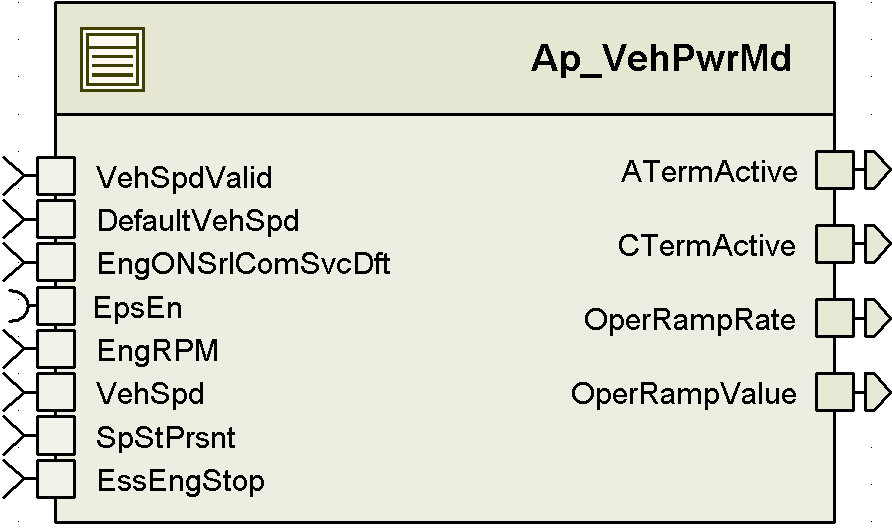
# Module -- VehiclePwrMode

# High-Level Description

This module determines the vehicle level power moding signal for the EPS system

# Figures



## Diagram – Function Data Sharing

### Diagram – Function (Name)

# 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 | |
| EngRPM\_Cnt\_u16 | | ATermActive\_Cnt\_lgc |
| EngONSrlComSvcDft\_Cnt\_lgc | | CTermActive\_Cnt\_lgc |
| VehSpd\_Kph\_f32 | | OperRampRate\_XpmS\_f32 |
| VehSpdValid\_Cnt\_lgc | | OperRampValue\_Uls\_f32 |
| EssEngStop\_Cnt\_lgc | |  |
| SpStPrsnt\_Cnt\_T\_lgc | |  |
| DefaultVehSpd | |  |

## 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 |
| CTermActive\_Cnt\_M\_lgc | 1 | FALSE | TRUE | VEHPWRMD\_START\_SEC\_VAR\_CLEARED\_BOOLEAN |
|  |  |  |  |  |
|  |  |  |  |  |

### 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\_RampDnRt\_UlspmS\_f32 |
| k\_RampUpRtLoSpd\_UlspmS\_f32 |
| k\_VehSpdAstLmt\_Kph\_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 | Units | Value |
| D\_ENGONRPM\_CNT\_U16 | 1 | RPM | 300 |
| D\_ENGRPMINVLD\_CNT\_U16 | 1 | Counts | 0xFFFF |
|  |  |  |  |

#### 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\_ZERO\_ULS\_F32 |
| D\_ONE\_ULS\_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. <None>

## Data Hiding Functions

1. Rte\_Call\_EpsEn\_OP\_GET

## Global Functions/Macros Defined by this Module

### Global Function #1

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Function Name** | **N/A** | 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)

## Local Functions/Macros Used by this MDD only

### Local Function #1

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Function Name** | **N/A** | 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

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 |
| ATermActive\_Cnt\_lgc | TRUE |
| CTermActive\_Cnt\_lgc | FALSE |
| EngRPM\_Cnt\_u16 | 0 |
| EngONSrlComSvcDft\_Cnt\_lgc | FALSE |
| VehSpd\_Kph\_f32 | 0 |
| VehSpdValid\_Cnt\_lgc | FALSE |
| OperRampRate\_XpmS\_f32 | 0 |
| OperRampValue\_Uls\_f21 | 0 |

