CLEARED\_32 |
| CmMtrCurr\_MtrCurr1OffsetLo\_Volt\_M\_f32 | See Data Dictionary | See Data Dictionary | See Data Dictionary | CMMTRCURR\_START\_SEC\_VAR\_CLEARED\_32 |
| CmMtrCurr\_MtrCurr2OffsetLo\_Volt\_M\_f32 | See Data Dictionary | See Data Dictionary | See Data Dictionary | CMMTRCURR\_START\_SEC\_VAR\_CLEARED\_32 |
| CmMtrCurr\_MtrCurr1OffsetZero\_Volt\_M\_f32 | See Data Dictionary | See Data Dictionary | See Data Dictionary | CMMTRCURR\_START\_SEC\_VAR\_CLEARED\_32 |
| CmMtrCurr\_MtrCurr2OffsetZero\_Volt\_M\_f32 | See Data Dictionary | See Data Dictionary | See Data Dictionary | CMMTRCURR\_START\_SEC\_VAR\_CLEARED\_32 |
| CmMtrCurr\_MtrCurrValCmd\_VoltCnt\_M\_f32 | See Data Dictionary | See Data Dictionary | See Data Dictionary | CMMTRCURR\_START\_SEC\_VAR\_CLEARED\_32 |
| CmMtrCurr\_MtrCurr1OffDelta\_VoltpVoltCnt\_M\_f32 | See Data Dictionary | See Data Dictionary | See Data Dictionary | CMMTRCURR\_START\_SEC\_VAR\_CLEARED\_32 |
| CmMtrCurr\_MtrCurr2OffDelta\_VoltpVoltCnt\_M\_f32 | See Data Dictionary | See Data Dictionary | See Data Dictionary | CMMTRCURR\_START\_SEC\_VAR\_CLEARED\_32 |
| CmMtrCurr\_CurrOffAvgCounter\_Cnt\_M\_u16 | See Data Dictionary | See Data Dictionary | See Data Dictionary | CMMTRCURR\_START\_SEC\_VAR\_CLEARED\_16 |
| CmMtrCurr\_CurrOffState\_Uls\_M\_enum | See Data Dictionary | See Data Dictionary | See Data Dictionary | CMMTRCURR\_START\_SEC\_VAR\_CLEARED\_UNSPECIFIED |
| CmMtrCurr\_CurroffProcessFlag\_M\_enum | See Data Dictionary | See Data Dictionary | See Data Dictionary | CMMTRCURR\_START\_SEC\_VAR\_CLEARED\_UNSPECIFIED |
| CmMtrCurr\_CurrOffTrimFlag\_Cnt\_M\_lgc | See Data Dictionary | See Data Dictionary | See Data Dictionary | CMMTRCURR\_START\_SEC\_VAR\_CLEARED\_BOOLEAN |
| CmMtrCurr\_MtrCurr1Offset\_Volt\_D\_f32 | See Data Dictionary | See Data Dictionary | See Data Dictionary | CMMTRCURR\_START\_SEC\_VAR\_CLEARED\_32 |
| CmMtrCurr\_MtrCurr2Offset\_Volt\_D\_f32 | See Data Dictionary | See Data Dictionary | See Data Dictionary | CMMTRCURR\_START\_SEC\_VAR\_CLEARED\_32 |
| CmMtrCurr\_CorrMtrCurr1\_Amp\_D\_f32 | See Data Dictionary | See Data Dictionary | See Data Dictionary | CMMTRCURR\_START\_SEC\_VAR\_CLEARED\_32 |
| CmMtrCurr\_CorrMtrCurr2\_Amp\_D\_f32 | See Data Dictionary | See Data Dictionary | See Data Dictionary | CMMTRCURR\_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) |
| CurrTempOffsetType | CurrTempOffsetX\_DegC\_s10p5 | CurrTempOffsetTblType | -50 | 150 |
|  | CurrOffsetY1\_Volts\_s4p11 | CurrTempOffsetTblType | -0.026 | 0.026 |
|  | CurrOffsetY2\_Volts\_s4p11 | CurrTempOffsetTblType | -0.026 | 0.026 |
| PhaseCurrCal\_DataType | EOLMtrCurrVcalCmd\_VoltCnts\_f32 | float | 0 | 80000 |
|  | EOLPhscurr1Gain\_AmpspVolt\_f32 | float | 20 | 125 |
|  | EOLMtrCurr2OffsetDiff\_Volts\_f32 | float | 1.0 | 3.0 |
|  | EOLMtrCurr1OffsetDiff\_Volts\_f32 | float | 1.0 | 3.0 |
|  | EOLMtrCurr1OffsetLo\_Volts\_f32 | float | 1.0 | 3.0 |
|  | EOLMtrCurr2OffsetLo\_Volts\_f32 | float | 1.0 | 3.0 |
|  | EOLPhscurr2Gain\_AmpspVolt\_f32 | float | 20 | 125 |
| CurrTempOffsetTblType[16] |  | Sint16 | Full | Full |

