BMW GROUP

Rolls-Royce Motor Cars Limited

Stm Classic Integration Manual

Project	BMW AUTOSAR 4 Core Rel. 3
Author	BMW AG
Release Date	2017-12-14
Version	5.2.0
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
5.2.0	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

Table of Contents

1 Introduction

This Integration Manual describes the basis functionality, API and the configuration and integration of the BMW System Function Stm.

General

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

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

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

Functional overview

The main objective of the Stm functionality is to supervise signals that are communicated on the system busses and maintain these states for the local ECU. This means:

This means:

- Make these states available as modes to other software.
- React on some changes of these states (for instance setting Dem enable conditions).
- React on timeouts of these communicated states (set default values, report error events accordingly).

The Stm module itself is modeled as a SWC residing above the RTE.

2 Acronyms and abbreviations

Abbreviatio / Acronym	Description
AUTOSAR	Automotive Open Systems Architecture group
BAC	BMW AUTOSAR Core
ECU	Electronical Control Unit
BswM	Basic Software Mode Manager
Com	Communication Module
Dem	Diagnostic Event Manager
PDU	Protocol Data Unit
RTE	Runtime Environment
Stm	Status Monitoring
SWC	Software Component
SWCD	Software Component Description

3 Related documentation

References

4 Limitations and Known Issues

Currently there are no limitations known.

5 Software Architecture

Dependencies on AUTOSAR modules

The current version of the Stm Module depends on the following BSW modules:

RTE

As a software component, the Stm module uses RTE client/server communication to communicate with other SWCs and BSW. Additionally the scheduling is done by the RTE.

Com

According to chapter 1.2 the Stm modules main objective is to monitor certain bus signals. Therefore, these signals have to be configured in Com and the data element to Com signal mapping has to be correctly implemented.

Dem

The Stm controls a Dem specific EnableCondition. It uses the Client-Server interface EnableCondition provided by the Dem.

BswM

The Stm module expects that it will be notified via ModeSwitchEvents, whether the ComSignal for centralErrorLock can be received or not. The receive ability depends on the state (started/stopped) of the ComPduGroup the ComSignal is part of. We suggest that the integrator monitors state changes of this ComPduGroup and notifies these by ModeSwitches with a corresponding rule configuration in the BswM.

Dependencies to BMW modules

Stm does not have dependencies to other modules.

6 Integration

Configuration of other Modules

Com

It is optional to configure the following signals in the Com module. In case you disable the corresponding feature in all configuration variants of container StmFeatureActivation, you do not need to configure the related ComSignal. Also in cases, you use RTE transformers to serialize network data into data elements the configuration is done outside of Com.

ComSignal for Stm data element vehicleState

The ComSignal for the data element vehicleState shall be configured. This signal is currently named ST_CON_VEH in the BMW message catalog and is part of the PDU CON_VEH in CAN and Flexray configurations. In case of Ethernet configuration it is part of event VehicleCondition of service VehicleCondition in package BMW.INFRASTRUCTURE. **Note:** This is a new variant of the vehicle state information, which starts in the service pack 2015.

ComSignal for Stm data element energyState

The ComSignal for the data element energyState shall be configured. This signal is currently named ST_ENERG_FZM in the BMW message catalog and is part of the PDU FZZSTD in CAN and Flexray configurations. In case of Ethernet configuration, it is part of field VehicleStatus (member statusEnergyFZM) of service StatusEnergy in package BMW.INFRASTRUCTURE. **Note:** This is a new variant of the vehicle state information, which starts in the service pack 2015.

ComSignal for Stm data element centralErrorLock

The ComSignal for the data element centralErrorLock shall be configured. This signal is currently named ST_ILK_ERRM_FZM in the BMW message catalog and is part of the PDU FZZSTD in CAN and Flexray configurations. In case of Ethernet configuration, it is part of field VehicleStatus (member statusInterlockErrorMemoryFZM) of service StatusEnergy in package BMW.INFRASTRUCTURE.

BswM

Configure a rule and depending ActionLists to switch a ModeSioauratioGroupf tait is the Mod,d depending the ComSignaGroupf state to

Dem

Stm interacts with Dem in a way, that it controls a so called Dem EnableCondition, which in turn controls if status changes (triggered by Dem_ReportErrorStatus/Dem_SetEventStatus) for Dem Events which are linked to this EnableCondition shall be ignored or not.

DemEnableConditionSupport

DemEnableConditionSupport hast to be enabled in Dem configuration.

DemEnableCondition/DemEnableConditionGroup

Inside Dem configuration an unique/special EnableCondition for Stm shall be configured. This EnableCondition shall be referenced by all EnableConditionGroups in the system, that depend on this EnableCondition. Example: At BMW the permission to report communication error relevant Dem events depends on the state of different sources (operating mode, diagnostic state and Stm centralErrorLock). Each of these sources is connected to its own EnableCondition and all these conditions are combined in one EnableConditionGroup. At the end, all Dem Events that represent communication errors shall reference this EnableGroup via their DemEnableConditionGroupRef parameter.

