Elektrobit

# EB tresos® AutoCore Generic 8 Com Services documentation

release notes update for the IpduM module

product release 8.8.7

Elektrobit Automotive GmbH
Am Wolfsmantel 46
91058 Erlangen, Germany
Phone: +49 9131 7701 0
Fax: +49 9131 7701 6333
Email: info.automotive@elektrobit.com

## Technical support

https://www.elektrobit.com/support

## Legal disclaimer

Confidential information.

All brand names, trademarks, and registered trademarks are property of their rightful owners and are used only for description.

# Table of Contents

# 1. Overview

This document provides you with the release notes to accompany an update to the `IpduM` module. Refer to the changelog Section 2.1, “Change log” for details of changes made for this update.

Release notes details

- EB tresos AutoCore release version: 8.8.7
- EB tresos Studio release version: 29.2.0
- AUTOSAR R4.0 Rev 3
- Build number: B577598

# 2. IpduM module release notes

- AUTOSAR R4.0 Rev 3
- AUTOSAR SWS document version: 2.2.0
- Module version: 3.3.50.B577598
- Supplier: Elektrobit Automotive GmbH

## 2.1. Change log

This chapter lists the changes between different versions.

### Module version 3.3.50

2022-11-04

- ASCIPDUM-1187 Fixed known issue: Unexpected container triggering with LAST_IS_BEST contained PDUs.
- Internal module improvement. This module version update does not affect module functionality.

### Module version 3.3.49

2022-10-28

- ASCIPDUM-1187 Fixed known issue: Unexpected container triggering with LAST_IS_BEST contained PDUs.
- Internal module improvement. This module version update does not affect module functionality.

### Module version 3.3.48

2022-09-16

- Internal module improvement. This module version update does not affect module functionality.

### Module version 3.3.47

2022-08-19

- Updated requirement Id format in module documentation and source code tracing comments. Note: This does not change the Baseline, nor functionality.

## Module version 3.3.46

2022-06-10

- Internal module improvement. This module version update does not affect module functionality

## Module version 3.3.45

2022-05-13

- Internal module improvement. This module version update does not affect module functionality

## Module version 3.3.44

2022-04-08

- Fixed known issue: Potential endless loop when processing received container PDUs

## Module version 3.3.43

2022-03-18

- Internal module improvement. This module version update does not affect module functionality

## Module version 3.3.42

2022-02-18

- Added CanFd padding service according to SAE J1939-22

## Module version 3.3.41

2022-01-28

- Added support for Multi-PG feature according to SAE J1939-22

## Module version 3.3.40

2021-06-25

- Internal module improvement. This module version update does not affect module functionality

## Module version 3.3.39

2021-05-28

- ASCIPDUM-1115 Fixed known issue: Incorrect memory allocated for TX container instances if the PduLengthType is uint32
- ASCIPDUM-1116 Fixed known issue: Undefined behavior when PduLengthType is uint32
- ASCIPDUM-1118 Fixed known issue: NULL pointer dereferenced with Flexible MainFunction Allocation and no TxPathways

## Module version 3.3.38

2021-04-30

- Internal module improvement. This module version update does not affect module functionality

## Module version 3.3.37

2021-03-05

- ASCIPDUM-1111 Fixed known issue: Underallocated static container not trimmed in case of partial update

## Module version 3.3.36

2021-02-12

- Internal module improvement. This module version update does not affect module functionality

## Module version 3.3.35

2021-01-22

- Internal module improvement. This module version update does not affect module functionality

## Module version 3.3.34

2020-12-11

- Added support for Variant Handling

## Module version 3.3.33

2020-09-25

- ASCIPDUM-1096 Fixed known issue: IPDUM_TRIGGERTRANSMIT Container with static Contained PDUs are not triggered if all IpduMContainedTxPdus were updated
- Added support for ASR4.4 IpduMContainerRxAcceptContainedPdu=IPDUM_ACCEPT_ALL behavior

## Module version 3.3.32

2020-07-31

- Internal module improvement. This module version update does not affect module functionality

## Module version 3.3.31

2020-06-19

- Internal module improvement. This module version update does not affect module functionality

## Module version 3.3.30

2020-05-22

- ASCIPDUM-1081 Fixed known issue: SduLength incorrectly established for Static Containers
- ASCIPDUM-1080 Fixed known issue: IpduM fails to generate if IpduMContainedTxPduPriority is not set and IpduMContainedTxPduPriorityHandling is TRUE