# 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\_CurrCorrErrFiltFc\_Hz\_f32 |
| k\_CurrCorrErrThresh\_Amps\_f32 |
| k\_MtrPosComputDelay\_Sec\_f32 |
| k\_MtrCurrEOLMinOffset\_Volts\_f32 |
| k\_MtrCurrEOLMaxOffset\_Volts\_f32 |
| k\_MtrCurrEOLMinGain\_AmpspVolts\_f32 |
| k\_MtrCurrEOLMaxGain\_AmpspVolts\_f32 |
| k\_CurrGainNumerator\_Amps\_f32 |
| k\_MaxCurrOffMtrVel\_RadpS\_f32 |
| k\_CurrOffGainKn\_Cnt\_u16 |
| k\_CurrCorrErrFiltFc\_Hz\_f32 |
| k\_CurrCorrErrThresh\_Amps\_f32 |
| k\_MtrPosComputDelay\_Sec\_f32 |
| k\_MtrCurrOffLoComOff\_Cnt\_u16 |
| k\_CurrOffNoofAvg\_Cnt\_u16 |

## 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\_ADCREF\_VOLT\_F32 | Single Precision float | Volt | 5.0F |
| D\_ADCFULLSCALE\_CNT\_U16 | 1 | Cnt | 4095U |
| D\_SCALERADTOCNTS\_ULS\_F32 | Single Precision float | Uls | 10430.3783505F |
| D\_REVWITHROUND\_ULS\_F32 | Single Precision float | Uls | 65536.5F |
| D\_ROUND\_ULS\_F32 | Single Precision float | Uls | 0.5F |
| D\_ADCCOMPOFFSAMPLESIZE\_CNT\_U16 | 1 | Cnt | 256u |
| D\_SCALE\_VOLTSPERCOUNT\_F32 | Single Precision float | VoltspSec | (D\_ADCREF\_VOLT\_F32/((float32)D\_ADCFULLSCALE\_CNT\_U16 \* (float32) D\_ADCCOMPOFFSAMPLESIZE\_CNT\_U16)) |
| D\_CNVRTP29TOP13\_CNT\_U16 | 1 | Cnt | 16U |
| D\_30DEGREES\_CNT\_U16 | 1 | Cnt | 5461U |
| D\_ONEDIVSQRT3\_F32 | Single Precision float | Cnt | 0.57735F |
| D\_POSITIVEONE\_CNT\_S8 | 1 | Cnt | 1 |
| D\_CURRDQMAX\_AMP\_F32 | Single Precision float | Amp | 220 |
| D\_MTRCURROFFHICOMOFF\_CNT\_U16 | 1 | Cnt | 4000U |
| D\_MTRCURROFFLOCOMOFF\_CNT\_U16 | 1 | Cnt | 500U |
| D\_CURROFFNOOFAVG\_CNT\_U16 | 1 | Cnt | 64U |
| D\_MTRCURROFFZEROCOMOFF\_CNT\_U16 | 1 | Cnt | 0U |
| D\_MINVCALCMD\_CNT\_F32 | Single Precision float | Cnt | 17500.0F |
|  |  |  |  |
|  |  |  |  |

