|  |  |
| --- | --- |
| **NVP**  **Non-Volatile Parameters** | |
| **Summary** | This is the Software Detailed Design Document for the *DAIMLER MMA* Project. |

|  |  |  |
| --- | --- | --- |
| **Author** | **Review** | **Approval** |
| Title: Stefan Dominte | See Project Master Document for the roles and Project Members List for the name of people | See Project Master Document for the roles and Project Members List for the name of people |
|  |  |  |
| **Distribution** | | |
| See Project Master Document for the roles and Project Members List for the name of people | See Project Master Document for the roles and Project Members List for the name of people | See Project Master Document for the roles and Project Members List for the name of people |

# Table of content

1.1. Revision history \* 3

1.2. Purpose and Scope 3

1.3. Referenced documents 3

1.3.1. External documents 3

1.3.2. Internal Documents 3

1.4. Terminology and definitions 3

2.1. Overview 4

2.2. Traceability 4

3.1. Services 5

3.1.1. nvp\_CheckIfBitSet 5

3.1.2. nvp\_CheckVehicleEquipmentStatus 6

3.1.3. nvp\_ComputeRawVoltageValues 7

3.1.4. nvp\_CopyVehicleEquipmentDefaultData 8

3.1.5. nvp\_CopyVehicleEquipmentEOLData 9

3.1.6. NVP\_Init 10

3.2. Variabiles 10

3.2.1. NVP\_stVehicleEquipmentData 10

3.3. Types 11

3.3.1. nvp\_stVehicleEquipmentData 11

3.4. Macros 11

3.4.1. KU16\_NVP\_OV\_MAX\_VALUE 11

3.4.2. KU16\_NVP\_OV\_MIN\_VALUE 12

3.4.3. KU16\_NVP\_UV\_MAX\_VALUE 12

3.4.4. KU16\_NVP\_UV\_MIN\_VALUE 12

3.4.5. KU32\_NVP\_VOLTAGE\_DIV\_FACTOR 12

3.4.6. KU32\_NVP\_VOLTAGE\_MUL\_FACTOR 12

3.4.7. KU8\_NVP\_READ\_DATA\_DEVELOPMENT 12

3.4.8. KU8\_NVP\_READ\_DATA\_VEHICLE\_EQUIPMENT 12

# Tables of Figures:

[Figure 1: nvp\_CheckIfBitSet function 6](#_Toc134446116)

[Figure 2: nvp\_CheckVehicleEquipmentStatus function Part1 7](#_Toc134446117)

[Figure 3: nvp\_CheckVehicleEquipmentStatus function Part2 7](#_Toc134446118)

[Figure 4: nvp\_ComputeRawVoltageValues function 8](#_Toc134446119)

[Figure 5: nvp\_CopyVehicleEquipmentDefaultData function 9](#_Toc134446120)

[Figure 6: nvp\_CopyVehicleEquipmentEOLData function 10](#_Toc134446121)

[Figure 7: NVP\_Init function 10](#_Toc134446122)

# General Information

## Revision history \*

|  |  |  |  |
| --- | --- | --- | --- |
| **Revision** | **Date** | **Author(s)** | **Description/comment** |
| 1.1.4.2 | 06/02/2023 | Stefan Dominte | Initial revision |
| 1.1.4.3 | 08/02/2023 | Stefan Dominte | Update with correct revision |
| 1.1.4.4 | 13/02/2023 | Stefan Dominte | Update after review |
| 1.1.4.5 | 03/03/2023 | Stefan Dominte | Update for R5.1 |
| 1.1.4.6 | 08/05/2023 | Stefan Dominte | Update for R6.0 |
|  |  |  |  |

*\* Template history is found in the CM tool used for templates*

## Purpose and Scope

The purpose of this document is to provide an overview of the NVP operation principle, and to present the implementation choices in terms of module and function splitting.