## Module version 3.3.29

2020-04-24

- Internal module improvement. This module version update does not affect module functionality

## Module version 3.3.28

2020-03-25

- ASCIPDUM-1062 Fixed known issue: The transmission of a multiplexed I-PDU might be blocked within the IpduM
- ASCIPDUM-1063 Fixed known issue: Return value for IpduM_Transmit is not specified in case of JIT update failure

## Module version 3.3.27

2020-02-21

- ASCIPDUM-1034 Fixed known issue: Off-by-one Error Causing an Out-Of-Bounds Read access

## Module version 3.3.26

2020-01-24

- ASCIPDUM-1029 Fixed known issue: Undefined data in dynamic AND/OR static PDU provided
- ASCIPDUM-1044 Fixed known issue: Transmission of containers stops after erroneous double transmission
- ASCIPDUM-1048 Fixed known issue: Parameter IpduMDequeueInTxConf is lacking constraint in the description
- ASCIPDUM-1049 Fixed known issue: TxConfirmation is denied when transmission is interrupted by IpduM_TriggerTransmit()

## Module version 3.3.25

2019-10-31

- ASCIPDUM-989 Fixed known issue: Generation error of IpduM where the container TxPdu has multiple PduR destinations
- ASCIPDUM-1028 Fixed known issue: Deferred IpduMContainerRxPdus can become corrupted with IpduMDedicatedIpduProcessingSupport

## Module version 3.3.23

2019-09-06

- Internal module improvement. This module version update does not affect module functionality

## Module version 3.3.22

2019-08-12

- ASCIPDUM-1021 Fixed known issue: High latency during transmission of TriggerTransmit-Queued Container PDUs
- ASCIPDUM-1022 Fixed known issue: Missing TxConfirmation for LAST_IS_BEST queued Contained PDUs

## Module version 3.3.21

2019-07-05

- Internal module improvement. This module version update does not affect module functionality.

## Module version 3.3.20

2019-06-14

- ASCIPDUM-1007 Fixed known issue: Duplicated container PDU is transmitted when IpduM_TriggerTransmit interrupts IpduM_Transmit
- ASCIPDUM-1012 Fixed known issue: Buffer overflow allows code injection
- ASCIPDUM-1009 Fixed known issue: Integer overflow might result in unexpected behavior of the ECU
- ASCIPDUM-1013 Fixed known issue: Out-of-bounds memory access for reception of a Container PDU with length 1

## Module version 3.3.19

2019-05-21

- Internal module improvement. This module version update does not affect module functionality.

## Module version 3.3.18

2019-04-18

- Internal module improvement. This module version update does not affect module functionality.

## Module version 3.3.17

2019-03-22

- ASCIPDUM-999 Fixed known issue: Undefined behavior if a contained PDU with a PDU length of zero is transmitted

## Module version 3.3.16

2019-02-15

- Internal module improvement. This module version update does not affect module functionality.

## Module version 3.3.15

2018-12-13

- Added support for priority handling of Tx ContainedIPdus with LastIsBest collection semantics

## Module version 3.3.14

2018-10-26

- PduR_IpduMTriggerTransmit and PduR_IpduMTxConfirmation are called inside ExclusiveArea
- ASCIPDUM-969 Fixed known issue: Deferred Rx container PDUs are incorrectly processed in IpduM_MainFunctionRx()
- ASCIPDUM-964 Fixed known issue: Send timeout is wrongly considered for LAST IS BEST container PDUs

## Module version 3.3.13

2018-09-28

- Internal module improvement. This module version update does not affect module functionality

## Module version 3.3.12

2018-07-27

- Add support for Contained to Container PDU Mapping Based on Static Configuration.

## Module version 3.3.11

2018-06-22

- Internal module improvement. This module version update does not affect module functionality

## Module version 3.3.10

2018-05-25

- Internal module improvement. This module version update does not affect module functionality

## Module version 3.3.9

2018-04-20

- Internal module improvement. This module version update does not affect module functionality
- Added PbcfgM support
- Improved size announcement for trigger transmit Container Tx
- Add support for UINT32 PduLengthType.