#### 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\_2PI\_ULS\_F32 |
| D\_FALSE\_CNT\_LGC |
| D\_VECUMIN\_VOLTS\_F32 |
| D\_ZERO\_ULS\_F32 |
| D\_ZERO\_CNT\_U16 |
| D\_ZERO\_CNT\_U32 |
| D\_MTRPOLESDIV2\_CNT\_U8 |
| D\_2MS\_SEC\_F32 |

### Module specific Lookup Tables Constants

(This is for lookup tables (arrays) with fixed values, same name as other tables)

|  |  |  |  |
| --- | --- | --- | --- |
| 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. Cosf
2. Sinf
3. Limit\_m
4. Abs\_f32\_m
5. TableSize\_m
6. FPM\_FloatToFixed\_m
7. FPM\_FixedToFloat\_m
8. IntplVarXY\_s16\_s16Xs16Y\_Cnt
9. LPF\_SvUpdate\_u16InFixKTrunc\_m
10. LPF\_OpUpdate\_u16InFixKTrunc\_m
11. LPF\_SvUpdate\_s16InFixKTrunc\_m
12. LPF\_OpUpdate\_s16InFixKTrunc\_m
13. LPF\_KUpdate\_f32\_m
14. LPF\_OpUpdate\_f32\_m

## Data Hiding Functions

1. CmMtrCurr\_Read\_MRFMtrVel\_MtrRadpS\_f32
2. CmMtrCurr\_Read\_Vecu\_Volt\_f32
3. CmMtrCurr\_Read\_Phs1Curr\_Cnt\_u16
4. CmMtrCurr\_Read\_Phs2Curr\_Cnt\_u16
5. CmMtrCurr\_Read\_DCPhsAComp\_Cnt\_u16
6. CmMtrCurr\_Read\_DCPhsBComp\_Cnt\_u16
7. CmMtrCurr\_Read\_DCPhsCComp\_Cnt\_u16
8. CmMtrCurr\_Read\_MtrCurr1TempOffset\_Volt\_f32
9. CmMtrCurr\_Read\_MtrCurr2TempOffset\_Volt\_f32
10. CmMtrCurr\_Read\_MtrElecPol\_Cnt\_s08
11. CmMtrCurr\_Read\_MtrPosElec\_Rev\_u0p16
12. CmMtrCurr\_Write\_ElecPosDelayComp\_Rad\_f32
13. CmMtrCurr\_Write\_MtrCurrQax\_Amp\_f32
14. CmMtrCurr\_Write\_MtrCurrDax\_Amp\_f32
15. CmMtrCurr\_Write\_CorrMtrPosElec\_Rev\_f32
16. CmMtrCurr\_Write\_MtrCurrK1\_Amps\_f32
17. CmMtrCurr\_Write\_MtrCurrK2\_Amps\_f32
18. CmMtrCurr\_Write\_MtrCurr1\_Volts\_f32
19. CmMtrCurr\_Write\_MtrCurr2\_Volts\_f32

## Global Functions/Macros Defined by this Module

None

## Local Functions/Macros Used by this MDD only

None

# 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\_Vecu\_Volt\_f32 | 5 |

## Initialization Functions

### Init: CmMtrCurr\_Init

#### Design Rationale

None

#### Module Outputs

None

#### Module Internal

## IF ((Rte\_Pim\_ShCurrCal()->EOLMtrCurrVcalCmd\_VoltCnts\_f32) >= D\_MINVCALCMD\_CNT\_F32)

CmMtrCurr\_MtrCurr1OffDelta\_VoltpVoltCnt\_M\_f32 = ((Rte\_Pim\_ShCurrCal()->EOLMtrCurr1OffsetDiff\_Volts\_f32) / (Rte\_Pim\_ShCurrCal()->EOLMtrCurrVcalCmd\_VoltCnts\_f32))

CmMtrCurr\_MtrCurr2OffDelta\_VoltpVoltCnt\_M\_f32 = ((Rte\_Pim\_ShCurrCal()->EOLMtrCurr2OffsetDiff\_Volts\_f32) / (Rte\_Pim\_ShCurrCal()->EOLMtrCurrVcalCmd\_VoltCnts\_f32))

