Vehicle Mode Management

Technical Reference

VMM

Version 1.06.00

Authors	Thomas Kuhl, Markus Schwarz
Status	Released

1 Document Information

1.1 History

Author	Date	Version	Remarks
Thomas Petrus	2008-06-02	1.0	Initial Version
Thomas Kuhl	2008-10-17	1.1	Update configuration chapter
Thomas Kuhl	2008-11-28	1.2	Update configuration chapter Updated Chapter 7.4 Add function description: Vmm_BusSm_EnableReceptionDM Add function description: Vmm_BusSm_DisableReceptionDM
Thomas Kuhl	2009-03-16	1.3	Add chapter 4.3 ECU Passive Handling Add API Vmm_Dcm_SetPassiveMode Update chapter “System configuration”
Thomas Kuhl	2009-08-10	1.4	add chapter 5.4 Critical code sections
Thomas Kuhl	2009-11-16	1.5	ESCAN00038948
Thomas Kuhl	2010-04-29	1.05.01	ESCAN00040934
Thomas Kuhl	2010-08-10	1.05.02	ESCAN00044654
Thomas Kuhl	2011-02-10	1.06.00	Extend description of Vmm_Init

Table 1-1 History of the Document

1.2 Reference Documents

No.	Title	Version
[1]	AUTOSAR_BasicSoftwareModules.pdf	V1.0.0
[2]	AUTOSAR_SWS_DET.pdf	V2.2.0
[3]	AN-ISC-8-1118 MICROSAR BSW Compatibility Check	V1.0.0

Table 1-2 Reference Documents

⚠	**Please note** We have configured the programs in accordance with your specifications in the questionnaire. Whereas the programs do support other configurations than the one specified in your questionnaire, Vector's release of the programs delivered to your company is expressly restricted to the configuration you have specified in the questionnaire.

Contents

Illustrations

Tables

2 Component History

The component history gives an overview over the important milestones that are supported in the different versions of the component.

Component Version	New Features
1.00.00	Initial Version

Table 2-1 Component history

3 Introduction

This document describes the Vehicle Mode Management (VMM) of the common software components. It describes the features, the API, integration hints and the configuration.

Supported Configuration Variants:	pre-compile, link-time	
Vendor ID:	VMM_VENDOR_ID	30 decimal (= Vector-Informatik, according to HIS)
Module ID:	VMM_MODULE_ID	226 decimal (according to ref[1])

4 Functional Description

The VMM is responsible to

- allow/inhibit I-PDU start/stop requests from BusSM
- start/stop its own I-PDU groups

depending on the communication status that is set by the diagnostic component (DCM) via the diagnostic service called “communication control”.

Additionally the Vmm is responsible to provide the ECU passive mode to the configured BusSM.

4.1 Handling of communication requests from BusSM

The VMM handles the I-PDU start/stop requests from the BusSM:

- If no VMM is used on a channel, each I-PDU start/stop request is directly forwarded to the Com.
- If a VMM is used and Tx/Rx is not inhibited by DCM request, the I-PDU start/stop request is directly forwarded to the Com.
- If a VMM is used and Tx/Rx is inhibited by DCM request, the I-PDU start/stop request is stored within the VMM.

4.2 Control of communication status by DCM

The DCM uses API Vmm_Dcm_CommunicationControl() to set the communication status for a given (or all) networks.

The communication status can be set for Nm- and/or Com-related messages.

The Tx and Rx behavior can be controlled separately.

4.2.1 Nm Handling

If Vmm_Dcm_CommunicationControl() addresses the Nm (by parameter VMM_MSG_TYPE_NM or VMM_MSG_TYPE_ALL), the Nm for the given channel is enabled/disabled depending on the requested Tx state.

4.2.2 Com Handling

If Vmm_Dcm_CommunicationControl() addresses the Com (by parameter VMM_MSG_TYPE_COM or VMM_MSG_TYPE_ALL), the Tx/Rx I-PDUs are started/stopped.

