BMW GROUP

Rolls-Royce Motor Cars Limited

Rmh Classic Integration Manual

Project	BMW AUTOSAR 4 Core Rel. 3
Author	BMW AG
Release Date	2017-12-14
Version	5.0.0
Status	Release
Hotline	+49 89 382 - 32233
Contact	bac@bmw.de https://asc.bmw.com/jira/browse/BSUP (extern) https://asc.bmwgroup.net/jira/browse/BSUP (intern)

Revision History

Version	Date	Description
5.0.0	2017-12-14	Initial version for SP2021

Company
Bayerische
Motoren Werke
Aktiengesellschaft

Postal address
BMW AG
80788 München

Office address
Forschungs- und
Innovationszentrum
(FIZ)
Hufelandstr. 1
80937 München

Telephone
Switchboard
+49 89 382-0

Internet
www.bmwgroup.com

BMW GROUP

Rolls-Royce Motor Cars Limited

Table of Contents

1 Introduction

General

For a general introduction to the BAC4 Platform Modules please refer to [1].

This document only describes topics related to the Rmh BAC4 Module.

This Integration Manual describes the basis functionality, API and the configuration of the BMW system function Rmh.

Functional overview

The main objective of the Request Message Handler (`Rmh`) functionality is to trigger the transmission of specific Com I-PDUs, when a certain (Com-)Signal is received embedded in a special so-called request message. The use case in the BMW bordnet is as follows: If certain TX-Pdus of an ECU are marked as being "requestable" in the bordnet database, the ECU has to support the reception of a specific request message. In the payload of the request message is defined, which of the ECUs configured requestable TX-Pdus shall be sent out once again. This feature is used for TX-Pdus, which are either sent out on-change/on-event only or if the cyclic frequency is very low. Other ECUs in the vehicle, which are interessted in the reception of such an TX-Pdu can then send a request message to this ECU, to get the current value instead of waiting for a long time until it is sent out again regularly.

The Rmh module itself is modeled as an AUTOSAR software component (SWC) residing above the RTE. Depending on the configuration variant an additional CD (complex driver) may be part of the Rmh module.

2 Acronyms and Abbreviations

Abbreviation Acronym:	Description:
AUTOSAR	Automotive Open Systems Architecture group
BAC	BMW AUTOSAR Core
Com	AUTOSAR Communication module
Det	AUTOSAR Default Error Tracer module
ECU	Electronical Control Unit
PDU	Protocol Data Unit
Rmh	BMW Request Message Handler system function (this module)
RTE	Runtime Environment
SWC	Software Component
SWCD	Software Component Description

All abbreviations used throughout this document -- except the ones listed here -- can be found in the official AUTOSAR glossary TR-Glossary.

3 Related documentation

References

[1] BAC4 General Concept for the Module Integration
BAC4_General_Concepts_for_the_Module_Integration.pdf

4 Limitations

None.

5 Software Architecture

Dependencies to other Modules

The RmhClassic is a standalone module. I.e. there is no generic part of Rmh, since the functionality of Rmh is only applicable to AUTOSAR CP (Classic AUTOSAR) platform.

Dependencies to AUTOSAR modules

Com

Rmh supports a configuration mode, where it partly takes the role as a complex driver, which directly calls BSW module Com (`Com_TriggerIPDUSend`). In the configuration mode as a pure AUTOSAR SWC it accesses Com indirectly via Rte_Read/Rte_Write.

RTE

As a software component, the Rmh module uses Rte client/server and sender/receiver communication to communicate with BSW modules.

Det

Rmh optionally reports development errors to the Det.

Dependencies to other modules

There are no such dependencies.

6 Integration

Configuration of other modules

The following modules shall be configured, before the module `Rmh` is compiled and linked.

Com

ComSignal for Rmh data element requestedMsgID

The ComSignal for the data element requestedMsgID shall be configured. This signal is currently named ID_FN_INQY in the BMW message catalog and is part of the multiplexed PDU ANFRAGE/Inquiry.

Optional ComSignal for Rmh data element dummySignal

In case the integrator has decided to go for `Rmh` variant ZERO_LENGTH_SIGNAL (see below), then for each Com-PDU, that can be requested, a Com dummy trigger signal with ComBitSize 0 and ComTransferProperty TRIGGERED has to configured and mapped to the corresponding Com-PDU.

Det

Configure the Det accordingly, that it provides the needed P-Port to the `Rmh` SWC.

Configuration

The `Rmh` configuration is formally described in `cfgdesc/RmhClassic_paramdef.arxml`.

Here the configuration values will be set in the right context.

Operation Variants of Rmh

As roughly stated in subsection 6.1.1, ‘‘Com’’ the `Rmh` module can be used in two different modes. These two modes differ in the mechanism used, how to trigger the transmission of the requested Com-PDU. The integrator has to decide which variant fits best in his integration scenario.

Use zero length dummy trigger signals

In this variant, the `Rmh` consists only of one pure AUTOSAR SWC communicating strictly over RTE. To allow the transmission of a Com-PDU, the following strategy is used: For each PDU, which is configured as "can be requested", the `Rmh` module generates an S/R P-Port over which it sends a uint8 dummy data element. The integrator now has to map each of these data elements to a dummy trigger Com signal,

which is part of the PDU to be transmitted and which has the property `ComTransferProperty` set to `TRIGGERED`. Since the task of these dummy signals is solely to trigger the PDU transmission and the content of these signals shall NOT show up in the transmitted PDU on the network, the used Com stack has to support the concept of zero-length-signals. That means the Com dummy signal must support a configuration setting ComBitSize with value 0.

