# Module --

# High-Level Description

Handle internal Nexteer Trouble Codes (NTCs) and update the Diagnostic Event Manager (DEM) via a client call to the DEM Interface module which handles the customer specific DTC setting requirements (e.g. DTC masking).

# Figures

## Component Diagram

DiagMgr.tif

# Variable Data Dictionary

|  |  |  |
| --- | --- | --- |
| Module Inputs | Module Outputs | |
| MEC\_Cnt\_enum | | DiagRmpToZeroActive\_Cnt\_lgc |
| MfgDiagInhibit\_Cnt\_lgc | | DiagStsDefTemp\_Cnt\_lgc |
|  | | DiagStsHWASbSystmFltPres\_Cnt\_lgc |
|  | | DiagStsCtrldDisRmpPres\_Cnt\_lgc |
|  | | DiagStsDefVehSpd\_Cnt\_lgc |
|  | | DiagRampRate\_XpmS\_f32 |
|  | | DiagRampValue\_Uls\_f32 |
|  | | DiagStsDefHWAScomExpVal\_Cnt\_lgc |
|  | | DiagStsWIRDisable\_Cnt\_lgc |
|  | | DiagStsNonRecRmpToZeroFltPres\_Cnt\_lgc |
|  | | DiagStsRecRmpToZeroFltPres\_Cnt\_lgc |

## Module Internal Variables

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Variable Name | Datatype | Resolution | Legal Range  (min) | Legal Range  (max) | Software Segment  {Data Type} |
| NTCInfo\_Cnt\_M\_str[512] | NTCInfo\_Str | N/A | (see structure definition) | (see structure definition) | DIAGMGR\_START\_SEC\_VAR\_CLEARED\_UNSPECIFIED |
| RspWhenProcessing\_Cnt\_M\_b32 | uint32 | 1 | FULL | FULL | DIAGMGR\_START\_SEC\_VAR\_CLEARED\_32 |
| RampDownWhenProcessing\_Cnt\_M\_lgc | boolean | N/A | FALSE | TRUE | DIAGMGR\_START\_SEC\_VAR\_CLEARED\_UNSPECIFIED |
| ActiveRmpRate\_UlspmS\_M\_f32 | float32 | Single precision float | 0.0001 | 0.5 | DIAGMGR\_START\_SEC\_VAR\_CLEARED\_32 |
| DiagStatus\_Cnt\_D\_b16 | uint16 | 1 | FULL | FULL | DIAGMGR\_START\_SEC\_VAR\_CLEARED\_16 |
| NTCInfoQueue\_Cnt\_M\_str[10] | NTCInfoQueue\_Str | N/A | (see structure definition) | (see structure definition) | DIAGMGR\_START\_SEC\_VAR\_CLEARED\_UNSPECIFIED |
| DiagMgrInitialized\_Cnt\_M\_lgc | boolean | N/A | FALSE | TRUE | DIAGMGR\_START\_SEC\_VAR\_CLEARED\_UNSPECIFIED |
| NTCInfoQueueIndex\_Cnt\_M\_u16 | uint16 | 1 | 0 | 10 | DIAGMGR\_START\_SEC\_VAR\_CLEARED\_16 |

### User defined typedef definition/declaration

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Typedef Name | Element Name | User Defined Type | Legal Range  (min) | Legal Range  (max) |
| typedef struct { } NTCInfo\_Str | Param | uint8 | 0 | 255 |
| Status | uint8 | 0 | 255 |
| typedef struct { } FltRsp\_Str | Response | uint32 : 24 | FULL | FULL |
| DEMEventID | uint32 : 8 | 0 | 255 |
| typedef struct { } NTCInfoQueue\_Str | NTC | NTCNumber | 0 | 511 |
| Param | uint8 | 0 | 255 |
| Status | NxtrDiagMgrStatus | NTC\_STATUS\_PASSED  NTC\_STATUS\_FAILED  NTC\_STATUS\_PREPASSED  NTC\_STATUS\_PREFAILED | |

# Constant Data Dictionary

## Calibration Constants

|  |
| --- |
| Constant Name |
| k\_FltRspTbl\_Cnt\_str[] |
| k\_FltRmpRate\_UlspmS\_f32[] |

