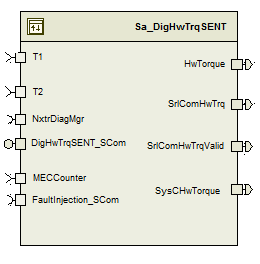
# Module -- Digital Handwheel Torque Function (SENT)

# High-Level Description

This module computes the digital handwheel torque signal from the SENT digital sensor inputs. It takes the sensor inputs, calculates the hw torque, compensates for trim, applies filtering and limits, and outputs the handwheel torque in HwNm. It uses long term correlated compensation to provide a T1 vs T2 correlation fault diagnostic. It also contains the service calls for a trim to be set or cleared.

# 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 | |
| T1\_HwNm\_f32 | | HwTorque\_HwNm\_f32 |
| T2\_HwNm\_f32 | | SrlComHwTrq\_HwNm\_f32 |
| MECCounter\_Cnt\_enum | | SrlComHwTrqValid\_Cnt\_Lgc |
|  | | SysCHWTorque |

## 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 |
| DigHwTrqSENT\_T1\_HwNm\_M\_f32 | See Data Dictionary | See Data Dictionary | See Data Dictionary | DIGHWTRQSENT\_START\_SEC\_VAR\_CLEARED\_32 |
| DigHwTrqSENT\_T2\_HwNm\_M\_f32 | See Data Dictionary | See Data Dictionary | See Data Dictionary | DIGHWTRQSENT\_START\_SEC\_VAR\_CLEARED\_32 |
| DigHwTrqSENT\_HwTrq\_HwNm\_M\_f32 | See Data Dictionary | See Data Dictionary | See Data Dictionary | DIGHWTRQSENT\_START\_SEC\_VAR\_CLEARED\_32 |
| DigHwTrqSENT\_TDiagFiltOut\_HwNm\_M\_f32 | See Data Dictionary | See Data Dictionary | See Data Dictionary | DIGHWTRQSENT\_START\_SEC\_VAR\_CLEARED\_32 |
| DigHwTrqSENT\_SSDiagFiltOut\_HwNm\_M\_f32 | See Data Dictionary | See Data Dictionary | See Data Dictionary | DIGHWTRQSENT\_START\_SEC\_VAR\_CLEARED\_32 |
| DigHwTrqSENT\_CMCFiltOut\_HwNm\_M\_f32 | See Data Dictionary | See Data Dictionary | See Data Dictionary | DIGHWTRQSENT\_START\_SEC\_VAR\_CLEARED\_32 |
| DigHwTrqSENT\_TrqSum\_HwNm\_M\_f32 | See Data Dictionary | See Data Dictionary | See Data Dictionary | DIGHWTRQSENT\_START\_SEC\_VAR\_CLEARED\_32 |
| DigHwTrqSENT\_DigHwTrqKSV\_M\_str | See Data Dictionary | See Data Dictionary | See Data Dictionary | DIGHWTRQSENT\_START\_SEC\_VAR\_CLEARED\_UNSPECIFIED |
| DigHwTrqSENT\_TDiagFiltKSV\_M\_str | See Data Dictionary | See Data Dictionary | See Data Dictionary | DIGHWTRQSENT\_START\_SEC\_VAR\_CLEARED\_UNSPECIFIED |
| DigHwTrqSENT\_SSFiltKSV\_M\_str | See Data Dictionary | See Data Dictionary | See Data Dictionary | DIGHWTRQSENT\_START\_SEC\_VAR\_CLEARED\_UNSPECIFIED |
| DigHwTrqSENT\_CMCFiltKSV\_M\_str | See Data Dictionary | See Data Dictionary | See Data Dictionary | DIGHWTRQSENT\_START\_SEC\_VAR\_CLEARED\_UNSPECIFIED |
| DigHwTrqSENT\_NvMBlkStatus\_ Cnt\_M\_u8 | See Data Dictionary | See Data Dictionary | See Data Dictionary | DIGHWTRQSENT\_START\_SEC\_VAR\_CLEARED\_8 |
| DigHwTrqSENT\_CMCFiltSV\_HwNm\_M\_f32 | See Data Dictionary | See Data Dictionary | See Data Dictionary | DIGHWTRQSENT\_START\_SEC\_VAR\_SAVED\_ZONEH\_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) |
| DigHwTrqSENTTrim\_DataType | kEOLHwTrqTrim\_HwNm\_f32 | float | FULL | FULL |
|  | kEOLHwTrqTrimPerformed\_Cnt\_Lgc | boolean | FALSE | TRUE |