Handling of I-PDUs of VMM

These I-PDUs are used to enable necessary communication while normal communication itself is blocked by DCM request. The I-PDU groups can be configured for each channel of the VMM.

- If Tx/Rx gets started by DCM, the configured Tx/Rx I-PDUs of VMM are stopped.
- If Tx/Rx gets stopped by DCM, the configured Tx/Rx I-PDUs of VMM are started.

Handling of I-PDUs of BusSM

- If Tx/Rx gets started by DCM, the Tx/Rx I-PDUs of BusSM are started if they are currently requested by the BusSM.
- If Tx/Rx gets stopped by DCM, the Tx/Rx I-PDUs of BusSM are stopped if they are currently not requested by the BusSM.

4.3 Control of ECU Passive Mode

The Vmm is informed by the DCM about the ECU passive mode and the Vmm distributes this mode to the configured BusSM (FrSM and CanSM).

4.4 Error Handling

4.4.1 Development Error Reporting

By default, development errors are reported to the DET using the service `Det_ReportError()` as specified in [2], if development error reporting is enabled (i.e. pre-compile parameter `VMM_DEV_ERROR_DETECT==STD_ON`).

If another module is used for development error reporting, the function prototype for reporting the error can be configured by the integrator, but must have the same signature as the service `Det_ReportError()`.

The reported VMM ID is 226.

The reported service IDs identify the services which are described in 7. The following table presents the service IDs and the related services:

Service ID	Service
0x01	Vmm_Dcm_CommunicationControl
0x02	Vmm_BusSmIpduGroupStart
0x03	Vmm_BusSmIpduGroupStop

Table 4-1 Mapping of service IDs to services

The errors reported to DET are described in the following table:

Error Code		Description
0x10	VMM_E_UNINIT	There are VMM services used without initialization of the VMM via Vmm_Init()

Table 4-2 Errors reported to DET

5 Integration

5.1 Scope of Delivery

The delivery of the VMM contains the files which are described in the chapters 5.1.1 and 5.1.2:

5.1.1 Static Files

File Name	Description
Vmm.c	This is the source file of the VMM. It contains the implementation of the main functionality
Vmm.h	This is the header file of the VMM, which is the interface for upper layers to the services of the VMM.
Vmm_Types.h	Header File which includes VMM specific data types.
Vmm_BusSM.h	Header File for interface to BusSM.
Vmm_Dcm.h	Header File for interface to Dcm.
Vmm.lib	This is the library of the VMM. (optional)

Table 5-1 Static files

5.1.2 Dynamic Files

The dynamic files are generated by the configuration tool GENy.

File Name	Description
Vmm_Lcfg.c	This is the link time configuration source file. It contains all link time configuration settings.
Vmm_cfg.h	This is the VMM configuration header file.