## Program(fixed) Constants

### Embedded Constants

#### Local

|  |  |  |  |
| --- | --- | --- | --- |
| Constant Name | Resolution | Units | Value |
| D\_FLTRSPNTCACTIVEBIT\_CNT\_B32 | N/A | Counts | 0x00800000 |
| D\_FLTRSPRECOVERABLEBIT\_CNT\_B32 | N/A | Counts | 0x00400000 |
| D\_FLTRSPHWASBSYSTMFLTBIT\_CNT\_B32 | N/A | Counts | 0x00200000 |
| D\_FLTRSPDEFVEHSPDBIT\_CNT\_B32 | N/A | Counts | 0x00100000 |
| D\_FLTRSPDEFTEMPBIT\_CNT\_B32 | N/A | Counts | 0x00080000 |
| D\_FLTRSPSCOMHWANOTVALIDBIT\_CNT\_B32 | N/A | Counts | 0x00040000 |
| D\_FLTRSPWIRDISABLEBIT\_CNT\_B32 | N/A | Counts | 0x00008000 |
| D\_FLTRSPNTCINHIBITRUNBIT\_CNT\_B32 | N/A | Counts | 0x00000040 |
| D\_FLTRSPNTCINHIBITNOTOPERATEBIT\_CNT\_B32 | N/A | Counts | 0x00000020 |
| D\_FLTRSPRAMPBITS\_CNT\_B32 | N/A | Counts | 0x0000000F |
| D\_TESTFAILEDBIT\_CNT\_B8 | N/A | Counts | 0x01 |
| D\_TESTFAILEDTHISOPCYCLEBIT\_CNT\_B8 | N/A | Counts | 0x02 |
| D\_PENDINGDTCBIT\_CNT\_B8 | N/A | Counts | 0x04 |
| D\_CONFIRMEDDTCBIT\_CNT\_B8 | N/A | Counts | 0x08 |
| D\_TESTNOTCOMPLETESINCELASTCLEARBIT\_CNT\_B8 | N/A | Counts | 0x10 |
| D\_TESTFAILEDSINCELASTCLEARBIT\_CNT\_B8 | N/A | Counts | 0x20 |
| D\_TESTNOTCOMPLETETHISOPCYCLEBIT\_CNT\_B8 | N/A | Counts | 0x40 |
| D\_WARNINGINDICATORBIT\_CNT\_B8 | N/A | Counts | 0x80 |
| D\_RAMPNONE\_CNT\_U8 | 1 | Counts | 0x0F |
| D\_RAMPF2\_CNT\_U8 | 1 | Counts | 0x0E |
| D\_RAMPF1\_CNT\_U8 | 1 | Counts | 0x0D |
| D\_EVTFAILBITS\_CNT\_B8 | N/A | Counts | D\_TESTFAILEDBIT\_CNT\_B8 | D\_TESTFAILEDTHISOPCYCLEBIT\_CNT\_B8 | D\_CONFIRMEDDTCBIT\_CNT\_B8 |
| D\_EVTPASSBITS\_CNT\_B8 | N/A | Counts | D\_TESTFAILEDBIT\_CNT\_B8 | D\_TESTNOTCOMPLETETHISOPCYCLEBIT\_CNT\_B8 |
| D\_RAMP0PCT\_ULS\_F32 | N/A | Counts | 0.0 |
| D\_IGNOREBITSFORSTORAGE\_CNT\_B8 | N/A | Counts | D\_TESTNOTCOMPLETETHISOPCYCLEBIT\_CNT\_B8 |
| D\_DIAGRMPRTLOLMT\_ULSPMS\_F32 | Single precision float | Uls/mS | 0.0001 |
| D\_DIAGRMPRTHILMT\_ULSPMS\_F32 | Single precision float | Uls/mS | 0.5 |
| D\_DIAGSTSNONRECRMPTOZEROBIT\_CNT\_B16 | N/A | Counts | 0x0001 |
| D\_DIAGSTSCTRLDDISRMPBIT\_CNT\_B16 | N/A | Counts | 0x0002 |
| D\_DIAGSTSRECRMPTOZEROBIT\_CNT\_B16 | N/A | Counts | 0x0004 |
| D\_DIAGSTSHWASBSYSTMFLTBIT\_CNT\_B16 | N/A | Counts | 0x0008 |
| D\_DIAGSTSDEFVEHSPDBIT\_CNT\_B16 | N/A | Counts | 0x0010 |
| D\_DIAGSTSDEFTEMPBIT\_CNT\_B16 | N/A | Counts | 0x0020 |
| D\_DIAGSTSSCOMHWANOTVALIDBIT\_CNT | N/A | Counts | 0x0040 |
| D\_DIAGSTSWIRDISABLEBIT\_CNT\_B16 | N/A | Counts | 0x0080 |