Configuration

The Stm configuration allows configuring the timeout supervision values for the four different ComSignals. The Stm configuration contains one configuration container StmGeneral and one container StmFeatureActivation.

StmGeneral

This container contains the configuration (parameters) of the Stm.

StmInitialCentralErrorLockTimeout

This FLOAT-parameter specifies the first/initial timeout in seconds after the ComPduGroup started to which the ComSignal mapped to data element centralErrorLock belongs. The default value as defined by Stm represents the value that is demanded by BMW specification.

Note: This parameter is ONLY for documentation purposes. The integrator has to make manually sure that the ComFirstTimeout configuration parameter of BSW module Com of the relevant signal is set to this value. The Stm module is a SWC and therefore can only parameterize values in its SWCD. Since there is in AUTOSAR NO connection from the SWCD to ComFirstTimeout configuration, this configuration value can only be documented.

StmCentralErrorLockTimeout

This FLOAT-parameter specifies the regular timeout in seconds after the ComPduGroup started to which the ComSignal mapped to data element centralErrorLock belongs. The default value as defined by Stm represents the value that is required by BMW specification at the time of development of this module. In case this requirement changes, the value can be adapted in a range from 1.0 to 32.0 seconds.

The chosen value will be generated into the SWCD of the Stm module. The BSW stack RTE tooling should care for the correct timeout configuration in the Com configuration. That is, the Com configuration value ComTimeout of the relevant signal should be set to the chosen value automatically. In case the tooling does not support this feature, the integrator has to manually correct the Com configuration.

StmEnergyStateTimeout

This FLOAT-parameter specifies the regular timeout in seconds after the ComPduGroup started to which the ComSignal mapped to data element energyState belongs. The default value as defined by Stm represents the value that is required by BMW specification at the time of development of this module. In case this requirement changes, the value can be adapted in a range from 1.0 to 32.0 seconds.

The chosen value will be generated into the SWCD of the Stm module. The BSW stack RTE tooling should care for the correct timeout configuration in the Com configuration. That is, the Com configuration value ComTimeout of the relevant signal should be set to the chosen value automatically. In case the tooling does not support this feature, the integrator has to manually correct the Com configuration.

StmVehicleStateTimeout

This FLOAT-parameter specifies the regular timeout in seconds after the ComPduGroup started to which the ComSignal mapped to data element vehicleState belongs. The default value as defined by Stm represents the value that is required by BMW specification at the time of development of this module. In case this requirement changes, the value can be adapted in a range from 0.1 to 32.0 seconds.

The chosen value will be generated into the SWCD of the Stm module. The BSW stack RTE tooling should care for the correct timeout configuration in the Com configuration. That is, the Com configuration value ComTimeout of the relevant signal should be set to the chosen value automatically. In case the tooling does not support this feature, the integrator has to manually correct the Com configuration.

StmComVariant

With this enumeration switch it can be configured whether Stm module is integrated within a boardnet where the central error lock signal/event is sent cyclically or not. In case Stm is integrated within a boardnet, where the central error lock is only sent "on-change", Stm has to implement timeout monitoring on its own and will call back the source of the central error lock signal/event via a C/S getter call. Currently the central error lock signal/event is sent cyclically in all BMW boardnet configurations, therefore the DEFAULT and to be used value of this parameter shall be set to STM_COM_CYCLIC_DATAELEMENTS.

StmFeatureActivation

This container is a so-called multiple configuration container. It allows you to configure multiple instances of these configuration, where you can choose at runtime, which of these configurations shall be chosen. This allows post-build-selectable integration scenarios. In this container you can enable/disable certain signal supervisions. It is designed with a postbuild-selectable configuration concept, to support scenarios, where the same ECU software is integrated in different vehicle configurations/bordnet topologies, where the integrator needs the possibility to select at runtime which signal supervisions are needed in the current vehicle. For a general concept, how the BMW AUTOSAR Core handles SWCs with post build selectable support see [3].

StmCentralErrorLockEnabled

Enable/Disable the central error lock supervision.

StmEnergyStateEnabled

Enable/Disable the energy state supervision.

StmVehicleStateEnabled

Enable/Disable the vehicle state supervision.

Configuration of the RTE

After performing the steps indicated in chapter 6.1 and 6.2, the RTE configuration can be started. In other way, the RTE will report an interface incompatibility error.

Assembly connectors

Dem connection

During integration, you have to connect the R-Port CentralErrorLockEnableCondition with the related Dem P-Port, that Dem must provide after the Stm specific EnableCondition has been configured (see chapter 6.1.3).

ModeDeclarationGroup connection

The R-Port EMLockSignalReceptionModeNotificationPort of Stm has to be connected with the corresponding P-Port of the mode manager. In case, you followed the suggestion (see chapter 5.1.4) to

control this ModeDeclarationGroup through the BswM module, the generated SWCD of the BswM module will provide the corresponding P-Port.