Table 5-2 Generated files

5.2 Include Structure

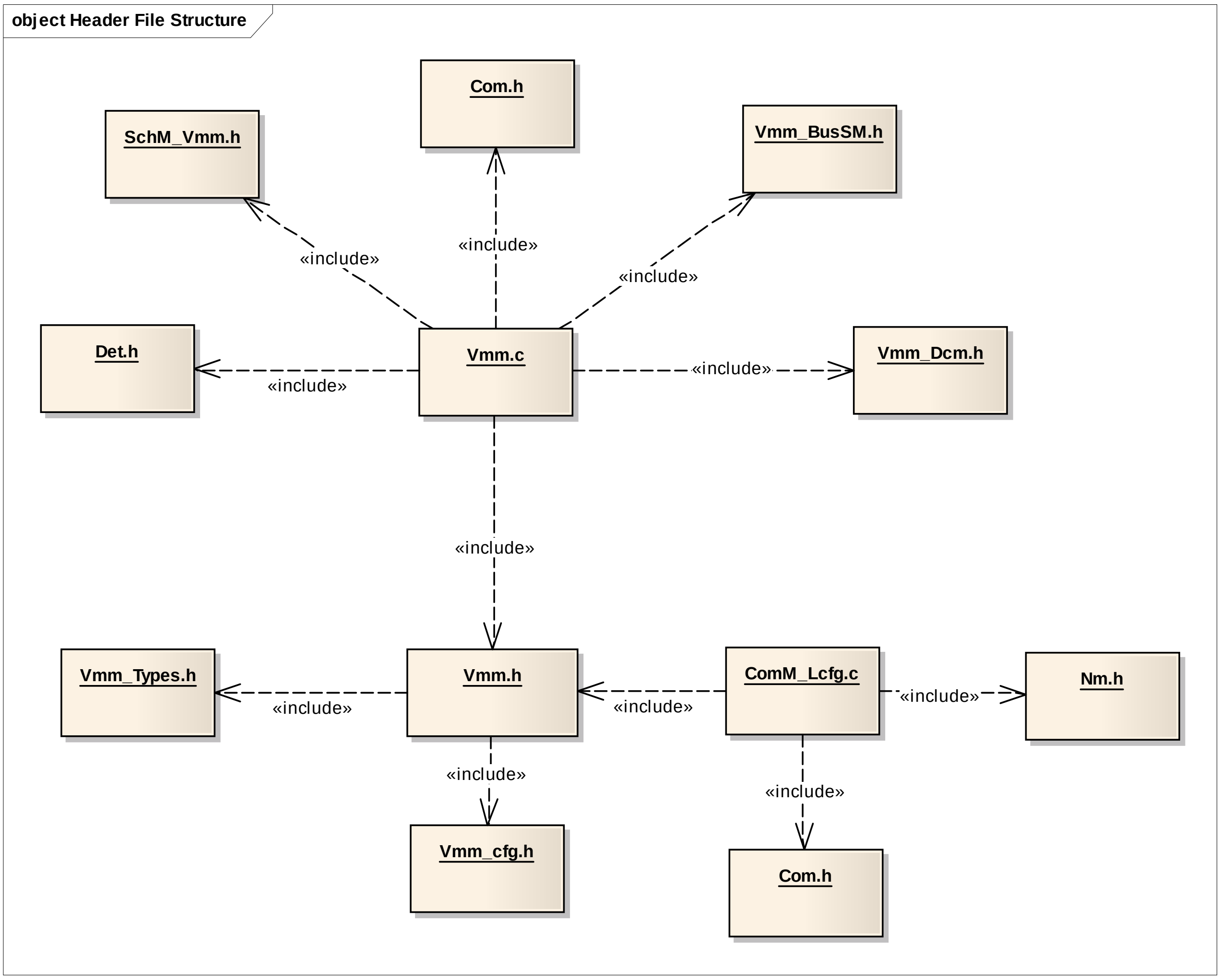


Figure 5-1 Include structure

5.3 Compiler Abstraction and Memory Mapping

The objects (e.g. variables, functions, constants) are declared by compiler independent definitions – the compiler abstraction definitions. Each compiler abstraction definition is assigned to a memory section.

The following table contains the memory section names and the compiler abstraction definitions that are used by the VMM. It illustrates their assignment among each other.

Compiler Abstraction Definitions Memory Mapping Sections	VMM_CONST	VMM_CODE	VMM_VAR	VMM_VAR_NOINIT	VMM_VAR_ZERO_INIT
VMM_START_SEC_CONST_8BIT VMM_STOP_SEC_CONST_8BIT	■				
VMM_START_SEC_CONST_32BIT VMM_STOP_SEC_CONST_32BIT	■				
VMM_START_SEC_CONST_UNSPECIFIED VMM_STOP_SEC_CONST_UNSPECIFIED	■				
VMM_START_SEC_CODE VMM_STOP_SEC_CODE		■			
VMM_START_SEC_VAR_NOINIT_8BIT VMM_STOP_SEC_VAR_NOINIT_8BIT				■	
VMM_START_SEC_VAR_ ZERO_INIT_UNSPECIFIED VMM_STOP_SEC_VAR_ ZERO_INIT_UNSPECIFIED					■