## Module Display Variables

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Variable Name | | Resolution | Legal Range  (min) | Legal Range  (max) | Software Segment |
| DigHwTrqSENT\_SumFiltOut\_HwNm\_D\_f32 | See Data Dictionary | See Data Dictionary | See Data Dictionary | DIGHWTRQSENT\_START\_SEC\_VAR\_CLEARED\_32 |
| DigHwTrqSENT\_CorrDiagFiltOut\_HwNm\_D\_f32 | See Data Dictionary | See Data Dictionary | See Data Dictionary | DIGHWTRQSENT\_START\_SEC\_VAR\_CLEARED\_32 |
| DigHwTrqSENT\_DigHwTrq\_HwNm\_D\_f32 | See Data Dictionary | See Data Dictionary | See Data Dictionary | DIGHWTRQSENT\_START\_SEC\_VAR\_CLEARED\_32 |
| DigHwTrqSENT\_HwTrqTrim\_HwNm\_D\_f32 | See Data Dictionary | See Data Dictionary | See Data Dictionary | DIGHWTRQSENT\_START\_SEC\_VAR\_CLEARED\_32 |
| DigHwTrqSENT\_TmpDigHwTrq\_HwNm\_D\_f32 | See Data Dictionary | See Data Dictionary | See Data Dictionary | DIGHWTRQSENT\_START\_SEC\_VAR\_CLEARED\_32 |

# 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\_HwTrqLPFFc\_Hz\_f32 |
| k\_T1vsT2TrqSum\_HwNm\_f32 |
| k\_T1T2TransFltLim\_HwNm\_f32 |
| k\_T1T2CMCLPFEnable\_HwNm\_f32 |
| k\_T1T2CMCLPFFc\_Hz\_f32 |
| k\_CMCLPFOutLim\_HwNm\_f32 |
| k\_T1T2SSLPFFc\_Hz\_f32 |
| k\_T1T2SSLim\_HwNm\_f32 |
| t\_T1T2DepTrsTimecon\_Hz\_u6p10 |
| t\_T1T2IndTrsTimecon\_HwNm\_u5p11 |
| k\_MaxHwTrqTrim\_HwNm\_f32 |

## 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\_DIFFTRQSCALE\_ULS\_F32 | Single precision floating point | unitless | 2.0 |
| D\_HWTRQMAXRANGE\_HWNM\_F32 | Single precision floating point | HwNm | 10.0 |
| D\_TRIMPERFORMED\_CNT\_LGC | boolean | boolean | TRUE |
| D\_TRIMNOTPERFORMED\_CNT\_LGC | boolean | boolean | FALSE |
| D\_MFGMODE\_CNT\_ENUM | enum | counts | ManufacturingMode |
| D\_ONERESOLUTIONCOUNT\_HWNM\_F32 | Single precision floating point | HwNm | 0. 00390625f |
| D\_DEFDIGHWTRQTRIM\_HWNM\_F32 | Single precision floating point | HwNm | 0.0 |
| D\_DEFSSDIAGFILTOUT\_HWNM\_F32 | Single precision floating point | HwNm | 0.0 |
| D\_HWTRQLPFSAMPLEINT\_SEC\_F32 | Single precision floating point | sec | D\_2MS\_SEC\_F32 |
| D\_TDIAGLPFSAMPLEINT\_SEC\_F32 | Single precision floating point | sec | 0.004 |
| D\_CMCLPFSAMPLEINT\_SEC\_F32 | Single precision floating point | sec | 0.1 |
| D\_SSLPFSAMPLEINT\_SEC\_F32 | Single precision floating point | sec | 0.1 |
| D\_SSFILTSVLMT\_HWNM\_F32 | Single precision floating point | HwNm | k\_T1T2SSLim\_HwNm\_f32 + D\_ONERESOLUTIONCOUNT\_HWNM\_F32 |
| D\_FAILEDANDFAILEDTHISOPCYCLE\_CNT\_U08 | 1 | counts | D\_TESTFAILEDBIT\_CNT\_B8 | D\_TESTFAILEDTHISOPCYCLEBIT\_CNT\_B8 |

#### 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 |
| NULL\_PTR |
| D\_ZERO\_ULS\_F32 |
| D\_2MS\_SEC\_F32 |
| RTE\_E\_OK |
| NVM\_REQ\_OK |
| FLTINJ\_DIGHWTRQSENT\_T1FAULT |
| FLTINJ\_DIGHWTRQSENT\_T2FAULT |

### 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. Limit\_m()
2. Abs\_f32\_m()
3. FPM\_FloatToFixed\_m()
4. FPM\_FixedToFloat\_m
5. IntplVarXY\_u16\_u16Xu16Y\_Cnt()
6. LPF\_Init\_f32\_m()
7. LPF\_OpUpdate\_f32\_m ()
8. LPF\_KUpdate\_f32\_m()
9. Tablesize\_m()