## Referenced documents

### External documents

|  |  |  |
| --- | --- | --- |
| **Id** | **Title** | **Reference** |
|  |  |  |
|  |  |  |
|  |  |  |

### Internal Documents

|  |  |  |
| --- | --- | --- |
| **Id** | **Title** | **Reference** |
|  | DAIMLER\_MMA\_SWarchitectureDesignInterfaceDescription.docx |  |
|  | NVP - Design Interface Description.docx |  |
|  | SBE\_4G\_NVP\_layout.xls |  |
|  |  |  |

## Terminology and definitions

The generic acronyms are available in the [AEM process and method wiki](https://alvteams.alv.autoliv.int/sites/aeuaeequalityassurance/AEM%20Process%20wiki/acronyms.aspx)

|  |  |
| --- | --- |
| **Terminology** | **Meaning** |
| AAU | Atomic architectural unit |
| SW | software |
|  |  |

# SW Module Detailed Design

## Overview

NVP (Non-Volatile Parameters) is a very small SW unit intended to compute and save locally the vehicle equipment configuration data and vehicle equipment development data. If NVM vehicle equipment configuration block does not contain valid data default parameters will be used instead. NVM vehicle equipment development block will always be used.

## Traceability

|  |  |  |  |
| --- | --- | --- | --- |
| Requirements | Criteria | Linked Runnable | Source |
| DSG\_NVP\_0001 | If Vehicle Equipment Configuration Data **has no valid** data the local function nvp\_CopyVehicleEquipmentDefaultData will set the Vehicle equipment block with NVP default data. | NVP\_Init() | ARCH\_SW\_NVP\_0022 |
| DSG\_NVP\_0002 | The local variable NVP\_stVehicleEquipmentData will hold the NVP default data if Vehicle Equipment Configuration Data **has no valid** data. | NVP\_Init() | ARCH\_SW\_NVP\_0022 |
| DSG\_NVP\_0003 | If Vehicle Equipment Configuration Data **has valid** data the local function nvp\_CopyVehicleEquipmentEOLData will be called in order to save locally the equipment data. | NVP\_Init() | ARCH\_SW\_NVP\_0022 |
| DSG\_NVP\_0004 | Local function nvp\_CheckVehicleEquipmentStatus will be used in order to check if vehicle equipment configuration **has valid** data. | NVP\_Init() | ARCH\_SW\_NVP\_0023 |
| DSG\_NVP\_0005 | If Vehicle Equipment Configuration Data **has valid** data the local function nvp\_ComputeRawVoltageValues will compute the raw voltage to physical voltage value. | NVP\_Init() | ARCH\_SW\_NVP\_0023 |
| DSG\_NVP\_0006 | If Vehicle Equipment Configuration Data **has valid** data the local function nvp\_CheckIfBitSet will be used in order to save locally a specific vehicle equipment bit. | NVP\_Init() | ARCH\_SW\_NVP\_0023 |

# Features

## Services

### nvp\_CheckIfBitSet

|  |  |  |  |
| --- | --- | --- | --- |
| Object | | | |
| Function that checks if a bit is set. | | | |
| **Prototype** | | | |
| LOCAL uint8 nvp\_CheckIfBitSet (uint8 value, uint8 bitNumber) | | | |
| **Parameters** | | | |
| value - value to be checked bitNumber - bit number to be checked | | | |
| **Exceptions** | | | |
| None | | | |
| **Precondition** | | | |
| None | | | |
| **Postcondition** | | | |
| None | | | |
| **Input parameters** | | | |
| Name | Type | Description | Range |
| bitNumber | uint8 | bit number to be checked | NA |
| **Output parameters** | | | |
| Name | Type | Description | Range |
| NA | NA | NA | NA |
| **Return value** | | | |
| Type | Description | | |
| uint8 | ONE - if bit is set and ZERO - bit not set | | |
| **Dynamic aspect** | | | |
| Who(callers) | Description | | |
| nvp\_CheckVehicleEquipmentStatus | \* | | |
| **Static aspect** | | | |
| \* | | | |
| **Constrains** | | | |
|  | | | |