Table 5-3 Complier Abstraction and Memory Mapping

5.4 Critical code sections

The VMM has the following defined critical code section:

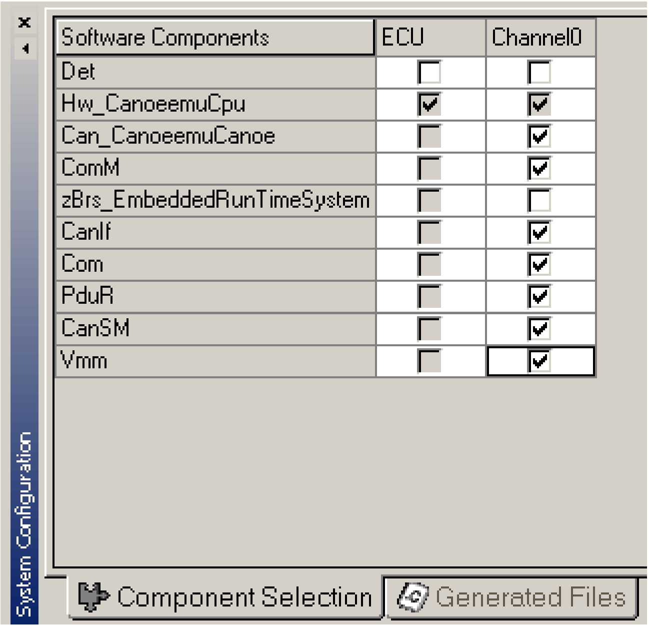
- VMM_EXCLUSIVE_AREA_0: must lock interrupts if VMM could be interrupted by any of the following task functions:
  - DCM_MainFunction()
  - CanSM_MainFunction()
  - FrSM_MainFunction()

It is recommended to use AUTOSAR OS ‘Resources’ for these exclusive areas to prevent priority inversions and dead-locks.

6 Configuration

The VMM component can be configured with the configuration tool GENy.

6.1 Activation of the VMM

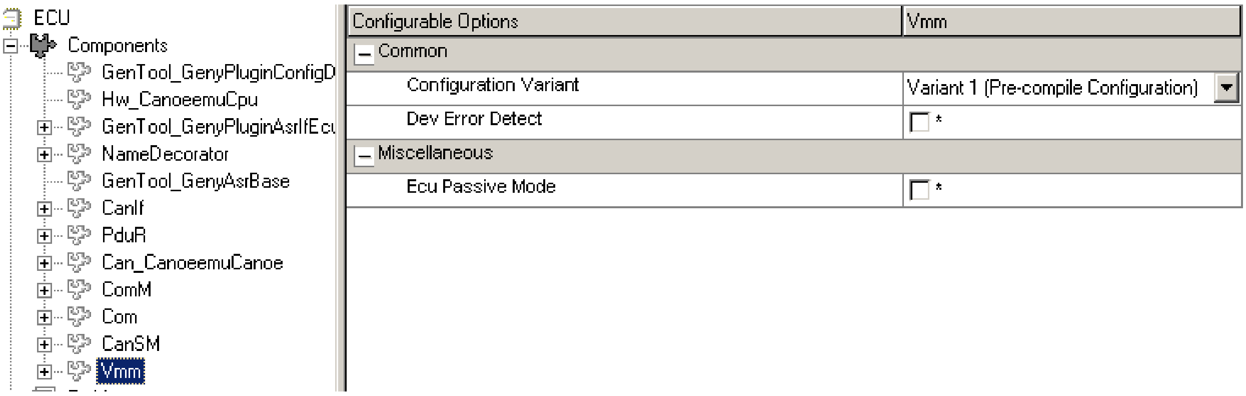


Software Components	ECU	Channel0
Det	☐	☐
Hw_CanoeemuCpu	☑	☑
Can_CanoeemuCanoe	☐	☑
ComM	☐	☑
zBrs_EmbeddedRunTimeSystem	☐	☐
CanIf	☐	☑
Com	☐	☑
PduR	☐	☑
CanSM	☐	☑
Vmm	☐	☑

Figure 6-1 VMM Activation

The VMM must be activated in the system configuration view.