## Initialization Functions

### Init: \_Init1

#### Design Rationale

#### Module Outputs

Rte\_IWrite\_VehPwrMd\_Init1\_OperRampRate\_XpmS\_f32(k\_RampDnRt\_UlspmS\_f32)

Rte\_IWrite\_VehPwrMd\_Init1\_OperRampValue\_Uls\_f32(D\_ZERO\_ULS\_F32)

#### Module Internal

(Initialize all module internal variables in this section)

## Periodic Functions

### Per: \_Per1

#### Design Rationale

There are a set of inputs that are read in but not used. This is because these inputs are likely to be used in the final design. The current implementation does not meet final design requirements.

#### Program Flow Start

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

*EngONSrlComSvcDft\_Cnt\_T\_lgc* = Rte\_IRead\_VehPwrMd\_Per1\_EngONSrlComSvcDft\_Cnt\_lgc()

Rte\_Call\_EpsEn\_OP\_GET(&*EPSEn\_Cnt\_T\_lgc*)

*EngRPM\_Cnt\_T\_u16 = Rte\_IRead\_VehPwrMd\_Per1\_EngRPM\_Cnt\_u16()*

*VehSpdValid\_Cnt\_T\_lgc* = Rte\_IRead\_VehPwrMd\_Per1\_VehSpdValid\_Cnt\_lgc()

*VehSpd\_Kph\_T\_f32 =* Rte\_IRead\_VehPwrMd\_Per1\_VehSpd\_Kph\_f32()

*EssEngStop\_Cnt\_T\_lgc = Rte\_IRead\_VehPwrMd\_Per1\_EssEngStop\_Cnt\_lgc()*

*SpStPrsnt\_Cnt\_T\_lgc = Rte\_IRead\_VehPwrMd\_Per1\_SpStPrsnt\_Cnt\_lgc()*

*NmState\_Cnt\_T\_u8 = NmGetStatus()*

#### Processing



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

Rte\_IWrite\_VehPwrMd\_Per1\_OperRampRate\_XpmS\_f32(*OperRampRate\_XpmS\_T\_f32*)

Rte\_IWrite\_VehPwrMd\_Per1\_OperRampValue\_Uls\_f32(*OperRampValue\_Uls\_T\_f32*)

Rte\_IWrite\_VehPwrMd\_Per1\_ATermActive\_Cnt\_lgc(*ATermActive\_Cnt\_T\_lgc*)

Rte\_IWrite\_VehPwrMd\_Per1\_CTermActive\_Cnt\_lgc(*CTermActive\_Cnt\_M\_lgc*)

#### Program Flow End

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

## Fault Recovery Functions

## Shutdown Functions

## Interrupt Functions

## Serial Communication Functions

# 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 |
| VehPwrMd\_Init1 | Executed once after the RTE is started | COLDINIT |
| VehPwrMd\_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 |
| VehPwrMd\_Init1 |  |
| VehPwrMd\_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

|  |  |  |  |
| --- | --- | --- | --- |
| **Rev #** | **Change Description** | **Date** | **Author Initials** |
| 1.0 | Initial MDD | 23May11 | LWW |
| 2.0 | Initial Chrysler version for VehPwrMode | 31JUL12 | SAH |
| 3.0 | Update per SER Rev 2B | 3JAN13 | BWL |
| 4.0 | Updated per SER Rev 2G and anomaly 4761 | 12Apr13 | M. Story |
| 5.0 | Add TWaitBusSleep check for shutdown | 21Jun13 | M. Story |
| 6.0 | Anomaly 5586 NetWork management changes | 02OCT13 | M. Story |
| 7.0 | Anomaly 5911 NM changes | 31OCT13 | M. Story |
| 9 | Updated for L version 02.00.00 and WR 01.0.04 | 20NOV13 | M. Story |
| 10 | Anomaly 6131 Cterm changes | 05DEC13 | M. Story |
| 11 | Anomaly 6206 | 10JAN14 | M. Story |