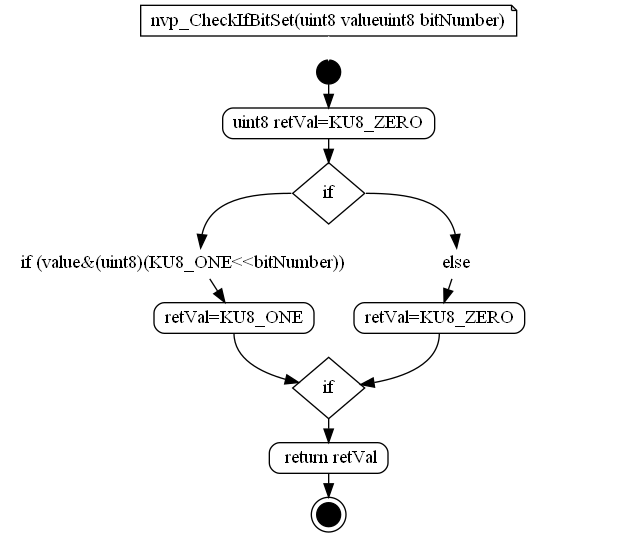


Figure : nvp\_CheckIfBitSet function

### nvp\_CheckVehicleEquipmentStatus

|  |  |  |  |
| --- | --- | --- | --- |
| Object | | | |
| Function that checks if the SSA vehicle equipment block was written before. | | | |
| **Prototype** | | | |
| LOCAL uint8 nvp\_CheckVehicleEquipmentStatus (void) | | | |
| **Parameters** | | | |
| None | | | |
| **Exceptions** | | | |
| None | | | |
| **Precondition** | | | |
| None | | | |
| **Postcondition** | | | |
| None | | | |
| **Input parameters** | | | |
| Name | Type | Description | Range |
| NA | NA | NA | NA |
| **Output parameters** | | | |
| Name | Type | Description | Range |
| NA | NA | NA | NA |
| **Return value** | | | |
| Type | Description | | |
| uint8 | ONE - if block was written ZERO - block was not written | | |
| **Dynamic aspect** | | | |
| Who(callers) | Description | | |
| NVP\_Init | \* | | |
| **Static aspect** | | | |
| \* | | | |
| **Constrains** | | | |
|  | | | |

Note: Diagram is too large and was splitted in two

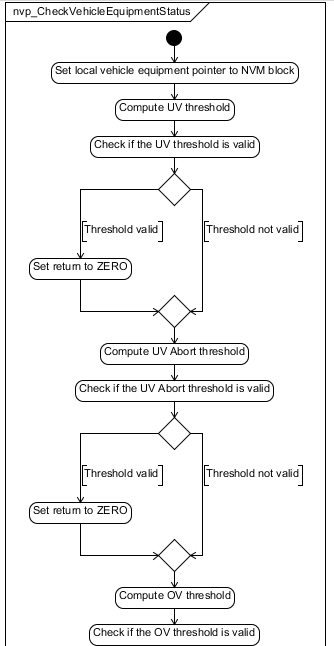


Figure : nvp\_CheckVehicleEquipmentStatus function Part1

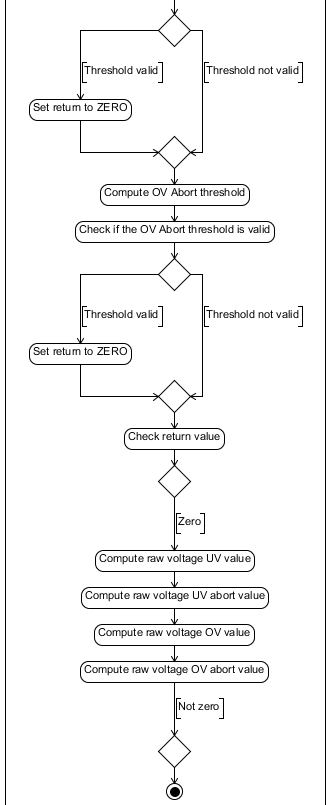


Figure : nvp\_CheckVehicleEquipmentStatus function Part2

### nvp\_ComputeRawVoltageValues

|  |  |  |  |
| --- | --- | --- | --- |
| Object | | | |
| Function that computes the raw voltage values into physical values. | | | |
| **Prototype** | | | |
| LOCAL uint16 nvp\_ComputeRawVoltageValues (uint16 u16VoltageValue) | | | |
| **Parameters** | | | |
| u16VoltageValue - raw voltage value | | | |
| **Exceptions** | | | |
| None | | | |
| **Precondition** | | | |
| None | | | |
| **Postcondition** | | | |
| None | | | |
| **Input parameters** | | | |
| Name | Type | Description | Range |
| u16VoltageValue | Uint16 | raw voltage value | NA |
| **Output parameters** | | | |
| Name | Type | Description | Range |
| NA | NA | NA | NA |
| **Return value** | | | |
| Type | Description | | |
| uint16 | Computed voltage value | | |
| **Dynamic aspect** | | | |
| Who(callers) | Description | | |
| nvp\_CheckVehicleEquipmentStatus | \* | | |
| **Static aspect** | | | |
| \* | | | |
| **Constrains** | | | |
|  | | | |