## Module version 3.3.8

2018-02-16

- Create per-Partition BswImplementation and BswInternalBehavior elements in BSWMD
- ASCIPDUM-885 Fixed known issue: Compilation fails for enabled IpduMDedicatedIpduProcessingSupport

## Module version 3.3.7

2018-01-19

- ASCIPDUM-885 Fixed known issue: Out-of-bounds access for deferred container Rx PDUs
- Flexible allocation of PDUs to MainFunctions

## Module version 3.3.6

2017-12-15

- ASCIPDUM-881 Fixed known issue: Container PDU is not transmitted when recovered from bus off situation
- Use BinarySearch for matching ContainedPdu ID
- ASCIPDUM-884 Fixed known issue: Generator error for timeout timers divisible by IpduMTxTimeBase

## Module version 3.3.5

2017-11-17

- ASCIPDUM-872 Fixed known issue: Missing TxConfirmation of contained PDUs
- ASCIPDUM-873 Fixed known issue: Loss of data for bursts of contained PDUs
- Deferred finalization (frozen) for TriggerTransmit Container PDU

## Module version 3.3.4

2017-10-20

- ASCIPDUM-871 Fixed known issue: IpduM_MainFunctionRx() blocks the interrupt too long

## Module version 3.3.3

2017-09-22

- ASCIPDUM-856 Fixed known issue: Multi-PDU-to-container handling is not post-build capable

## Module version 3.3.2

2017-08-25

- Internal module improvement. This module version update does not affect module functionality

## Module version 3.3.1

2017-07-28

- Improve sending of container PDU due to send timeout trigger
- ASCIPDUM-837 Fixed known issue: Container PDU delayed with transmission confirmation timeout time
- Internal module improvement. This module version update does not affect module functionality

## Module version 3.3.0

2017-06-30

- Internal module improvement. This module version update does not affect module functionality
- Lower layer module isn't informed about the transmit request in case the trigger transmit mode is used for the container PDU
- ASCIPDUM-831 Fixed known issue: Transmission timer is not initialized when adding a contained I-PDU to a new container PDU

## Module version 3.2.18

2017-06-02

- ASCIPDUM-804 Fixed known issue: Contained PDU is not transmitted in case of IPDUM_COLLECT_LAST_IS_BEST
- ASCIPDUM-805 Fixed known issue: Container PDU transmitted with wrong contained PDUs in case of IPDUM_COLLECT_LAST_IS_BEST
- ASCIPDUM-808 Fixed known issue: Container PDU is not transmitted on the network for a long period
- ASCIPDUM-811 Fixed known issue: Container PDU transmits wrong contained PDUs in case of IPDUM_COLLECT_QUEUED
- ASCIPDUM-814 Fixed known issue: Container PDU transmitted twice followed by the loss of the next instance of the container PDU
- ASCIPDUM-817 Fixed known issue: Corruption of run-time data during IpduM_TxConfirmation()
- IPDUM_GET_SHORT_HEADER_ID depends on CPU_BYTE_ORDER
- Removed restriction to little-endian byte order for contained I-PDU headers (Multiple-PDU-to-Container handling)

## Module version 3.2.17

2017-05-05

- Internal module improvement. This module version update does not affect module functionality

## Module version 3.2.16

2017-03-31

- ASCIPDUM-768 Fixed known issue: Invalid memory access when Container Tx PDU length exceeds configured PduLength
- Internal module improvement. This module version update does not affect module functionality

## Module version 3.2.15

2017-03-03

- ASCIPDUM-758 Fixed known issue: Header ID and DLC are processed incorrectly on big-endian platform
- Internal module improvement. This module version update does not affect module functionality
- Internal module improvement. This module version update does not affect module functionality

## Module version 3.2.14

2017-02-03

- ASCIPDUM-749 Fixed known issue: IpduM_MainFunctionRx() and IpduM_MainFunctionTx() violate cycle time
- ASCIPDUM-754 Fixed known issue: Incorrect association of container PDU and contained PDUs
- ASCIPDUM-750 Fixed known issue: Dereferenced null pointer in `IpduM_MainFunctionRx()`
- Internal module improvement. This module version update does not affect module functionality

## Module version 3.2.13

2017-01-05

- ASCIPDUM-739 Fixed known issue: Out of bounds access during reception
- Added Support for queuing of container PDUs.