6.2 System Configuration



- ECU
  - Components
    - GenTool_GenyPluginConfigD
    - Hw_CanoeemuCpu
    - GenTool_GenyPluginAsrIfEcu
    - NameDecorator
    - GenTool_GenyAsrBase
    - CanIf
    - PduR
    - Can_CanoeemuCanoe
    - ComM
    - Com
    - CanSM
    - Vmm

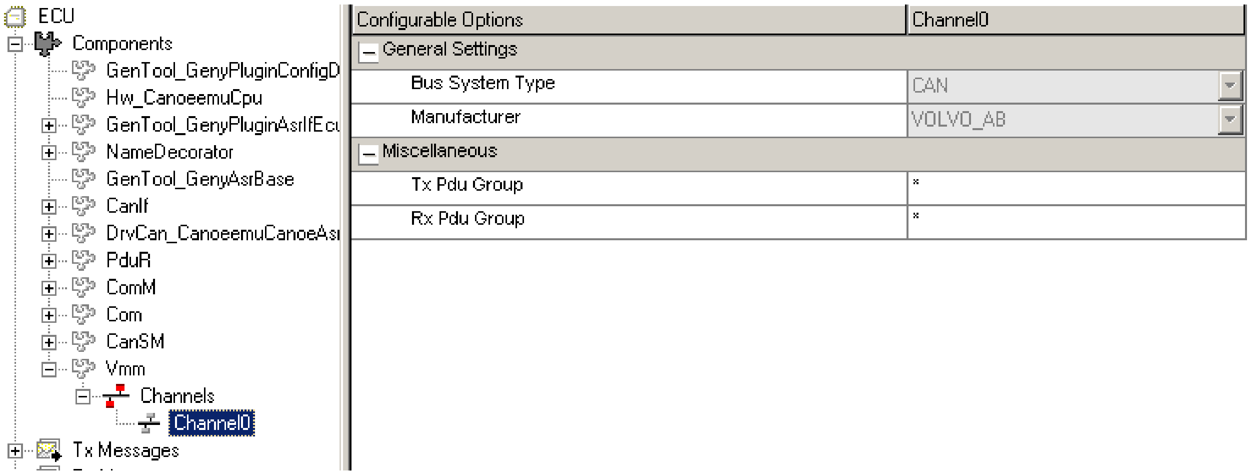
Configurable Options	Vmm
Common	
Configuration Variant	Variant 1 (Pre-compile Configuration)
Dev Error Detect	☐ *
Miscellaneous	
Ecu Passive Mode	☐ *

Figure 6-2 System Configuration

Configuration attributes	Value	Short description
General Settings		
Configuration Variant	■ Variant 1 (Pre-compile Configuration) ■ Variant 2 (Link-time Configuration)	Specify the supported configuration variant.
Dev Error Detect	■ On ■ Off	If 'Development Error Detection' is enabled, all development errors are reported to the Development Error Tracer (DET). Note: In general, the development error detection is recommended during pre-test phase. It is not recommended to enable the development error detection in production code due to increased runtime and ROM needs.
ECU Passive Mode	■ On ■ Off	Enable/Disable ECU Passive Mode Handling

Table 6-1 System Configuration

6.3 Channel Configuration



ECU
Components
GenTool_GenyPluginConfigD
Hw_CanoeemuCpu
GenTool_GenyPluginAsrIfEcu
NameDecorator
GenTool_GenyAsrBase
CanIf
DrvCan_CanoeemuCanoeAsr
PduR
ComM
Com
CanSM
Vmm
Channels
Channel0
Tx Messages

Configurable Options	Channel0
General Settings	
Bus System Type	CAN
Manufacturer	VOLVO_AB
Miscellaneous	
Tx Pdu Group	*
Rx Pdu Group	*

Figure 6-3 Channel Configuration

Configuration attributes	Value	Short description
General Settings		
Tx Pdu Group	■ Value	This value defines the Tx I-PDU group handle of Com signals which must be send during the communication control phase.
Rx Pdu Group	■ Value	This value defines the Rx I-PDU group handle of Com signals which must be received during the communication control phase.