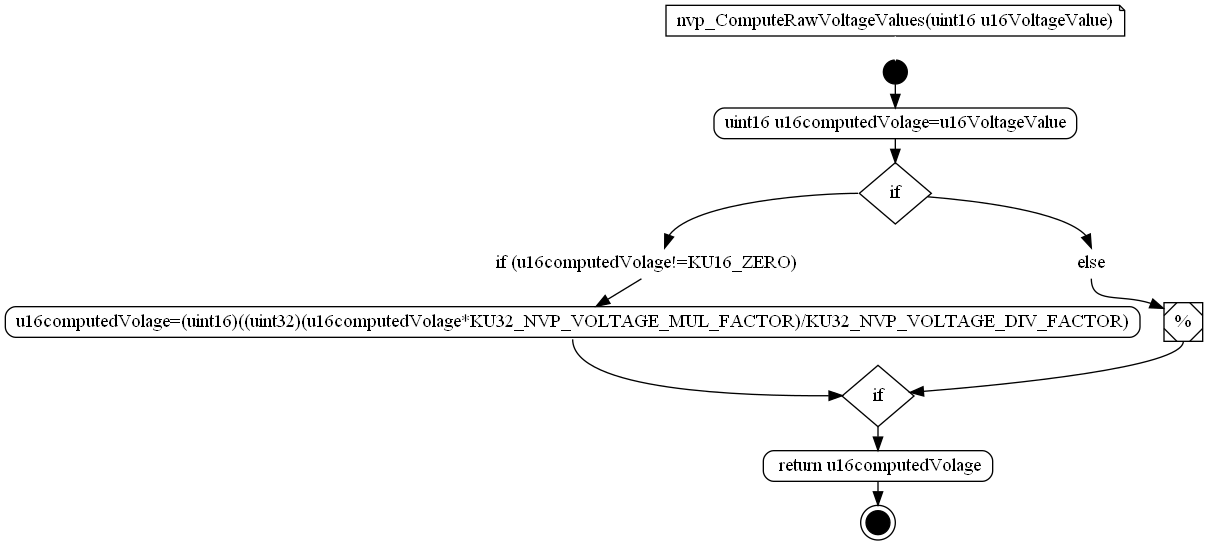


Figure : nvp\_ComputeRawVoltageValues function

### nvp\_CopyVehicleEquipmentDefaultData

|  |  |  |  |
| --- | --- | --- | --- |
| Object | | | |
| Function will handle the CALIB Vehicle equipment data to local structure. | | | |
| **Prototype** | | | |
| LOCAL void nvp\_CopyVehicleEquipmentDefaultData (uint8 u8ReadDataTarget) | | | |
| **Parameters** | | | |
| u8ReadDataTarget - target to read data from Vehicle equipment or Development | | | |
| **Exceptions** | | | |
| None | | | |
| **Precondition** | | | |
| None | | | |
| **Postcondition** | | | |
| None | | | |
| **Input parameters** | | | |
| Name | Type | Description | Range |
| NA | NA | NA | NA |
| **Output parameters** | | | |
| Name | Type | Description | Range |
| NA | NA | NA | NA |
| **Return value** | | | |
| Type | Description | | |
| NA | None | | |
| **Dynamic aspect** | | | |
| Who(callers) | Description | | |
| NVP\_Init | \* | | |
| **Static aspect** | | | |
| \* | | | |
| **Constrains** | | | |
|  | | | |



Figure : nvp\_CopyVehicleEquipmentDefaultData function

### nvp\_CopyVehicleEquipmentEOLData

|  |  |  |  |
| --- | --- | --- | --- |
| Object | | | |
| Function will handle the Ssa Coding Vehicle equipment data to local structure. | | | |
| **Prototype** | | | |
| LOCAL void nvp\_CopyVehicleEquipmentEOLData (uint8 u8ReadDataTarget) | | | |
| **Parameters** | | | |
| u8ReadDataTarget - target to read data from Vehicle equipment or Development | | | |
| **Exceptions** | | | |
| None | | | |
| **Precondition** | | | |
| None | | | |
| **Postcondition** | | | |
| None | | | |
| **Input parameters** | | | |
| Name | Type | Description | Range |
| u8ReadDataTarget | uint8 | u8ReadDataTarget - target to read data from Vehicle equipment or Development | NA |
| **Output parameters** | | | |
| Name | Type | Description | Range |
| NA | NA | NA | NA |
| **Return value** | | | |
| Type | Description | | |
| NA | None | | |
| **Dynamic aspect** | | | |
| Who(callers) | Description | | |
| NVP\_Init | \* | | |
| **Static aspect** | | | |
| \* | | | |
| **Constrains** | | | |
|  | | | |