## ELSE

## CmMtrCurr\_MtrCurr1OffDelta\_VoltpVoltCnt\_M\_f32 = D\_ZERO\_ULS\_F32

## CmMtrCurr\_MtrCurr2OffDelta\_VoltpVoltCnt\_M\_f32 = D\_ZERO\_ULS\_F32

## END

LPF\_KUpdate\_f32\_m(k\_CurrCorrErrFiltFc\_Hz\_f32, D\_2MS\_SEC\_F32, &CmMtrCurr\_CurrCorrDiagKSV\_M\_str)

## Periodic Functions

### Per: CmMtrCurr\_Per1

#### Design Rationale

None

#### Program Flow Start

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

#### Store Module Inputs to Local copies

*FiltCntrlTemp\_DegC\_T\_f32=Rte\_IRead\_CmMtrCurr\_Per1\_FiltCntrlTemp\_DegC\_f32()*

#### Processing



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

Rte\_Iwrite\_CmMtrCurr\_Per1\_MtrCurr1TempOffset\_Volt\_f32(MtrCurr1TempOffset\_Volts\_T\_f32)

Rte\_Iwrite\_CmMtrCurr\_Per1\_MtrCurr2TempOffset\_Volt\_f32(MtrCurr2TempOffset\_Volts\_T\_f32)

#### Program Flow End

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

### Per: CmMtrCurr\_Per2

#### Design Rationale

None

#### Program Flow Start

Rte\_Call\_CmMtrCurr\_Per2\_CP0\_CheckpointReached()

#### Store Module Inputs to Local copies

MtrCurrAlpha\_Rev\_T\_f32=Rte\_Iread\_CmMtrCurr\_Per2\_MtrCurrAngle\_ Rev\_f32()

CorrMtrPosElec\_Rev\_T\_f32=Rte\_Iread\_CmMtrCurr\_Per2\_CorrMtrCurrPosition\_Rev\_f32()

MtrCurrK1\_Amps\_T\_f32=Rte\_Iread\_CmMtrCurr\_Per2\_MtrCurrK1\_Amp\_f32()

MtrCurrK2\_Amps\_T\_f32=Rte\_Iread\_CmMtrCurr\_Per2\_MtrCurrK2\_Amp\_f32()

ADCMtrCurr1\_Volts\_T\_f32=Rte\_Iread\_CmMtrCurr\_Per2\_ADCMtrCurr1\_Volts\_f32()

ADCMtrCurr2\_Volts\_T\_f32=Rte\_Iread\_CmMtrCurr\_Per2\_ADCMtrCurr2\_Volts\_f32()

#### Processing



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

None

#### Program Flow End

Rte\_Call\_CmMtrCurr\_Per2\_CP1\_CheckpointReached()

### Per: CmMtrCurr\_Per3

#### Design Rationale

None

#### Program Flow Start

Rte\_Call\_CmMtrCurr\_Per3\_CP0\_CheckpointReached()

#### Store Module Inputs to Local copies

ADCMtrCurr1\_Volts\_T\_f32=Rte\_Iread\_CmMtrCurr\_Per3\_ADCMtrCurr1\_Volts\_f32()

ADCMtrCurr2\_Volts\_T\_f32=Rte\_Iread\_CmMtrCurr\_Per3\_ADCMtrCurr2\_Volts\_f32()

Vecu\_Volt\_T\_f32=Rte\_Iread\_CmMtrCurr\_Per3\_Vecu\_Volt\_f32()

MtrVel\_MtrRadpS\_T\_f32 = Rte\_Iread\_CmMtrCurr\_Per3\_MtrVel\_MtrRadpS\_f32()

VehSpd\_Kph\_T\_f32= Rte\_Iread\_CmMtrCurr\_Per3\_VehSpd\_Kph\_f32()

VhSpdValid\_Cnt\_T\_lgc= Rte\_Iread\_CmMtrCurr\_Per3\_VhSpdValid\_Cnt\_lgc()

SrlComSvcDft\_Cnt\_T\_b32=Rte\_Iread\_CmMtrCurr\_Per3\_SrlComSvcDft\_Cnt\_b32()