#### Global

|  |
| --- |
| Constant Name |
| <None> |

### Module specific Lookup Tables Constants

|  |  |  |  |
| --- | --- | --- | --- |
| Constant Name | Resolution | Value | Software Segment |
|  |  |  |  |

# 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. TableSize\_m()

## Data Hiding Functions

1. <None>

## Global Functions/Macros Defined by this Module

### Retrieve NTC Failure State

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Function Name** | NxtrDiagMgr\_GetNTCFailed | Type | Min | Max |
| **Arguments Passed** | NTC | NTCNumber | 0 | 511 |
|  | EventFailed | Boolean pointer | FULL | FULL |
| **Return Value** | (anonymous) | Std\_ReturnType | RTE\_E\_OK | |

#### Description



### Retrieve NTC Status

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Function Name** | NxtrDiagMgr\_GetNTCStatus | Type | Min | Max |
| **Arguments Passed** | NTC | NTCNumber | 0 | 511 |
|  | Status | uint8 pointer | FULL | FULL |
| **Return Value** | (anonymous) | Std\_ReturnType | RTE\_E\_OK | |

#### Description



### Reset Event Status

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Function Name** | NxtrDiagMgr\_ResetEventStatus | Type | Min | Max |
| **Arguments Passed** | None |  |  |  |
| **Return Value** | (anonymous) | Std\_ReturnType | RTE\_E\_OK | |

#### Description



### Set Status

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Function Name** | NxtrDiagMgr\_SetNTCStatus | Type | Min | Max |
| **Arguments Passed** | NTC | NTCNumber | 0 | 511 |
|  | Param | UInt8 | 0 | 255 |
|  | Status | NxtrDiagMgrStatus | NTC\_STATUS\_PASSED  NTC\_STATUS\_FAILED  NTC\_STATUS\_PREPASSED  NTC\_STATUS\_PREFAILED | |
| **Return Value** | (anonymous) | Std\_ReturnType | RTE\_E\_OK | |

#### Description









### Set Operation State

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Function Name** | NxtrDiagMgr\_SetOperationCycleState | Type | Min | Max |
| **Arguments Passed** | OperationCycle | NxtrOpCycle | FULL | FULL |
|  | CycleState | NxtrOpCycleState | FULL | FULL |
| **Return Value** | (anonymous) | Std\_ReturnType | Std\_ReturnType | |

#### Description



### Report NTC Status

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Function Name** | NxtrDiagMgr\_ReportNTCStatus | Type | Min | Max |
| **Arguments Passed** | NTC | NTCNumber | 0 | 511 |
|  | Param | uint8 | 0 | 255 |
|  | Status | NxtrDiagMgrStatus | NTC\_STATUS\_PASSED  NTC\_STATUS\_FAILED  NTC\_STATUS\_PREPASSED  NTC\_STATUS\_PREFAILED | |
| **Return Value** | RetVal | Std\_ReturnType | E\_OK  E\_NOT\_OK | |

#### Design Rationale

This function is designed to queue up diagnostic faults that occur prior to the diagnostic manager being initialized. The logic is structured in such a way that there is a limitation that it will only capture the last status being passed to the function for a given NTC if the NTC is called multiple times prior to initialization. Parameter bits are “OR’d” together for a given NTC. The function will return E\_NOT\_OK if the queue is full.

#### Description



## Local Functions/Macros Used by this MDD only

### Set Bits

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Function Name** | SetBits\_u8 | Type | Min | Max |
| **Arguments Passed** | Data | const uint8 pointer | 0 | 255 |
|  | BitMask | uint8 | 0 | 255 |
| **Return Value** | N/A |  |  |  |

#### Description

\*Data |= BitMask

### Clear Bits

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Function Name** | ClrBits\_u8 | Type | Min | Max |
| **Arguments Passed** | Data | const uint8 pointer | 0 | 255 |
|  | BitMask | uint8 | 0 | 255 |
| **Return Value** | N/A |  |  |  |

#### Description

\*Data &= ~BitMask

### Read Bit

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Function Name** | ReadBit\_u8 | Type | Min | Max |
| **Arguments Passed** | Data | uint8 | 0 | 255 |
|  | BitMask | uint8 | 0 | 255 |
| **Return Value** | (anonymous) | boolean | FULL | FULL |