Table 6-2 Channel Configuration

6.4 Nm Configuration

For usage of a Nm channel the following features has be enabled inside the Nm configurations:

- Nm ‘Com Control Enabled’ has to be enabled
- CanNm ‘Com Control Enabled’ has to be enabled.

7 API Description

7.1 Vmm_InitMemory

Vmm_InitMemory

Prototype	
	`void` **`Vmm_InitMemory`**`( void )`
Parameter	
-	
Return code	
-	-
Functional Description	
Pre-Initialize the VMM.	
Particularities and Limitations	
■ Must be called before Vmm_Init() during the initialization phase.	
Call context	
■ -	

Table 7-1 Vmm_InitMemory

7.2 Vmm_Init

Vmm_Init

Prototype	
“Multiple Identity Configuration” disabled: void **Vmm_Init**(void)	
“Multiple Identity Configuration” enabled: void **Vmm_Init**(const Vmm_ConfigSetType ConfigPtr)	
Parameter	
ConfigPtr	Pointer to the VMM configuration that shall be used. There is one configuration for each identity. The configurations are stored in Vmm_Lcfg.c (variables of type Vmm_ConfigSetType). Note: The pointer is only used for use case “Multiple Identity Configuration”.
Return code	
-	-
Functional Description	
Initialize the VMM. For “Multiple Identity Configurations”, the Vmm is initialized with a pointer to the configuration for the identity that shall be used. Each identity configuration contains the channels that are active in this configuration. I.e the VMM performs only actions for channels which are configured for the active identity.	
Particularities and Limitations	
■ Must be called during the initialization phase. ■ Interrupts must be disabled during initialization.	
Call context	
■ -	

Table 7-2 Vmm_Init

7.3 Vmm_Dcm_CommunicationControl

Vmm_Dcm_CommunicationControl

Prototype	
	`Std_ReturnType` **`Vmm_Dcm_CommunicationControl`** `( NetworkHandleType Channel, Vmm_MsgType msgType, boolean rxState, boolean txState )`
Parameter	
Channel	Network Handle Note: If channel is set to 0xFF, all channels are addressed.
msgType	■ VMM_MSG_TYPE_NM, only NM is affected ■ VMM_MSG_TYPE_COM, only Com is affected ■ VMM_MSG_TYPE_ALL, Nm and Com are affected
rxState	■ TRUE, enable Rx path ■ FALSE, disable Rx path
txState	■ TRUE, enable Tx path ■ FALSE, disable Tx path
Return code	
E_OK	■ API accepted
E_NOT_OK	■ VMM is not initialized
Functional Description	
This function is called from the DCM and is used to switch off/on the communication in conjunction to the given parameter.	
Particularities and Limitations	
■ -	
Call context	
■ Task and Interrupt context	

Table 7-3 Vmm_Dcm_CommunicationControl

7.4 Vmm_BusSm_IpduGroupStart

Vmm_BusSm_IpduGroupStart	
Prototype	
	`void` **`Vmm_BusSm_IpduGroupStart`** `(NetworkHandleType Channel, Com_PduGroupIdType IpduGroupId, boolean Initialize)`
Parameter	
Channel	network handle
IpduGroupId	BusSm I-PDU Group ID
Initialize	■ TRUE, start with Com default values ■ FALSE, start not with Com default values
Return code	
Functional Description	
This function is called from the bus station manager. It is used to start an I-PDU group.	
Particularities and Limitations	
■ -	
Call context	
■ Task and Interrupt context	

