# pointsModule --

# High-Level Description

This module is responsible for performing the qualification of the Wheel Speed inputs to the Wheel Imbalance Rejection algorithm.

# Figures

WIRInputQual.tif

## Diagram – Function Data Sharing

This diagram shows all data that is shared between functions within the module.

No Shared Data

### Diagram – Function (Name)

This diagram describes the functional characteristics and data flow of a given function.

# 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 | |
| SrlComLWhlSpdVld\_Cnt\_lgc | | QualWhlFreqL\_Hz\_f32 |
| SrlComRWhlSpdVld\_Cnt\_lgc | | QualWhlFreqR\_Hz\_f32 |
| SrlComLWhlSpd\_Hz\_f32 | | WhlFreqQualified\_Cnt\_lgc |
| SrlComRWhlSpd\_Hz\_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 |
| PrevQualWhlSpdLeft\_Hz\_M\_f32 | single precision float | 0 | 40 | CLEARED\_32 |
| PrevQualWhlSpdRight\_Hz\_M\_f32 | single precision float | 0 | 40 | CLEARED\_32 |
| QualLevelLeft\_Cnt\_M\_u16 | 1 | 0 | 10 | CLEARED\_16 |
| QualLevelRight\_Cnt\_M\_u16 | 1 | 0 | 10 | CLEARED\_16 |
| QualErrAccLeft\_Cnt\_M\_u16 | 1 | FULL | FULL | CLEARED\_16 |
| QualErrAccRight\_Cnt\_M\_u16 | 1 | FULL | FULL | 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 |  |  |  |  |
|  |  |  |  |  |

# 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\_WhlSpdQPStep\_Cnt\_u16 |
| k\_WhlSpdQLimit\_Cnt\_u16 |
| k\_WhlSpdQNStep\_Cnt\_u16 |
| t\_FreqScaleTblX\_Hz\_u7p9 |
| k\_WhlSpdQualDiag\_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\_WHLSPDMIN\_HZ\_F32 | single precision float | Hz | 0 |
| D\_WHLSPDMAX\_HZ\_F32 | single precision float | Hz | 40 |

#### 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 |
| <None> |
|  |

### 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. Min\_m
3. FPM\_FloatToFixed\_m
4. DiagPStep\_m
5. DiagNStep\_m
6. DiagFailed\_m

## Data Hiding Functions

1. <None>

## Global Functions/Macros Defined by this Module

### Global Function #1

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Function Name** | None | Type | Min | Max | UTP Tol. |
| **Arguments Passed** |  |  |  |  |  |
|  |  |  |  |  |  |
| **Return Value** |  |  |  |  |  |

#### Description

(Place flowchart/design for local function)

## Local Functions/Macros Used by this MDD only

### Qualify Wheel Speed

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Function Name** | QualifyWhlSpd | Type | Min | Max | UTP Tol. |
| **Arguments Passed** | WhlSpd\_Ptr\_T\_f32 | pointer to float32 | 0 | 40 |  |
|  | PrevQualWhlSpd\_Ptr\_T\_f32 | pointer to float32 | 0 | 40 |  |
|  | WhlSpdValid\_Cnt\_T\_lgc | boolean | FALSE | TRUE |  |
|  | QualLevel\_Ptr\_T\_u16 | pointer to uint16 | 0 | 10 |  |
| **Return Value** | N/A |  |  |  |  |

#### Description



### Wheel Speed In Range Check

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Function Name** | WhlSpdInRange | Type | Min | Max | UTP Tol. |
| **Arguments Passed** | WhlSpd\_Hz\_T\_f32 | float32 | 0 | 40 |  |
| **Return Value** | InRange\_Cnt\_T\_lgc | boolean | FALSE | TRUE | 0 |

#### Description



### Wheel Speed Qualification Check

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Function Name** | WhlSpdQualCheck | Type | Min | Max | UTP Tol. |
| **Arguments Passed** | QualLevel\_Cnt\_T\_u16 | uint16 | 1 | 10 |  |
|  | InRange\_Cnt\_T\_lgc | boolean | FALSE | TRUE |  |
|  | QualErrAcc\_Ptr\_T\_u16 | pointer to uint16 | FULL | FULL |  |
| **Return Value** | WhlSpdQualfied\_Cnt\_T\_lgc | Boolean | FALSE | TRUE | 0 |

#### 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 |
| SrlComLWhlSpdVld\_Cnt\_lgc | FALSE |
| SrlComRWhlSpdVld\_Cnt\_lgc | FALSE |
| SrlComLWhlSpd\_Hz\_f32 | 0 |
| SrlComRWhlSpd\_Hz\_f32 | 0 |
| QualWhlFreqL\_Hz\_f32 | 0 |
| QualWhlFreqR\_Hz\_f32 | 0 |
| WhlFreqQualified\_Cnt\_lgc | TRUE |

## Initialization Functions

None

## Periodic Functions

### Per: \_Per1

#### Design Rationale

None

#### Program Flow Start

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

#### Store Module Inputs to Local copies

WhlSpdLeftValid\_Cnt\_T\_lgc = Rte\_IRead\_WIRInputQual\_Per1\_SrlComLWhlSpdVld\_Cnt\_lgc()

WhlSpdLeft\_Hz\_T\_f32 = Rte\_IRead\_WIRInputQual\_Per1\_SrlComLWhlSpd\_Hz\_f32()

WhlSpdRightValid\_Cnt\_T\_lgc = Rte\_IRead\_WIRInputQual\_Per1\_SrlComRWhlSpdVld\_Cnt\_lgc()

WhlSpdRight\_Hz\_T\_f32 = Rte\_IRead\_WIRInputQual\_Per1\_SrlComRWhlSpd\_Hz\_f32()

#### Processing



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

Rte\_IWrite\_WIRInputQual\_Per1\_QualWhlFreqL\_Hz\_f32(WhlSpdLeft\_Hz\_T\_f32)

Rte\_IWrite\_WIRInputQual\_Per1\_QualWhlFreqR\_Hz\_f32(WhlSpdRight\_Hz\_T\_f32)

Rte\_IWrite\_WIRInputQual\_Per1\_WhlFreqQualified\_Cnt\_lgc(WhlFreqQualified\_Cnt\_T\_lgc)

#### Program Flow End

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

## Fault Recovery Functions

None

## Shutdown Functions

None

## Interrupt Functions

None

## Serial Communication Functions

None

# Execution Requirements

## Execution Sequence of the Module

(Describe in words relevant details about the execution sequence of the different sub modules.)

## 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 |
| WIRInputQual\_Per1 | 2ms | 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 |
| WIRInputQual\_Per1 | RTE\_START\_SEC\_AP\_WIRINPUTQUAL\_APPL\_CODE |
|  |  |

## Local Functions

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

|  |  |
| --- | --- |
| Name of Sub Module | Software Segment |
| QualifyWhlSpd | N/A (Inline with calling function) |
| WhlSpdInRange | N/A (Inline with calling function) |
| WhlSpdQualCheck | N/A (Inline with calling function) |

# Known Issues / Limitations With Design

1. Inline function defined in globalmacro.h are not unit tested

# Revision Control Log

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Item #** | **Rev #** | **Change Description** | **Date** | **Author Initials** |
| 1 | 1 | Initial version | 29-July-13 | LWW |