## Module version 3.2.12

2016-11-04

- Corrected setting of transmission timer of container PDU

## Module version 3.2.11

2016-09-23

- Incorporated Bugzilla RfC 71983: Introduce separate main functions for reception and transmission

## Module version 3.2.10

2016-07-01

- Added *Multiple-PDU-to-Container*handling for Tx

## Module version 3.2.9

2016-02-05

- Added support for Debug & Trace with custom header file configurable via parameter `BaseDbgHeaderFile`

## Module version 3.2.8

2015-06-19

- Added *Multiple-PDU-to-Container*handling for Rx
- ASCIPDUM-586 Fixed known issue: The IpduM module reports an error for legal setting of `IpduMInitializationBySignalValue and IpduMEnableJitUpdate`

## Module version 3.2.7

2015-01-07

- Removed AUTOSAR 3.x compliant symbolic name value macros and updated the logic to only provide AUTOSAR 4.0.2 compliant macros if macro IPDUM_PROVIDE_LEGACY_SYMBOLIC_NAMES is defined

- Changed signature of Com Rx callout `IpduM_ProcessRequestPdu`according to AUTOSAR bugzilla Rfc #52342

## Module version 3.2.6

2014-10-02

- Implemented *Just-In-Time*update of parts
- Added initialization of multiplexed I-PDU with initialization data of dynamic and static part from COM module

## Module version 3.2.5

2014-04-25

- Internal module improvement. This module version update does not affect module functionality

## Module version 3.2.4

2013-10-11

- ASCIPDUM-452 Fixed known issue: EB-specific feature calls `Com_TriggerIPDUSend()` with wrong PDU-ID
- Changed the module structures for optimal memory usage
- Updated symbolic name value naming schema according to AUTOSAR 4.0 Rev 3

## Module version 3.2.3

2013-06-14

- Added nonfunctional code improvements for shared data handling
- ASCIPDUM-425 Fixed known issue: If `IpduMTxConfirmationTimeout`is set to zero, confirmation calls to the PduR are blocked
- ASCIPDUM-429 Fixed known issue: The IpduM generates code even if the configured destination field does not fit in the I-PDU
- ASCIPDUM-412 Fixed known issue: The IpduM code generation fails if the IpduM configuration refers to a PduR destination PDU that has disabled confirmation PDU-ID and disabled IpduM Tx confirmation

- ASCIPDUM-437 Fixed known issue: The IpduM generates erroneous code if `IpduMDestinationBit`is not byte-aligned while byte copy is enabled or `IpduMDestinationBit`is not set to zero while zero copy is enabled

## Module version 3.2.2

2013-02-07

- ASCIPDUM-342 Fixed known issue: The PduR IpduM transmission confirmation function is called with the wrong PDU-ID

## Module version 3.2.1

2012-10-12

- Changed the top-level structure of the software-component description in the ARXML files from `/AUTOSAR/IpduM`to `/AUTOSAR_IpduM`
- Updated to AUTOSAR 4.0 Rev 3

## Module version 3.2.0

2012-09-28

- ASCIPDUM-315 Fixed known issue: Transmission of incorrect data in case of zero size transmit queues
- Implemented AUTOSAR 4.0 ComStack Handle ID policy

## Module version 3.1.2

2012-08-17

- Implemented definition of *Exclusive Area*in Basic Software Module Description

## Module version 3.1.1

2012-06-22

- Internal module improvement. This module version update does not affect module functionality

### Module version 3.1.0

2012-03-16

- Updated the include structure regarding the symbolic name value header files
- Updated `SchM_Enter/Exit()` calls to match AUTOSAR 4.0

### Module version 3.0.4

2012-02-17

- Added BSWMD support

### Module version 3.0.3

2012-01-20

- Improved speed of the template generator

### Module version 3.0.2

2011-12-09

- Internal module improvement. This module version update does not affect module functionality

### Module version 3.0.1

2011-09-30

- ASCIPDUM-210 Fixed known issue: The unattended wizard *Calculate Handle IDs*does not generate Handle IDs for the IpduM

### Module version 3.0.0

2011-09-02

- Initial AUTOSAR 4.0 version

## 2.2. New features

- CanFd Padding Service according to SAE J1939-22

## 2.3. Elektrobit-specific enhancements

This chapter lists the enhancements provided by the module.

- Priority queuing for transmission of dynamic PDUs

  Description:

  The IpduM is able to provide a priority queue for each transmit path. `IpduMQueueSize` specifies the queue size. A value of `0` means not using a queue at all. `IpduMTxDynamicPriority` defines the priority of each PDU. `0` stands for the highest priority.