## Data Hiding Functions

1. Rte\_Call\_NxtrDiagMgr\_SetNTCStatus()
2. Rte\_Call\_NxtrDiagMgr\_GetNTCStatus()
3. Rte\_Pim\_DigTrqTrim()

## Global Functions/Macros Defined by this Module

None

## Local Functions/Macros Used by this MDD only

### Trim Not Performed Diagnostic

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Function Name** | TrimNotPerfDiag | Type | Min | Max | UTP Tol. |
| **Arguments Passed** | MECCounter\_Cnt\_T\_enum | ManufModeType | 0 | 2 | n/a |
| **Return Value** | n/a |  |  |  |  |

#### 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\_T1\_HwNm\_f32 | 0.0 |
| Rte\_InitValue\_T2\_HwNm\_f32 | 0.0 |
| Rte\_InitValue\_HwTorque\_HwNm\_f32 | 0.0 |
| Rte\_InitValue\_SrlComHwTrq\_HwNm\_f32 | 0.0 |
| Rte\_InitValue\_SrlComHwTrqValid | FALSE |
| Rte\_InitValue\_MECCounter\_Cnt\_enum | 0 |
| Rte\_InitValue\_SysCHwTorque | 0 |

## Initialization Functions

### Init: DigHwTrqSENT\_Init1

#### Design Rationale

TDiag filter does not need initialization because the state variable initialization value is zero and the filter constant is calculated each time the filter is updated.

#### Module Outputs

None

#### Module Internal

#### Check NvM Error Status



## Periodic Functions

### Per: DigHwTrqSENT\_Per1

#### Design Rationale

None

#### Program Flow Start

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

#### Store Module Inputs to Local copies

DigHwTrqSENT\_T1\_HwNm\_M\_f32 = Rte\_IRead\_\_DigHwTrqSENT\_Per1\_T1\_HwNm\_f32();

DigHwTrqSENT\_T2\_HwNm\_M\_f32 = Rte\_IRead\_DigHwTrqSENT\_Per1\_T2\_HwNm\_f32();

#### Handwheel Torque Calculation



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

DigHwTrqSENT\_TmpDigHwTrq\_HwNm\_D\_f32 = TmpDigHwTrq\_HwNm\_T\_f32

DigHwTrqSENT\_DigHwTrq\_HwNm\_D\_f32 = DigHwTrq\_HwNm\_T\_f32

Rte\_IWrite\_DigHwTrqSENT\_Per1\_HwTorque\_HwNm\_f32(DigHwTrqSENT\_HwTrq\_HwNm\_M\_f32)

Rte\_IWrite\_DigHwTrqSENT\_Per1\_SysCHwTorque\_HwNm\_f32(DigHwTrqSENT\_HwTrq\_HwNm\_M\_f32);

#### Program Flow End

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

### Per: DigHwTrqSENT\_Per2

#### Design Rationale

None

#### Program Flow Start

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

#### Store Module Inputs to Local copies

None

#### T1 vs T2 Comparison Diagnostic



#### Serial Comm Outputs



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

Rte\_IWrite\_DigHwTrqSENT\_Per2\_SrlComHwTrq\_HwNm\_f32 (DigHwTrqSENT\_HwTrq\_HwNm\_M\_f32)

Rte\_IWrite\_DigHwTrqSENT\_Per2\_SrlComHwTrqValid\_Cnt\_lgc (SrlComHwTrqValid\_Cnt\_T\_lgc)

#### Program Flow End

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

### Per: DigHwTrqSENT\_Per3

#### Design Rationale

None

#### Program Flow Start

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

#### Store Module Inputs to Local copies

None

#### Steady State Filter



#### Common Mode Compensation



#### Program Flow End

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

## Fault Recovery Functions

None

## Shutdown Functions

None

## Interrupt Functions

None

## Serial Communication Functions

### Scomm: DigHwTrqSENT\_Scom\_ClrTrqTrim

#### Design Rationale

None

#### Program Flow Start

n/a

#### Store Module Inputs to Local copies

None

#### Clear Handwheel Torque Trim



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

None

#### Program Flow End

n/a

### Scomm: DigHwTrqSENT\_Scom\_SetTrqTrim

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Function Name** | DigHwTrqSENT\_SCom\_SetTrqTrim | Type | Min | Max |
| **Arguments Passed** | NA |  |  |  |
| **Return Value** | RetValue | Uint8 | 0 | 34 |

#### Design Rationale

None

#### Program Flow Start

n/a

#### Store Module Inputs to Local copies

None

#### Set Handwheel Torque Trim



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

DigHwTrqSENT\_HwTrqTrim\_HwNm\_D\_f32 = HwTrqTrim\_HwNm\_T\_f32

#### Program Flow End

n/a

### Scomm: DigHwTrqSENT\_Scom\_TrimData

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Function Name** | DigHwTrqSENT\_SCom\_ TrimData | Type | Min | Max |
| **Arguments Passed** | EOLHwTrqTrim\_HwNm\_f32  EOLHwTrqTrimPerformed\_Cnt\_Lgc | Float  Boolean |  |  |
| **Return Value** | n/a |  |  |  |

