# Module – State Output Control

# High-Level Description

This function performs the ramp up and ramp down of the Torque Command.

# 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) |
| ~~TrqCmd\_MtrNm\_f32~~ | ~~FinalTrqCmd\_MtrNm\_f32~~ |
| ~~SrlComSvcDft\_Cnt\_b32~~ | OutputRampMult\_Uls\_f32 |
| DiagRampRate\_XpmS\_32 | RampDwnStatusComplete\_Cnt\_lgc |
| DiagRampValue\_Uls\_f32 |  |
| OperRampRate\_XpmS\_f32 |  |
| OperRampValue\_Uls\_f32 |  |
| RampSrlComSvcDft\_Cnt\_lgc |  |
| DiagStsDiagRmpActive\_Cnt\_lgc |  |

## 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 |
| ~~AttenFactor\_Uls\_M\_f32~~ | Single precision floating point | 1.175494351e-038 | 3.402823466e+038 |  |
| ~~ActvRampUsr\_Cnt\_M\_u16~~ | 1 | 0 | 16 |  |
| PrevOutputRampMult\_Uls\_M\_f32 | Single precision floating point | 0 | 1 | STOPCTRL\_START\_SEC\_VAR\_NOINIT\_32 |
| PrevTargetRampMult\_Uls\_M\_f32 | Single precision floating point | 0 | 1 | STOPCTRL\_START\_SEC\_VAR\_NOINIT\_32 |
| PrevRate\_XpmS\_M\_f32 | Single precision floating point | 0.0001 | 0.5 | STOPCTRL\_START\_SEC\_VAR\_NOINIT\_32 |
| RampState\_M\_Str | RampState\_T | See 3.1.1 | See 3.1.1 | STOPCTRL\_START\_SEC\_VAR\_NOINIT\_UNSPECIFIED |

### 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) |
| RampState\_T | StartTime\_mS\_u32  Duration\_mS\_u32  StartVal\_Uls\_f32  EndVal\_Uls\_f32 | uint32  uint32  float32  float32 | 0  0  0  0 | 2^32-1  2^32-1  1  1 |

# 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 |
|  |

## 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\_TWO\_MS\_U32 | 1 | 2 |
| D\_MAXRAMP\_XPMS\_F32 | Single precision floating point | 0.5 |

#### 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 |
|  |

### 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: \_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\_StOpCtrl\_Per1\_CP0\_CheckpointReached()Store Module Inputs to Local copies

OperRampRate\_XpmS\_T\_f32 as float32

OperRampValue\_Uls\_T\_f32 as float32

DiagRampValue\_Uls\_T\_f32 as float32

DiagRampRate\_XpmS\_T\_f32 as float32

DiagStsDiagRmpActive\_Cnt\_T\_lgc as Boolean

RampSrlComSvcDft\_Cnt\_T\_lgc as Boolean

Rate\_T\_f32 as float32

Target\_T\_f32 as float32

DiffOutputRampMult\_T\_f32 as float32

DiffRate\_T\_f32 as float32

OperRampRate\_XpmS\_T\_f32 = Rte\_IRead\_StOpCtrl\_Per1\_OperRampRate\_XpmS\_f32

OperRampValue\_Uls\_T\_f32 =Rte\_Iread\_StOpCtrl\_Per1\_OperRampValue\_Uls\_f32

DiagRampValue\_Uls\_T\_f32=Rte\_Iread\_StOpCtrl\_Per1\_DiagRampValue\_Uls\_f32

DiagRampRate\_XpmS\_T\_f32=Rte\_Iread\_StOpCtrl\_Per1\_DiagRampRate\_XpmS\_f32

DiagStsDiagRmpActive\_Cnt\_T\_lgc = Rte\_Iread\_StOpCtrl\_Per1\_DiagStsDiagRmpActive\_Cnt\_lgc

RampSrlComSvcDft\_Cnt\_T\_lgc= Rte\_Iread\_StOpCtrl\_Per1\_RampSrlComSvcDft\_Cnt\_lgc



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

Rte\_Iwrite\_StOpCtrl\_Per1\_RampDwnStatusComplete\_Cnt\_lgc(RampDwnStatusComplete\_T\_lgc)

Rte\_Iwrite\_StOpCtrl\_Per1\_OutputRampMult\_Uls\_f32(NewOutputRampMult\_T\_f32)

#### Program Flow End

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

## Fault Recovery Functions

None

## Shutdown Functions

None

## Interrupt Functions

None

## Serial Communication Functions

None

## Local Function/Macro Definitions

### RampLib

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Function Name** | RampLib | Type | Min | Max |
| **Arguments Passed** | rampState\_T\_Str | RampState\_T | \* | \* |
|  |  |  |  |  |
| **Return Value** | Output\_Uls\_T\_f32 | Float |  |  |

#### \*Note: For ranges on structure elements check Table 3.1.1 of MDD

#### Description

Rte\_Call\_SystemTime\_DtrmnElapsedTime\_mS\_u32(rampState\_T\_Str.StartTime\_mS\_u32, &ElapsedRamp\_mS\_T\_u32)



# 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 |
| StOpCtrl\_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 |
| StOpCtrl\_Per1() | RTE\_START\_SEC\_AP\_STOPCTRL\_APPL\_CODE RTE\_STOP\_SEC\_AP\_STOPCTRL\_APPL\_CODE |
|  |  |

## Local Functions

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

|  |  |
| --- | --- |
| Name of Sub Module | Software Segment |
| RampLib | RTE\_START\_SEC\_AP\_STOPCTRL\_APPL\_CODE  RTE\_STOP\_SEC\_AP\_STOPCTRL\_APPL\_CODE |

# Known Issues / Limitations With Design

1. (Item #1)

# Revision Control Log

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Item #** | **Rev #** | **Change Description** | **Date** | **Author Initials** |
| 1 | 1.0 | Initial release | 07-Jun-11 | SAH |
| 2 | 2.0 | FDD SF05 | 5-Jan-12 | NRAR |
| 3 | 3.0 | Value for D\_MAXRAMP\_XPMS\_F32 is fixed | 6-Jan-12 | NRAR |
| 4 | 4.0 | DiagStsF1Active\_Cnt\_lgc is renamed to DiagStsDiagRmpActive\_Cnt\_lgc | 12-Jan-12 | NRAR |
| 5 | 4.0 | PrevRate\_XpmS\_M\_f32 range correction | 23-Jan-12 | NRAR |
| 6 | 5.0 | Anom #3272 Ramp output vs. target fix | 13-Aug-12 | BWL |
| 7 | 6.0 | Added checkpoints and memmap software segment is updated for static variables | 23-Sep-12 | Selva |