Figure : nvp\_CopyVehicleEquipmentEOLData function

### NVP\_Init

|  |  |  |  |
| --- | --- | --- | --- |
| Object | | | |
| NVP init function will be called periodicaly to check if vehicle equipment configuration data was written. | | | |
| **Prototype** | | | |
| void NVP\_Init (void) | | | |
| **Parameters** | | | |
| None | | | |
| **Exceptions** | | | |
| None | | | |
| **Precondition** | | | |
| None | | | |
| **Postcondition** | | | |
| None | | | |
| **Input parameters** | | | |
| Name | Type | Description | Range |
| NA | NA | NA | NA |
| **Output parameters** | | | |
| Name | Type | Description | Range |
| NA | NA | NA | NA |
| **Return value** | | | |
| Type | Description | | |
| NA | None | | |
| **Dynamic aspect** | | | |
| Who(callers) | Description | | |
| Ct\_MBBL\_SsaAppl\_Coding\_MainFunction | Every 10ms | | |
| **Static aspect** | | | |
| \* | | | |
| **Constrains** | | | |
|  | | | |



Figure : NVP\_Init function

### **Variables**

#### NVP\_stVehicleEquipmentData

|  |  |  |
| --- | --- | --- |
| Type | Value |  |
| nvp\_stVehicleEquipmentData | NA | |
| **Description** | | |
| Variable that holdes the vehicle equipment data. | | |
| **Definition** | | |
| nvp\_stVehicleEquipmentData NVP\_stVehicleEquipmentData | | |
| **Remarks** | | |
| NA | | |

## Types

### nvp\_stVehicleEquipmentData

|  |  |  |
| --- | --- | --- |
| **Field Name** | **Field Type** | **Field description** |
| VehicleSteeringConfig | uint8 : 1 | To store the VehicleSteeringConfig bit |
| BSRActivation | uint8 : 1 | To store the BSRActivation bit |
| BltHandOverAvailabiliy | uint8 : 1 | To store the BltHandOverAvailabiliy bit |
| Reserved1 | uint8 : 1 | To store the Reserved1 bit |
| HWALeftActivation | uint8 : 1 | To store the HWALeftActivation bit |
| HWARightActivation | uint8 : 1 | To store the HWARightActivation bit |
| APIActivation | uint8 : 1 | To store the APIActivation bit |
| Reserved2 | uint8 : 1 | To store the Reserved2 bit |
| PreSafeCycleActivation | uint32 | To store the PreSafeCycleActivation bytes |
| ORCPreSafe8Activation | uint8 : 1 | To store the ORCPreSafe8Activation bit |
| StartUpActivation | uint8 : 1 | To store the StartUpActivation bit |
| Reserved3 | uint8 : 3 | To store the Reserved3 bits |
| ReleaseOutOfMemory | uint8 : 1 | To store the ReleaseOutOfMemory bit |
| BSROutOfMemory | uint8 : 1 | To store the BSROutOfMemory bit |
| PreSafeOutOfMemory | uint8 : 1 | To store the PreSafeOutOfMemory bit |
| Reserved4 | uint8 : 8 | To store the Reserved4 byte |
| Reserved5 | uint8 : 8 | To store the Reserved5 byte |
| UndervoltageAbort | uint16 | To store the UndervoltageAbort bytes |
| Undervoltage | uint16 | To store the Undervoltage bytes |
| OvervoltageAbort | uint16 | To store the OvervoltageAbort bytes |
| Overvoltage | uint16 | To store the Overvoltage bytes |
| MeasFrame1Activation | uint8 : 1 | To store the MeasFrame1Activation bit |
| MeasFrame2Activation | uint8 : 1 | To store the MeasFrame2Activation bit |
| MeasFrame3Activation | uint8 : 1 | To store the MeasFrame3Activation bit |
| MeasFrame4Activation | uint8 : 1 | To store the MeasFrame4Activation bit |
| MeasFrame5Activation | uint8 : 1 | To store the MeasFrame5Activation bit |
| MeasFrame6Activation | uint8 : 1 | To store the MeasFrame6Activation bit |
| MeasFrame7Activation | uint8 : 1 | To store the MeasFrame7Activation bit |
| MeasFrame8Activation | uint8 : 1 | To store the MeasFrame8Activation bit |
| PreSafeFuncActivation | uint8 : 1 | To store the PreSafeFuncActivation bit |
| DisplayMsgActivation | uint8 : 1 | To store the DisplayMsgActivation bit |
| CounterLimitActivation | uint8 : 1 | To store the CounterLimitActivation bit |
| CRCFaultActivation | uint8 : 1 | To store the CRCFaultActivation bit |
| Reserved6 | uint8 : 4 | To store the Reserved6 bits |

