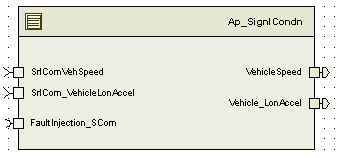
# Module – Signal Conditioning

# High-Level Description

This function conditions a signal received from SER prior to its distribution to other functions. Typical conditioning methods may include filters, slew rates, gain values or limits.

# Figures

## Diagram – Function Data Sharing



# 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.

(Note: Full variable names required in table.)

(Note: All global variables including End Of Line data used should be shown here)

|  |  |
| --- | --- |
| Module Inputs (Global Variable Name) | Module Outputs (Global Variable Name) |
| SrlComVehSpd\_Kph\_f32 | VehSpd\_Kph\_f32 |
| SrlCom\_VehicleLonAccel\_KphpS\_f32 | Vehicle\_LonAccel\_KphpS\_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 |
| CurrSrlComVehSpd\_Kph\_M\_f32 | Single precision floating point | 0 | 350 | SIGNLCONDN\_START\_SEC\_VAR\_NOINIT\_32 |
| CurrSrlComVehLonAccel\_KphpS\_M\_f32 | Single precision floating point | -180 | 180 | SIGNLCONDN\_START\_SEC\_VAR\_NOINIT\_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) |

# 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\_VehSpdSlewRate\_KphpSec\_f32 |
| k\_VehAccelSlewRate\_KphpSecSq\_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 | Value |
| D\_VEHLONACCELGAIN\_KPHPS\_F32 | N/A | 3.6 |

#### 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\_2MS\_SEC\_F32 |
| BC\_SIGNLCONDN\_FAULTINJECTIONPOINT |
| FLTINJ\_SRLCOMVEHSPD\_SGNLCOND |
| FLTINJ\_SRLCOMVEHLONACCEL\_SGNLCOND |

### 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 |  |  |  |

## Lookup Table Definitions

# Software Module Implementation

## Initialization Functions

None

## Periodic Functions

### Per: SignlCondn \_Per1

#### Design Rationale

NOTE: For “starttime” calculations there is tendency for underflow and this is expected in s/w design. So for unittesting, VBA model should be implemented

such that it handles underflow and behaves like source code design.

#### Program Flow Start

#### Rte\_Call\_SignlCondn\_Per1\_CP0\_CheckpointReached()Store Module Inputs to Local copies

SrlComVehSpd\_Kph\_T\_f32 = RteRte\_IRead\_SignlCondn\_Per1\_SrlComVehSpeed\_Kph\_f32

SrlComVehLonAccel\_KphpS\_T\_f32 = Rte\_IRead\_SignlCondn\_Per1\_SrlCom\_VehicleLonAccel\_KphpS\_f32()

#### Signal Conditioning



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

Rte\_Iwrite\_SignlCondn\_Per1\_VehicleSpeed\_Kph\_f32 (CurrSrlComVehSpd\_Kph\_M\_f32)

Rte\_IWrite\_SignlCondn\_Per1\_Vehicle\_LonAccel\_KphpS\_f32(CurrSrlComVehLonAccel\_KphpS\_M\_f32)

#### Program Flow End

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

## Fault Recovery Functions

None

## Shutdown Functions

None

## Interrupt Functions

None

## Serial Communication Functions

# 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 |
| SignlCondn\_Per1 | 2 ms | ALL States |
|  |  |  |

## 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 |
| SignlCondn\_Per1 |  |
|  |  |

## 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

# Revision Control Log

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Item #** | **Rev #** | **Change Description** | **Date** | **Author Initials** |
| 1 | 1.0 | Initial release | 14-May-12 | NRAR |
| 2 | 2.0 | Updated as per FDDVer002, FaultInjectionPoint added to SrlComVehSpeed signal | 20-Aug-12 | NRAR |
| 3 | 3.0 | Corrected Fault Injection function call | 05-Sep-12 | NRAR |
| 4 | 4.0 | Added checkpoints and memmap software segment is updated for static variables | 25-Sep-12 | Selva |
| 5 | 5.0 | Updated to SF-33 ver 003 | 23-May-13 | SP |