CurroffProcessFlag\_T\_enum = CmMtrCurr\_CurroffProcessFlag\_M\_enum

#### Processing

   












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

Rte\_Iwrite\_CmMtrCurr\_Per3\_ComOffset\_Cnt\_u16(ComOffset\_Cnt\_T\_u16)

#### Program Flow End

Rte\_Call\_CmMtrCurr\_Per3\_CP1\_CheckpointReached()

## Fault Recovery Functions

None

## Shutdown Functions

None

## Interrupt Functions

### CurrDQPer1

#### Design Rationale

None

#### Program Flow Start

N/A

#### Store Module Inputs to Local Copies

CmMtrCurr\_Read\_MRFMtrVel\_MtrRadpS\_f32(&MRFMtrVel\_MtrRadpS\_T\_f32)

CmMtrCurr\_Read\_Vecu\_Volt\_f32(&Vecu\_Volt\_T\_f32)

CmMtrCurr\_Read\_Phs1Curr\_Cnt\_u16(&Phs1Curr\_Cnt\_T\_u16)

CmMtrCurr\_Read\_Phs2Curr\_Cnt\_u16(&Phs2Curr\_Cnt\_T\_u16)

CmMtrCurr\_Read\_MtrCurr1TempOffset\_Volt\_f32(&MtrCurr1TempOffset\_Volt\_T\_f32)

CmMtrCurr\_Read\_MtrCurr2TempOffset\_Volt\_f32(&MtrCurr2TempOffset\_Volt\_T\_f32)

CmMtrCurr\_Read\_MtrElecPol\_Cnt\_s08(&MtrElecPol\_Cnt\_T\_s08)

CmMtrCurr\_Read\_MtrPosElec\_Rev\_u0p16(&MtrPosElec\_Rev\_T\_u0p16)

CmMtrCurr\_Read\_ADC2OffsetComp\_Cnt\_u8p8(&ADC2OffsetComp\_Cnt\_T\_u8p8)

Phs1Curr\_Volts\_T\_f32 = D\_ZERO\_ULS\_F32

Phs2Curr\_Volts\_T\_f32 = D\_ZERO\_ULS\_F32

#### Processing















 



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

CmMtrCurr\_Write\_ElecPosDelayComp\_Rad\_f32(ElecPosDelayComp\_Rad\_T\_f32)

CmMtrCurr\_Write\_MtrCurrQax\_Amp\_f32(MtrCurrFinalQax\_Amps\_T\_f32)

CmMtrCurr\_Write\_MtrCurrDax\_Amp\_f32(MtrCurrFinalDax\_Amps\_T\_f32)

CmMtrCurr\_Write\_CorrMtrPosElec\_Rev\_f32(CorrMtrPosElec\_Rev\_T\_f32)

CmMtrCurr\_Write\_MtrCurrK1\_Amps\_f32(MtrCurrK1\_Amps\_T\_f32)

CmMtrCurr\_Write\_MtrCurrK2\_Amps\_f32(MtrCurrK2\_Amps\_T\_f32)

CmMtrCurr\_Write\_MtrCurr1\_Volts\_f32(Phs1Curr\_Volts\_T\_f32)

CmMtrCurr\_Write\_MtrCurr2\_Volts\_f32(Phs2Curr\_Volts\_T\_f32)

#### Program Flow End

N/A

## Serial Communication Functions

### Scomm: CmMtrCurrTempOffset\_Scom\_Get

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Function Name** | CmMtrCurrTempOffset\_Scom\_Get | Type | Min | Max | UTP Tol. |
| **Arguments Passed** | CurrTempOffCal | CurrTempOffsetType \* |  |  |  |
| **Return Value** | void | NA | NA | NA |  |

#### Design Rationale

None

#### Program Flow Start

None

#### Store Module Inputs to Local copies

None

#### Processing



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

None

#### Program Flow End

None

### Scomm: CmMtrCurrTempOffset\_Scom\_Set

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Function Name** | CmMtrCurrTempOffset\_Scom\_Set | Type | Min | Max | UTP Tol. |
| **Arguments Passed** | CurrTempOffCal | CurrTempOffsetType \* |  |  |  |
| **Return Value** | void | NA | NA | NA |  |