## Macros

### KU16\_NVP\_OV\_MAX\_VALUE

|  |  |
| --- | --- |
| Name | Value |
| KU16\_NVP\_OV\_MAX\_VALUE | ((uint16)3674) |
| **Definition** | |
| #define KU16\_NVP\_OV\_MAX\_VALUE ((uint16)3674) | |
| **Description** | |
| Overvoltage max value. | |

### KU16\_NVP\_OV\_MIN\_VALUE

|  |  |
| --- | --- |
| Name | Value |
| KU16\_NVP\_OV\_MIN\_VALUE | ((uint16)3266) |
| **Definition** | |
| #define KU16\_NVP\_OV\_MIN\_VALUE ((uint16)3266) | |
| **Description** | |
| Overvoltage minim value. | |

### KU16\_NVP\_UV\_MAX\_VALUE

|  |  |
| --- | --- |
| Name | Value |
| KU16\_NVP\_UV\_MAX\_VALUE | ((uint16)2041) |
| **Definition** | |
| #define KU16\_NVP\_UV\_MAX\_VALUE ((uint16)2041) | |
| **Description** | |
| Undervoltage max value. | |

### KU16\_NVP\_UV\_MIN\_VALUE

|  |  |
| --- | --- |
| Name | Value |
| KU16\_NVP\_UV\_MIN\_VALUE | ((uint16)1531) |
| **Definition** | |
| #define KU16\_NVP\_UV\_MIN\_VALUE ((uint16)1531) | |
| **Description** | |
| Undervoltage minim value. | |

### KU32\_NVP\_VOLTAGE\_DIV\_FACTOR

|  |  |
| --- | --- |
| Name | Value |
| KU32\_NVP\_VOLTAGE\_DIV\_FACTOR | ((uint32)10) |
| **Definition** | |
| #define KU32\_NVP\_VOLTAGE\_DIV\_FACTOR ((uint32)10) | |
| **Description** | |
| Voltage divide factor. | |

### KU32\_NVP\_VOLTAGE\_MUL\_FACTOR

|  |  |
| --- | --- |
| Name | Value |
| KU32\_NVP\_VOLTAGE\_MUL\_FACTOR | ((uint32)49) |
| **Definition** | |
| #define KU32\_NVP\_VOLTAGE\_MUL\_FACTOR ((uint32)49) | |
| **Description** | |
| Voltage multiply factor. | |

### KU8\_NVP\_READ\_DATA\_DEVELOPMENT

|  |  |
| --- | --- |
| Name | Value |
| KU8\_NVP\_READ\_DATA\_DEVELOPMENT | ((uint8)0x0001) |
| **Definition** | |
| #define KU8\_NVP\_READ\_DATA\_DEVELOPMENT ((uint8)0x0001) | |
| **Description** | |
| Vehicle development ID. | |

### KU8\_NVP\_READ\_DATA\_VEHICLE\_EQUIPMENT

|  |  |
| --- | --- |
| Name | Value |
| KU8\_NVP\_READ\_DATA\_VEHICLE\_EQUIPMENT | ((uint8)0x0000) |
| **Definition** | |
| #define KU8\_NVP\_READ\_DATA\_VEHICLE\_EQUIPMENT ((uint8)0x0000) | |
| **Description** | |
| Vehicle equipment ID. | |

# EEPROM

The EEPROM parameters are all specified in [Doc3 = SBE\_4G\_NVP\_layout.xls].

Refer to this document for more details.

# Configuration

No special configuration for NVP software component.

# Compilation Options

No compilation options for NVP software component.