Note: This feature is currently marked as optional in AUTOSAR, so you have to check with your stack vendor, whether he truly supports this.

Pro: This variant does not need any ComplexDeviceDriverSwComponent support. This might be a benefit in ASIL environments, where the usage of non ASIL qualified ComplexDrivers is critical.

Contra: Depending on the number of PDUs that can be requested in your ECU, the `Rmh` has an equivalent number of ports, which consume resources. The additionally needed dummy Com signals also take up certain resources. Finally the configuration overhead for the integrator (data element mapping to Com signal mapping, creation of dummy Com signals) may be higher than in variant two.

Use direct BSW call to Com_TriggerIPDUSend

With this `Rmh` variant, the module consists of two components: An ApplicationSwComponent and a ComplexDeviceDriverSwComponent. The ComplexDeviceDriverSwComponent is nothing more than a small wrapper around the BSW API `Com_TriggerIPDUSend` of the Com module, which is not accessible via RTE. In contrast to variant one, the ApplicationSwComponent part of `Rmh` does not write dummy data elements to the RTE, but calls a C/S operation of the `Rmh` ComplexDeviceDriverSwComponent, which basically directly calls `Com_TriggerIPDUSend`, to transmit the requested PDU.

Pro: Lower resource footprint, less configuration overhead. **Contra**: Direct call to BSW code might be critical in ASIL environments. During generation of `Rmh` code/SWCD the configuration of the Com module must be read. This is an additional dependency that demands a real AUTOSAR compliant Com configuration (which should of course not be a big issue).

The `Rmh` configuration allows configuring which requested message IDs are supported and which Com-PDUs shall be triggered by which request message ID. The `Rmh` configuration contains the following containers at root level:

- CommonPublishedInformation
- Rmh General

Rmh General

This container contains the configuration (parameters) of the `Rmh`.

RmhDevErrorDetect

BOOLEAN parameter to enable/disable the reporting of development errors. This can be useful to detect errors in the Com/Rmh configuration, when Rmh tries to trigger non-existent Com-PDUs.

RmhPduTriggerMode

This is a CHOICE-container, where the integrator can choose between Rmh variant one (RMH_MODE_ZERO_LENGTH_TRIGGER_SIGNAL) and two (RMH_MODE_TRIGGER_IPDU_SEND) (see subsection 6.2.1, "Operation Variants of Rmh").

RmhRequestPduMapping

This container exists in variant one and variant two. Only the contents of this container slightly differ in both variants. For each PDU which can be requested in the ECU one container of this type has to be instantiated.

RmhRequestedMsgId

In both variants, this INTEGER-parameter is needed to identify the requested message. (Note that the ID is taken from the CAN frame ID of the requested message).

RmhTriggerSignalPPortIdentifier

Only in mode RMH_MODE_ZERO_LENGTH_TRIGGER_SIGNAL. This STRING-parameter describes the P-Port name of the Rmh module for sending the dummy data element to trigger the Com-PDU denoted by the RmhRequestMsgId. We suggest, that you name this parameter in a way that the correspondence to the requested Com-PDU is visible.

RmhRequestedComTxPdu

Only in mode RMH_MODE_TRIGGER_IPDU_SEND. This is a reference to the Com-PDU that shall be sent/triggered, when RmhRequestMsgId is requested.

Configuration of the RTE/SchM

The R-Port Det of Rmh has to be connected with the corresponding P-Port of the BSW module Det. The R-Port TriggerIPDUSend of Rmh ApplicationSwComponent has to be connected to the corresponding P-Port TriggerIPDUSend of the Rmh_cdd ComplexDeviceDriverSwComponent in case variant RMH_MODE_TRIGGER_IPDU_SEND has been chosen.

Event Mapping

The Rmh ApplicationSwComponent has the following runnable (in brackets it is noted, whether it can be invoked concurrently):

– RxRequestMsg (false)

Depending at your RTE tooling at least the runnables that cannot be invoked concurrently must be mapped to tasks. The Rmh ComplexDeviceDriverSwComponent (variant RMH_MODE_TRIGGER_IPDU_SEND only) has the following runnable (in brackets it is noted, whether it can be invoked concurrently):

– TriggerComIPDUSend (false)

Depending at your RTE tooling at least the runnables that cannot be invoked concurrently must be mapped to tasks.

Data Mapping

The following S/R data element shall be mapped to the corresponding ComSignal (see subsection 6.1.1, "Com"):

– requestedMsgID (R-Port RxRequestedMsgID)

The following data element(s) mapping is only required if Rmh variant RMH_MODE_ZERO_LENGTH_TRIGGER_SIGNAL is used: The following S/R data element(s) shall be mapped to the corresponding ComSignal(s) (see subsection 6.1.1, "Com"):

– dummySignal (R-Port Trigger$_{<RmhTriggerSignalPPortIdentifier>}$)

Exclusive Areas

Rmh has concurrently NO exclusive area.

Software Integration

SWCD Adaption

The file Rmh_ext_interfaces.arxml contains one external dependency, which is defined here to express the expectations of Rmh (see explanation in [1]):

– The ClientServer interface of Det module to report development errors via ReportError operation.

In case the interface signature looks different in your Det implementation (which is very unlikely), you could adapt it here, at least to a extent that does not change the C-signature of the runnable, implementing the C/S call.

Startup/Initialization

Rmh does not need an explicit startup or initialization. It is triggered by data received events in case it is integrated. In case you do not want the Rmh to interact on data receive events you have to stop the corresponding ComPduGroup, which contains the request message/data element that triggers Rmh.

Normal Operation

See above (startup/initialization). Not applicable.

Shutdown/Deactivation

See above (startup/initialization). Not applicable.