- Requesting service messages to request the transmission of a specific PDU from another ECU

  Description:

  To support a special type of multiplexed messages called requesting service messages, the functionality of the AUTOSAR IpduM is extended. A requesting service message is identified by a selector value set to `1`. On reception of a requesting service message, the ECU sends out the requested Com I-PDU. The requested Com I-PDU is specified by a global PDU-ID in the data field of the dynamic part of the requesting service message.

- Automatic selector for *automatic setting of the selector value* by the IpduM

  Description:

  IpduM is extended to support *automatic setting of selector value*. The configuration parameter `IpduMTxAutomaticSelector` is used to support this feature. If this parameter is enabled, the selector values for the transmit PDUs are set by the IpduM itself and if disabled, the selector value is not set by the IpduM. The selector value is also configurable using `IpduMTxSelectorValue`. During reception, the dynamic part is accepted and assembled only if the selector value is valid.

- Code and run-time optimizations

  Description:

  The code and run-time has been optimized. This optimization includes:

  - Detection of development errors

    Disabling this feature reduces the ROM consumption and reduces the execution time of the module code.

  - Usage of static parts

    Disabling this feature reduces the execution time and reduces the ROM consumption of the module code.

  - Version information API

  Disabling this API reduces the ROM consumption of the module code.

  - Zero Copy

    Enabling this feature reduces the execution time and reduces the ROM consumption of the module code.

  - Byte-wise copy

    Enabling this feature reduces the execution time and reduces the ROM consumption of the module code.

  - Dynamic part queue

    Disabling this feature reduces the execution time and reduces the ROM consumption of the module code.

  - Automatic selector

    Disabling this feature reduces the execution time and reduces the ROM consumption of the module code.

  - Static memory allocation

    Decreasing this parameter reduces the RAM consumption of the module configuration.

- Optional initialization of static and dynamic parts

  Description:

  For the enabled `IpduMInitializationBySignalValue`, the static and dynamic parts are initialized in retrieving signal values from the upper layer module by `IpduM_Init`. Otherwise the static and dynamic parts are only initialized by the unused area pattern configured.

  Rationale:

  The pre-compile switch was introduced to allow backward compatibility of the mandatory parameter `IpduMInitialDynamicPart`.

- Optional *Just-In-Time* update

  Description:

  For the enabled `IpduMEnableJitUpdate`, the *Just-In-Time* update functionality is provided in general. For the individual static and dynamic parts, the parameter `IpduMJitUpdate` has to be handled according to the SWS.

  Rationale:

  The pre-compile switch was introduced to allow backward compatibility of the mandatory parameter `IpduMInitialDynamicPart`.

- Possibility to select whether dequeuing of IpduMContainerTxPdus with *IpduMContainerTxTriggerMode* set to *IPDUM_DIRECT* happens in the context of `IpduM_MainFunctionTx()` or `IpduM_TxConfirmation()` by the configuration parameter `IpduMDequeueInTxConf`. Queued ContainerTxPdus with IpduMContainerTxTriggerMode set to IPDUM_TRIGGERTRANSMIT are dequeued in IpduM_TxConfirmation() regardless of the value of the parameter.
- Binary Search algorithm for matching contained PDU header ID

  Description:

  For matching contained PDUs header ID the `IpduM` module makes use of `Binary Search` algorithm in order to reduce runtime consumption. This is needed especially when a container PDU with `IpduMContainerRxAcceptContainedPdu` set to `IPDUM_ACCEPT_ALL` is received.
- J1939 Multi-PG support according to SAE J1939-22

  Description:

  In order to accomodate the Multi-PG feature from the J1939 stack, the following features are present in `IpduM`:

  - MetaData handling with demultiplexed parts on the receiver side according to the Autosar specification.
  - Support for Metadata handling (CanId32 type only) on the transmission side with the addition that the change of Metadata value is a triggering factor for the container.
  - CanFd Padding Service according to SAE J1939-22.

## 2.4. Deviations

This chapter lists the deviations of the module from the AUTOSAR standard.

- IpduM supports only little endian byte order for IpduM segments

  Description:

  For the configuration parameter `IpduMByteOrder` [ECUC_IpduM_00162] in the configuration container `IpduMRxIndication` and `IpduMTxRequest` only the value `LITTLE_ENDIAN` is allowed. This also violates [SWS_IpduM_00166] which requests that Com and IpduM must have the same setting regarding the endianness.

  Rationale:

  The EB tresos Studio Com importer is able to convert `BIG_ENDIAN` segments to `LITTLE_ENDIAN` segments. For direct configuration and import from ECU configuration files this limitation has to be considered, but does not restrict any PDU layout.