#### Description

IF 0 = (Data & BitMask) THEN

return FALSE

ELSE

return TRUE

ENDIF

### Set Bits

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Function Name** | SetBits\_u16 | Type | Min | Max |
| **Arguments Passed** | Data | const uint16 pointer | FULL | FULL |
|  | BitMask | uint16 | FULL | FULL |
| **Return Value** | N/A |  |  |  |

#### Description

\*Data |= BitMask

### Clear Bits

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Function Name** | ClrBits\_u16 | Type | Min | Max |
| **Arguments Passed** | Data | const uint16 pointer | FULL | FULL |
|  | BitMask | uint16 | FULL | FULL |
| **Return Value** | N/A |  |  |  |

#### Description

\*Data &= ~BitMask

### Read Bit

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Function Name** | ReadBit\_u32 | Type | Min | Max |
| **Arguments Passed** | Data | uint32 | 0x000000 | 0xFFFFFF |
|  | BitMask | uint32 | 0x000000 | 0xFFFFFF |
| **Return Value** | (anonymous) | boolean | FULL | FULL |

#### Description

IF 0 = (Data & BitMask) THEN

return FALSE

ELSE

return TRUE

ENDIF

### Construct the Storage Array

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Function Name** | CreateStorageArray | Type | Min | Max |
| **Arguments Passed** | OutputStrgPtr | const NTCStrgArray pointer | FULL | FULL |
| **Return Value** | N/A |  |  |  |

#### Description





### Local Function #n

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Function Name** | (Exact name used) | Type | Min | Max |
| **Arguments Passed** | (if none, write None) |  |  |  |
|  | (Insert more rows for additional passed arguments) |  |  |  |
| **Return Value** | (if no value returned, write N/A) |  |  |  |

#### Description

(Place flowchart/design for local function)

# Software Module Implementation

## Runtime Environment (RTE) Initial Values

|  |  |
| --- | --- |
| Data | Value |
| DiagRmpToZeroActive\_Cnt\_lgc | FALSE |
| DiagStsDefTemp\_Cnt\_lgc | FALSE |
| DiagStsHWASbSystmFltPres\_Cnt\_lgc | FALSE |
| DiagStsCtrldDisRmpPres\_Cnt\_lgc | FALSE |
| DiagStsDefVehSpd\_Cnt\_lgc | FALSE |
| DiagRampRate\_XpmS\_f32 | 0 |
| DiagRampValue\_Uls\_f32 | 0 |
| DiagStsNonRecRmpToZeroFltPres\_Cnt\_lgc | FALSE |
| DiagStsRecRmpToZeroFltPres\_Cnt\_lgc | FALSE |
| DiagStsScomHWANotValid\_Cnt\_lgc | FALSE |
| DiagStsWIRDisable\_Cnt\_lgc | FALSE |
| MEC\_Cnt\_enum | ProductionMode |
| MfgDiagInhibit\_Cnt\_lgc | FALSE |

## Initialization Functions

### Init: \_Init1

#### Design Rationale

None

#### Module Internals and Outputs



## Periodic Functions

### Per: \_Per1

#### Design Rationale

In this periodic information is collected about all NTCs mapped to given a DTC and the combined information is used to determine whether to update the DTC’s status information:

* DTCs are updated to FAILED if any associated NTC is currently failed regardless of whether all NTCs have been tested.
* DTCs are updated to PASSED if all associated NTCs have been tested and have passed; if any associated NTC has not been tested, the status of the DTC is not updated to PASSED.
* DTCs are not updated to FAILED if an applicable mask is active.

#### Program Flow Start

N/A

#### Store Module Inputs to Local copies

#### Set Appropriate Bits











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

None

#### Program Flow End

N/A

### Per: \_Trns1

#### Design Rationale

None

#### Program Flow Start

N/A

#### Store Module Inputs to Local copies

None

#### Construct the Storage Array

CreateStorageArray(Rte\_Pim\_NTCStrgArray())

Rte\_Call\_NTCStrgArray\_SetRamBlockStatus(TRUE)

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

None

#### Program Flow End

N/A

### Per: \_Trns2

#### Design Rationale

#### Program Flow Start

N/A

#### Store Module Inputs to Local copies

None

#### Restart the DEM Operation Cycle



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

None

#### Program Flow End

N/A

## Fault Recovery Functions

None

## Shutdown Functions

None

## Interrupt Functions

None

## Fault Recovery Functions

None

## Shutdown Functions

None

## Interrupt Functions

None

## Serial Communication Functions

### SComm: DiagMgr\_SCom\_ReadStrgArray

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | | Type | Min | Max |
| **Arguments Passed** | NTCStrgArrayPtr | NTCStrgArray pointer | FULL | FULL |
| **Return Value** | N/A |  |  |  |

