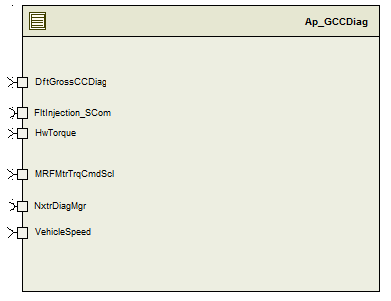
# Module – Gross Cross Check Diagnostic

# High-Level Description

This module computes the Gross Cross Check Diagnostics. It takes the handwheel torque,vehicle speed ,Motor Nm to diagnose .

# 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 | |
| HwTorque\_HwNm\_f32 | |  |
| DftGrossCCDiag\_Cnt\_lgc | |  |
| MRFMtrTrqCmdScl\_MtrNm\_f32 | |  |
| VehicleSpeed\_Kph\_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 |
| GCCDiag\_HwTrqLPF\_M\_str | See Data Dictionary | See Data Dictionary | See Data Dictionary | GCCDIAG\_START\_SEC\_VAR\_CLEARED\_UNSPECIFIED |
| GCCDiag\_MtrTrqLPF\_M\_str | See Data Dictionary | See Data Dictionary | See Data Dictionary | GCCDIAG\_START\_SEC\_VAR\_CLEARED\_UNSPECIFIED |
| GCCDiag\_PNAccumulator\_Cnt\_M\_u16 | See Data Dictionary | See Data Dictionary | See Data Dictionary | GCCDIAG\_START\_SEC\_VAR\_CLEARED\_16 |

### 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) |
| None |  |  |  |  |

## Module Display Variables

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Variable Name | | Resolution | Legal Range  (min) | Legal Range  (max) | Software Segment |
| GCCDiag\_PNAccumulator\_Cnt\_D\_u16 | See Data Dictionary | See Data Dictionary | See Data Dictionary | GCCDIAG\_START\_SEC\_VAR\_CLEARED\_16 |

# 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\_GCC\_PNSettings\_str |
| t\_GCC\_VehSpd\_Kph\_u9p7 |
| t2\_GCC\_UprBoundX\_HwNm\_s4p11 |
| t2\_GCC\_UprBoundY\_MtrNm\_u4p12 |
|  |

## 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\_FILTERFRQ\_HZ\_F32 | Single precision floating point | Hetrz | 1 |

#### 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\_NEGONE\_CNT\_S16 |
| 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. FPM\_FloatToFixed\_m()
2. FPM\_FixedToFloat\_m
3. BilinearXMYM\_u16\_s16XMu16YM\_Cnt ()
4. LPF\_Init\_f32\_m()
5. LPF\_OpUpdate\_f32\_m ()
6. LPF\_KUpdate\_f32\_m()
7. Tablesize\_m()
8. DiagPStep\_m()
9. DiagNStep\_m()
10. DiagFailed\_m()

## Data Hiding Functions

1. Rte\_Call\_NxtrDiagMgr\_SetNTCStatus()

## 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 |
| None |  |

## Initialization Functions

### Init: GCCDiag\_Init1

#### Design Rationale

The two filters need the initialization of filter constants only and the state variables are initialized to zero by MemMap section.

#### Module Outputs

None

#### Module Internal

LPF\_KUpdate\_f32\_m(D\_FILTERFRQ\_HZ\_F32, D\_2MS\_SEC\_F32, &GCCDiag\_HwTrqLPF\_M\_str);

LPF\_KUpdate\_f32\_m(D\_FILTERFRQ\_HZ\_F32, D\_2MS\_SEC\_F32, &GCCDiag\_MtrTrqLPF\_M\_str);

## Periodic Functions

### Per: GCCDiag\_Per1

#### Design Rationale

The Code has been optimized to remove the Inverter for the variable MtrCmdOk\_T\_lgc , the functionality is same as FDD.

#### Program Flow Start

N/A

#### Store Module Inputs to Local copies

/\* Read the Inputs to the Temporary Variables \*/

HwTorque\_HwNm\_T\_f32 = Rte\_IRead\_GCCDiag\_Per1\_HwTorque\_HwNm\_f32();

VehicleSpeed\_Kph\_T\_f32 = Rte\_IRead\_GCCDiag\_Per1\_VehicleSpeed\_Kph\_f32();

MRFMtrTrqCmdScl\_MtrNm\_T\_f32 = Rte\_IRead\_GCCDiag\_Per1\_MRFMtrTrqCmdScl\_MtrNm\_f32();

DftGrossCCDiag\_Cnt\_T\_lgc = Rte\_IRead\_GCCDiag\_Per1\_DftGrossCCDiag\_Cnt\_lgc();

/\* Fault Injection for the HwTrq,MtrTrq and VehSpd \*/

#if (STD\_ON == BC\_GCCDIAG\_FAULTINJECTIONPOINT)

Rte\_Call\_FltInjection\_SCom\_FltInjection(&HwTorque\_HwNm\_T\_f32, FLTINJ\_GCCDIAG\_HWTRQ);

Rte\_Call\_FltInjection\_SCom\_FltInjection(&VehicleSpeed\_Kph\_T\_f32, FLTINJ\_GCCDIAG\_VEHSPD);

Rte\_Call\_FltInjection\_SCom\_FltInjection(&MRFMtrTrqCmdScl\_MtrNm\_T\_f32, FLTINJ\_GCCDIAG\_MTRTRQ);

#endif

#### Gross Cross Check Diagnostics





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

#### Program Flow End

N/A

## Fault Recovery Functions

None

## Shutdown Functions

None

## Interrupt Functions

None

# 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 |
| GCCDiag\_Init1() | Once (at initialization) | COLD INIT |
| GCCDiag\_Per1() | 2 ms | ALL |

## Execution Requirements for Serial Communication Functions

|  |  |
| --- | --- |
| Function Name | Sub-Module called by (Serial Comm Function Name) |
| None |  |

# 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 |
| GCCDiag\_Init1() | RTE\_AP\_GCCDIAG\_APPL\_CODE |
| GCCDiag\_Per1() | RTE\_AP\_GCCDIAG\_APPL\_CODE |

## Local Functions

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

|  |  |
| --- | --- |
| Name of Sub Module | Software Segment |
| None |  |

# Known Issues / Limitations With Design

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

Revision Control Log

|  |  |  |  |
| --- | --- | --- | --- |
| **Rev #** | **Change Description** | **Date** | **Author Initials** |
| 1.0 | Initial Version per SF46 Gross Cross Check Diagnostics | 26-Aug-14 | VS |