#### Design Rationale

None

#### Program Flow Start

None

#### Store Module Inputs to Local copies

None

#### Processing



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

None

#### Program Flow End

None

### Scomm: CmMtrCurr\_Scom\_CalGain

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Function Name** | CmMtrCurr\_Scom\_CalGain | Type | Min | Max | UTP Tol. |
| **Arguments Passed** | None | None |  |  |  |
| **Return Value** | RetrunValue | Std\_ReturnType |  |  |  |

#### Design Rationale

None

#### Program Flow Start

None

#### Store Module Inputs to Local copies

Rte\_Read\_MtrVel\_MtrRadpS\_f32(&MtrVel\_MtrRadpS\_T\_f32)

Rte\_Read\_VehSpd\_Kph\_f32(&VehSpd\_Kph\_T\_f32)

Rte\_Read\_VhSpdValid\_Cnt\_lgc(&VhSpdValid\_T\_Cnt\_lgc)

#### Processing



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

None

#### Program Flow End

None

### Scomm: CmMtrCurr\_Scom\_CalOffset

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Function Name** | CmMtrCurr\_Scom\_CalOffset | Type | Min | Max | UTP Tol. |
| **Arguments Passed** | None | None |  |  |  |
| **Return Value** | RetrunValue | Std\_ReturnType |  |  |  |

#### Design Rationale

None

#### Program Flow Start

None

#### Store Module Inputs to Local copies

Rte\_Read\_MtrVel\_MtrRadpS\_f32(&MtrVel\_MtrRadpS\_T\_f32)

Rte\_Read\_VehSpd\_Kph\_f32(&VehSpd\_Kph\_T\_f32)

Rte\_Read\_VhSpdValid\_Cnt\_lgc(&VhSpdValid\_T\_Cnt\_lgc)

#### Processing



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

Rte\_Write\_CurrentGainSvc\_Cnt\_lgc(CmMtrCurr\_CurrentGainSvc\_Cnt\_M\_lgc)

#### Program Flow End

None

### Scomm: CmMtrCurr\_Scom\_MtrCurrOffReadStatus

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Function Name** | CmMtrCurr\_Scom\_MtrCurrOffReadStatus | Type | Min | Max | UTP Tol. |
| **Arguments Passed** | CurrOffStatus | MtrCurrOffProcessFlag \* |  |  |  |
| **Return Value** | void | NA | NA | NA |  |

#### Design Rationale

None

#### Program Flow Start

None

#### Store Module Inputs to Local copies

None

#### Processing



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

None

#### Program Flow End

None

### Scomm: CmMtrCurr\_Scom\_ReadMtrCurrCals

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Function Name** | CmMtrCurr\_Scom\_ReadMtrCurrCals | Type | Min | Max | UTP Tol. |
| **Arguments Passed** | ShCurrCalPtr | PhaseCurrCal\_DataType \* |  |  |  |
| **Return Value** | void | NA | NA | NA |  |

#### Design Rationale

None

#### Program Flow Start

None

#### Store Module Inputs to Local copies

None

#### Processing



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

None

#### Program Flow End

None

### Scomm: CmMtrCurr\_Scom\_SetMtrCurrCals

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Function Name** | CmMtrCurr\_Scom\_SetMtrCurrCals | Type | Min | Max | UTP Tol. |
| **Arguments Passed** | ShCurrCalPtr | PhaseCurrCal\_DataType \* |  |  |  |
| **Return Value** | void | NA | NA | NA |  |

#### Design Rationale

None

#### Program Flow Start

None

#### Store Module Inputs to Local copies

None

#### Processing



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

None

#### Program Flow End

None

# Execution Requirements

## 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 |
| CmMtrCurr\_Init | On Init | Init |
| CmMtrCurr\_Per1 | 100ms | All |
| CmMtrCurr\_Per2 | 2ms | Operate |
| CmMtrCurr\_Per3 | 2ms | Operate |
| CurrDQPer1 | 125us (MtrCtrl ISR) | All |