#### Design Rationale

None

#### Program Flow Start

n/a

#### Store Module Inputs to Local copies

None

#### Handwheel Torque Trim Data



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

#### Program Flow End

n/a

### Scomm: DigHwTrqSENT\_Scom\_WriteData

#### Design Rationale

None

#### Program Flow Start

n/a

#### Store Module Inputs to Local copies

None

#### Handwheel Torque Write Data



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

n/a

#### Program Flow End

n/a

# Execution Requirements

## Execution Sequence of the Module

## 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 |
| DigHwTrqSENT\_Init1() | Once (at initialization) | STARTUP |
| DigHwTrqSENT\_Per1() | 2 ms | ALL |
| DigHwTrqSENT\_Per2() | 4 ms | ALL |
| DigHwTrqSENT\_Per3() | 100 ms | ALL |

## Execution Requirements for Serial Communication Functions

|  |  |
| --- | --- |
| Function Name | Sub-Module called by (Serial Comm Function Name) |
| DigHwTrqSENT\_SCom\_ClrTrqTrim () | EPSInternalRoutineControl() |
| DigHwTrqSENT\_SCom\_SetTrqTrim () | EPSInternalRoutineControl() |
| DigHwTrqSENT\_SCom\_TrimData() | EPSInternalRoutineControl() |
| DigHwTrqSENT\_SCom\_WriteData() | EPSInternalRoutineControl() |

# 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 |
| DigHwTrqSENT\_Init1() | RTE\_ SA\_DIGHWTRQSENT\_APPL\_CODE |
| DigHwTrqSENT\_Per1() | RTE\_ SA\_DIGHWTRQSENT\_APPL\_CODE |
| DigHwTrqSENT\_Per2() | RTE\_ SA\_DIGHWTRQSENT\_APPL\_CODE |
| DigHwTrqSENT\_Per3() | RTE\_ SA\_DIGHWTRQSENT\_APPL\_CODE |
| DigHwTrqSENT\_SCom\_ClrTrqTrim () | RTE\_ SA\_DIGHWTRQSENT\_APPL\_CODE |
| DigHwTrqSENT\_SCom\_SetTrqTrim () | RTE\_ SA\_DIGHWTRQSENT\_APPL\_CODE |
| DigHwTrqSENT\_SCom\_TrimData() | RTE\_SA\_DIGHWTRQSENT\_APPL\_CODE |
| DigHwTrqSENT\_SCom\_WriteData() | RTE\_SA\_DIGHWTRQSENT\_APPL\_CODE |

## Local Functions

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

|  |  |
| --- | --- |
| Name of Sub Module | Software Segment |
| TrimNotPerfDiag | SA\_DIGHWTRQSENT\_CODE |

# Known Issues / Limitations With Design

1. INLINE functions defined in globalmacro.h are not unit tested
2. Serial communication outputs are being processed in the DigHwTrqSENT\_Per2() function which is called every 4ms. Processing is done at 4 ms instead of 2 ms for processor throughput considerations. If the actual serial outputs are transmitted at a rate that is not a multiple of 4 ms, there will be corresponding jitter in the outputs. E.g. if the outputs are transmitted at 10 ms, the torque value will be updated over an 8ms interval one time, and a 12 ms interval the next.

Revision Control Log

|  |  |  |  |
| --- | --- | --- | --- |
| **Rev #** | **Change Description** | **Date** | **Author Initials** |
| 1.0 | Initial Version per ES04C\_HWTrqFunc\_v003.mdl and ES\_04C\_HwTrqFunc\_Calibrations\_Constants.m | 01-Jul-13 | KMC |
| 2.0 | Correct name of HwTrq\_HwNm\_M\_f32 in section 3.1. Add RTE\_E\_OK and NVM\_REQ\_OK in section 4.2.1.2. | 15-Jul-13 | KMC |
| 3.0 | Correct reads from SCom functions to be direct rather than buffered. | 31-Jul-13 | KMC |
| 4.0 | Add Scom function to read TrimData | 14-Jan-14 | LK |
| 5.0 | Updated MDD to match with latest source file | 16-Jan-14 | LK |
| 6.0 | Updated per Design Review CR 11619 | 03-Mar-14 | SB |
| 7.0 | Updated per Design Review CR 11619 | 27-Mar-14 | SB |
| 8.0 | Implemented ES04C Rev 006 | 09-Jun-14 | SB |
| 9.0 | Implemented ESC04C Rev 007 and added the function DigHwScomm\_WriteData | 30-July-14 | VS |
| 10.0 | Updated to match source code 12242 | 04-Aug-14 | SB |