#### Design Rationale

None

#### Program Flow Start

N/A

#### Store Module Inputs to Local copies

None

#### Construct the Storage Array

CreateStorageArray(NTCStrgArrayPtr)

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

None

#### Program Flow End

N/A

### SComm: DiagMgr\_SCom\_GetNTCParamByte

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | | Type | Min | Max |
| **Arguments Passed** | NTC | NTCNumber | 0 | 511 |
|  | NTCParamByte | uint8 pointer | FULL | FULL |
| **Return Value** | N/A |  |  |  |

#### Design Rationale

None

#### Program Flow Start

N/A

#### Store Module Inputs to Local copies

None

#### Construct the Storage Array

\*NTCParamByte = NTCInfo\_Cnt\_M\_str[NTC].Param

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

None

#### Program Flow End

N/A

# Execution Requirements

## Execution Sequence of the Module

DiagMgr\_Trns1 should run prior to the NVMWriteAll function call that occurs during the shutdown process.

## 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 |
| \_Init1 | Once | COLD\_INIT |
| \_Per1 | 10 ms | ALL |
| \_Trns1 | Triggered | On OFF mode entry |
| \_Trns2 | Triggered | On OFF mode exit |

## Execution Requirements for Serial Communication Functions

|  |  |
| --- | --- |
| Function Name | Sub-Module called by (Serial Comm Function Name) |
| DiagMgr\_SCom\_ReadStrgArray | EPS\_DiagSrvcs |
| DiagMgr\_SCom\_GetNTCParamByte | EPS\_DiagSrvcs |

# 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 |
| \_Init1 |  |
| \_Per1 |  |
| \_Trns1 |  |
| \_Trns2 |  |

## Global and Local Functions

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

|  |  |
| --- | --- |
| Name of Sub Module | Software Segment |
| NxtrDiagMgr\_GetNTCFailed |  |
| NxtrDiagMgr\_GetNTCStatus |  |
| NxtrDiagMgr\_ResetEventStatus |  |
| NxtrDiagMgr\_SetNTCStatus |  |
| NxtrDiagMgr\_SetOperationCycleState |  |
| SetBits\_u8 |  |
| ClrBits\_u8 |  |
| ReadBit\_u8 |  |
| SetBits\_u16 |  |
| ClrBits\_u16 |  |
| ReadBit\_u32 |  |
| CreateStorageArray |  |

# Known Issues / Limitations With Design

1. (Item #1)

# Revision Control Log

|  |  |  |  |
| --- | --- | --- | --- |
| **Rev #** | **Change Description** | **Date** | **Author Initials** |
| 1 | Initial MDD version | 30-Mar-11 | LW |
| 3 | Corrected typo in local function CreateStorageArray. | 04-Apr-11 | YY |
| 4 | Ensure that test-complete bit is accommodated. | 05-Apr-11 | YY |
| 5 | Correct strategy for selecting DTCs to retain in storage-array. | 05-Apr-11 | YY |
| 6 | States and Modes based changes for combining F1 and F2 outputs. | 11-Apr-11 | LWW |
| 7 | Removed Fault masking and replaced Dem direct Event Status access with DemIf Event Status call. | 7-Jun-11 | JJW |
| 8 | Updated for anomaly 3258. Removed unused local functions. | 17-Jun-11 | LWW |
| 9 | Added \_Trns2 to restart DEM operation cycle on OFF mode exit (anomaly 2342) CR#4270 | 17-Jun-11 | BDO |
| 10 | Initial component implementation | 13-Jan-12 | LWW |
| 11 | Changed NTCInfo\_T to NTCInfo\_Str for Data Dictionary compatibility | 27-Jan-12 | OT |
| 15 | Updates to support bit 5, 6 from DTC outline Structure Version 2. Changed the version to 15 to align with the version in synergy | 17-Apr-12 | VK |
| 16 | Updates to mask all faults when in OFF state | 22-Apr-12 | VK |
| 17 | Updates to inihibit warminit bit in all states except operate | 01-May-12 | VK |
| 18 | Anomaly 3313 (MaxRampRate overindexing) | 22-May-12 | OT |
| 19 | Update for new fault response bits | 20-Jun-12 | LWW |
| 20 | Added fault queue to handle faults being called prior to diagnostic manager initialization | 20-Jul-12 | LWW |