## Execution Requirements for Serial Communication Functions

|  |  |
| --- | --- |
| Function Name | Sub-Module called by (Serial Comm Function Name) |
| CmMtrCurr\_Scom\_CalGain |  |
| CmMtrCurr\_Scom\_CalOffset |  |
| CmMtrCurr\_Scom\_MtrCurrOffReadStatus |  |
| CmMtrCurr\_Scom\_ReadMtrCurrCals |  |
| CmMtrCurr\_Scom\_SetMtrCurrCals |  |
| CmMtrCurrTempOffset\_Scom\_Get |  |
| CmMtrCurrTempOffset\_Scom\_Set |  |
|  |  |

# 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 |
| CmMtrCurr\_Init | RTE\_START\_SEC\_SA\_CMMTRCURR\_APPL\_CODE |
| CmMtrCurr\_Per1 | RTE\_START\_SEC\_SA\_CMMTRCURR\_APPL\_CODE |
| CmMtrCurr\_Per2 | RTE\_START\_SEC\_SA\_CMMTRCURR\_APPL\_CODE |
| CmMtrCurr\_Per3 | RTE\_START\_SEC\_SA\_CMMTRCURR\_APPL\_CODE |
| CurrDQPer1 |  |
| CmMtrCurr\_Scom\_CalGain | RTE\_START\_SEC\_SA\_CMMTRCURR\_APPL\_CODE |
| CmMtrCurr\_Scom\_CalOffset | RTE\_START\_SEC\_SA\_CMMTRCURR\_APPL\_CODE |
| CmMtrCurr\_Scom\_MtrCurrOffReadStatus | RTE\_START\_SEC\_SA\_CMMTRCURR\_APPL\_CODE |
| CmMtrCurr\_Scom\_ReadMtrCurrCals | RTE\_START\_SEC\_SA\_CMMTRCURR\_APPL\_CODE |
| CmMtrCurr\_Scom\_SetMtrCurrCals | RTE\_START\_SEC\_SA\_CMMTRCURR\_APPL\_CODE |
| CmMtrCurrTempOffset\_Scom\_Get | RTE\_START\_SEC\_SA\_CMMTRCURR\_APPL\_CODE |
| CmMtrCurrTempOffset\_Scom\_Set | RTE\_START\_SEC\_SA\_CMMTRCURR\_APPL\_CODE |

## Local Functions

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

|  |  |
| --- | --- |
| Name of Sub Module | Software Segment |
|  |  |

# Known Issues / Limitations With Design

1. INLINE functions defined in GlobalMacro.h are not unit tested.
2. ADC component outputs the ADC calibration compensation which is in turn used by Current Measurement to correct its offset volt . This is implementation done across two components. The measured currents from ADC conversion from counts to volts is done in this component instead of ADC component as described by the FDD

# Revision Control Log

|  |  |  |  |
| --- | --- | --- | --- |
| **Rev #** | **Change Description** | **Date** | **Author Initials** |
| 1.0 | Initial Version (FDD 01 C Ver 005) | 01-Dec-12 | Selva |
| 2.0 | Configured global read/write macros for global data (anomaly 4696) | 23-Mar-13 | OT |
| 3.0 | Fixes for Anamoly 5561 and A5566 added | 4-Sep-13 | Selva |
| 4.0 | Updated to FDD 01C Ver 006 | 07-Oct-13 | VK |
| 5.0 | Anomaly 5967 and 5873 | 06-Nov-13 | SP |
| 6.0 | Range corrections for ‘MtrCurr1OffDelta\_VoltpVoltCnts\_M\_f32’ and ‘MtrCurr2OffDelta\_VoltpVoltCnts\_M\_f32’ | 9-Nov-13 | SR |
| 7.0 | Changed CurrCorrDiag filtering from fixed point to floating point to achieve specified range and resolution – CR 10895 | 20-Nov-13 | KMC |
| 8.0 | Updated for FDD 01C v007 | 20-Apr-14 | Selva |
| 9.0 | Updated for FDD 01C v008 | 27-June-14 | Selva |
| 10.0 | Updated for FDD 01C v009 | 20-May-16 | NS |