Table 7-4 ComM_BusSm_IpduGroupStart

7.5 Vmm_BusSm_IpduGroupStop

Vmm_BusSm_IpduGroupStop

Prototype	
	`void` **`Vmm_BusSm_IpduGroupStop`** `(NetworkHandleType Channel, Com_PduGroupIdType IpduGroupId )`
Parameter	
Channel	network handle
IpduGroupId	BusSM I-PDU Group ID
Return code	
Functional Description	
This function is called from the bus station manager. It is used to stop an I-PDU group.	
Particularities and Limitations	
■ -	
Call context	
■ Task and Interrupt context	

Table 7-5 Vmm_BusSm_IpduGroupStop

7.6 Vmm_BusSm_EnableReceptionDM

Vmm_BusSm_EnableReceptionDM	
Prototype	
	`void` **`Vmm_BusSm_EnableReceptionDM`** `(NetworkHandleType Channel, Com_PduGroupIdType IpduGroupId )`
Parameter	
Channel	network handle
IpduGroupId	BusSM I-PDU Group ID
Return code	
Functional Description	
This function is called from the bus station manager. It is used to enable the deadline monitoring for the given I-PDU group.	
Particularities and Limitations	
■ -	
Call context	
■ Task and Interrupt context	

Table 7-6 Vmm_BusSm_EnableReceptionDM

7.7 Vmm_BusSm_DisableReceptionDM

Vmm_BusSm_DisableReceptionDM

Prototype	
	void **Vmm_BusSm_DisableReceptionDM** (NetworkHandleType Channel, Com_PduGroupIdType IpduGroupId)
Parameter	
Channel	network handle
IpduGroupId	BusSM I-PDU Group ID
Return code	
Functional Description	
This function is called from the bus station manager. It is used to disable the deadline monitoring for the given I-PDU group.	
Particularities and Limitations	
■ -	
Call context	
■ Task and Interrupt context	

Table 7-7 Vmm_BusSm_DisableReceptionDM

7.8 Vmm_Dcm_SetPassiveMode

Vmm_Dcm_SetPassiveMode

Prototype	
	Std_ReturnType **Vmm_Dcm_PassiveMode**(boolean passiveState)
Parameter	
passiveState	■ TRUE, enable the ECU passive mode ■ FALSE, disable the ECU passive mode
Return code	
E_OK	■ API accepted
E_NOT_OK	■ VMM is not initialized
Functional Description	
This function is called from the DCM and is used to enable/disable the ECU passive mode inside the BusSM.	
Particularities and Limitations	
■ -	
Call context	
■ Task and Interrupt context	

Table 7-8 Vmm_Dcm_SetPassiveMode

7.9 Callback Functions

The VMM does not have any callback functions.

7.10 Service Ports

There are currently no service ports.

7.11 Services used by Vmm

In the following table services provided by other components, which are used by the Vmm are listed. For details about prototype and functionality refer to the documentation of the providing component.

Component	API
DET	Det_ReportError
Com	Com_IpduGroupStop
Com	Com_IpduGroupStart
Com	Com_EnableReceptionDM
Com	Com_DisableReceptionDM
NmIf	Nm_EnableCommunication
NmIf	Nm_DisableCommunication
NmOsek	TalkNM
NmOsek	SilentNM
EcuM	EcuM_GeneratorCompatibilityError refer to [3]

Table 7-9 Services used by the Vmm

8 Limitations

8.1 Nm Passive mode support

The Vmm does not support configurations with the enabled feature “Passive Mode” inside the AUTOSAR Nm.

9 Abbreviations

9.1 Abbreviations

Abbreviation	Description
VMM	Vehicle Mode Management
Com	BSW module in AUTOSAR providing signal based communication.
ComM	AUTOSAR Communication Manager
EcuM	AUTOSAR Ecu Manager

10 Contact

Visit our website for more information on

> News
> Products
> Demo software
> Support
> Training data
> Addresses

www.vector-informatik.com