Event Mapping

Stm has the following runnables (in brackets it is noted, whether it can be invoked concurrently):

- Runnable_ErrorCentralErrorLock (true).
- Runnable_ErrorEnergyMode (true).
- Runnable_ErrorVehicleState (true).
- Runnable_StopCentralErrorLockSupervision (false).
- Runnable_InitialCELSupervision (false).
- Runnable_ReceiveCentralErrorLock (true).
- Runnable_ReceiveEnergyMode (true).
- Runnable_ReceiveVehicleState (true).

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 elements shall be mapped to the corresponding ComSignals (see chapter 6.1):

- centralErrorLock (R-Port CentralErrorLockRx).
- energyState (R-Port EnergyModeRx).
- vehicleState (R-Port VehicleStateRx).

Note: The three data elements shown above are primitive data types from the viewpoint of Stm. In the bordnet configuration these data types are embedded within signalgroups/complex datatypes. In the AUTOSAR-Extracts provided by BMW therefore complex (record types) data elements are provided within the corresponding ports of the outer ECU composition. To connect sub-elements (members) of the ports of the outer composition with the primitive type data elements of Stm ports a port-interface mapping has to created, which is then attached to the delegation assembly connector.

Example for a port interface mapping for centralErrorLock datat-element:

```
<?xml version="1.0"?>
<PORT-INTERFACE-MAPPING-SET>
  <SHORT-NAME>StmMappingSet</SHORT-NAME>
  <PORT-INTERFACE-MAPPINGS>
    <VARIABLE-AND-PARAMETER-INTERFACE-MAPPING>
      <SHORT-NAME>Mapping_Stm_CentralErrorLockRx</SHORT-NAME>
      <DATA-MAPPINGS>
        <DATA-PROTOTYPE-MAPPING>
          <FIRST-DATA-PROTOTYPE-REF DEST="VARIABLE-DATA-PROTOTYPE">/BMW/INFRASTRU
          <SECOND-DATA-PROTOTYPE-REF DEST="VARIABLE-DATA-PROTOTYPE">/BMW/Platform
          <SUB-ELEMENT-MAPPINGS>
            <SUB-ELEMENT-MAPPING>
```

```
                    <FIRST-ELEMENTS>
                      <APPLICATION-COMPOSITE-DATA-TYPE-SUB-ELEMENT-REF>
                        <APPLICATION-COMPOSITE-ELEMENT-IREF>
                          <ROOT-DATA-PROTOTYPE-REF DEST="VARIABLE-DATA-PROTOTYPE">/BMW
                          <TARGET-DATA-PROTOTYPE-REF DEST="APPLICATION-RECORD-ELEMENT"
                        </APPLICATION-COMPOSITE-ELEMENT-IREF>
                      </APPLICATION-COMPOSITE-DATA-TYPE-SUB-ELEMENT-REF>
                    </FIRST-ELEMENTS>
                  </SUB-ELEMENT-MAPPING>
                </SUB-ELEMENT-MAPPINGS>
              </DATA-PROTOTYPE-MAPPING>
            </DATA-MAPPINGS>
          </VARIABLE-AND-PARAMETER-INTERFACE-MAPPING>
        </PORT-INTERFACE-MAPPINGS>
      </PORT-INTERFACE-MAPPING-SET>
```

Exclusive Areas

The exclusive area "CentralErrorLockReadWrite_EA" has to be configured in the RTE.

Software Integration

SWCD

The file Stm_ext_interfaces.arxml contains two external dependencies, which are defined here to express the expectations of Stm (see explanation in [1]):

- A mode declaration group that indicates, whether the centralErrorLock signal is currently receivable or not.
- The ClientServer interface of Dem module to control so called EnableConditions (see chapter 6.1.3).

Mode Declaration Group ErrorMemoryLockSignalReceptionMode

The Stm expects that there exists a ModeDeclarationGroup that is controlled by integration code, that indicates, whether the centralErrorLock is currently receivable or not. In general, this depends on the state of the ComPduGoup, which the related ComSignal is connected to. The integrator shall configure the BswM rule system in a way that as soon as the corresponding PduGroup is stopped, the ModeDeclarationGroup changes to mode EM_LOCK_NOT_RECEIVABLE. In case the PduGroup is started, the mode shall be changed to EM_LOCK_RECEIVABLE.

Interface EnableCondition of Dem

In Stm_ext_interfaces.arxml you find BMWs interpretation of EnableCondition interface of the Dem BSW module. In case the interface signature looks different in your Dem 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/Initialisation

Stm does not need a dedicated initialization or startup. It has no cyclically runnables but is fully driven by data received events/data received error events/mode switch events.

Normal Operation

Stm is automatically in normal operation.

Shutdown/Deactivation

Not needed (see above).