Requirements:

ECUC_IpduM_00162

- `IpduMRxDirectComInvocation` not supported

Description:

The configurable optimization *direct invocation of the COM module (bypassing the PduR) as defined in SWS_IpduM_00140* is not implemented.

Rationale:

This optimization violates the AUTOSAR layered architecture.

Requirements:

SWS_IpduM_00140, ECUC_IpduM_00142

- `PduR_IpduMRxIndication()`, `PduR_IpduMTransmit()`, `PduR_IpduMTriggerTransmit()`, and `PduR_IpduMTxConfirmation()` are mandatory

Description:

`PduR_IpduMRxIndication()`, `PduR_IpduMTransmit()`, `PduR_IpduMTriggerTransmit()`, and `PduR_IpduMTxConfirmation()` are mandatory and not optional interfaces as specified by requirement SWS_IpduM_00105.

Rationale:

`PduR_IpduMRxIndication()`, `PduR_IpduMTransmit()`, `PduR_IpduMTriggerTransmit()`, and `PduR_IpduMTxConfirmation()` can only be optional when following optimizations are implemented:

- The IpduM Tx path can be disabled when PDUs are only received via the IpduM.
- The IpduM Rx path can be disabled when PDUs are only sent via the IpduM.

These optimizations are not implemented by the IpduM.

Requirements:

SWS_IpduM_00105, SWS_IpduM_00104

- Configuration parameter `IpduMTxConfirmationPduId` is not OPTIONAL.

Description:

As per AUTOSAR_SWS_IPDUMultiplexer V4.0.3, `IpduMTxConfirmationPduId` can be optional. But it is also mentioned that the existence of this parameter is essential for the PduR generation tool to actually find a `symbolicNameValue` for the outgoing PDU. Also disabling the parameter demands lots of rework in the generator.

Requirements:

ECUC_IpduM_00158

- Non-compliant deviations in vendor-specific module definition file

  Description:

  The vendor-specific module definition file (VSMD) has non-compliant deviations to the AUTOSAR specification:

  Violations against Rule EcucSws_1014: Additional vendor specific parameter definitions (using ParameterTypes), container definitions and references shall be added to the VSMD according to the alphabetical order.

  - StMD-Node: `/AUTOSAR/IpduM`
  - StMD-Node: `/AUTOSAR/IpduM/IpduMConfig/IpduMRxPathway/IpduMRxIndication/IpduMRxDynamicPart/IpduMSegment`
  - StMD-Node: `/AUTOSAR/IpduM/IpduMConfig/IpduMRxPathway/IpduMRxIndication/IpduMRxStaticPart/IpduMSegment`
  - StMD-Node: `/AUTOSAR/IpduM/IpduMConfig/IpduMTxPathway/IpduMTxRequest`
  - StMD-Node: `/AUTOSAR/IpduM/IpduMConfig/IpduMTxPathway/IpduMTxRequest/IpduMTxDynamicPart/IpduMSegment`
  - StMD-Node: `/AUTOSAR/IpduM/IpduMConfig/IpduMTxPathway/IpduMTxRequest/IpduMTxStaticPart/IpduMSegment`
  - StMD-Node: `/AUTOSAR/IpduM/IpduMGeneral`

  Rationale: Additional vendor specific parameter definitions have a specific order in the GUI which may differ to the alphabetical order.

- `IpduMDequeueInTxConf` selects when dequeuing happens

  Description:

  If `IpduMDequeueInTxConf` is *FALSE*, dequeuing happens only in `IpduM_MainFunctionTx()`.

  Applies only to IpduMContainerTxPdus with IpduMContainerTxTriggerMode set to IPDUM_DIRECT.

  Queued ContainerTxPdus with IpduMContainerTxTriggerMode set to IPDUM_TRIGGERTRANSMIT are dequeued in IpduM_TxConfirmation() regardless of the value of the parameter.

  Requirements:

  SWS_IpduM_00190

- `IpduMDequeueInTxConf` selects when dequeuing happens

  Description:

  If `IpduMDequeueInTxConf` is *TRUE*, dequeuing happens also in `IpduM_TxConfirmation()`.

  Requirements:

  SWS_IpduM_00190

- Dequeuing in case of overflow

  Description:

  By the requirement SWS_IpduM_00199 it is specified that the oldest instance shall be overwritten if the queue is full. SWS_IpduM_00190 does state that the next oldest one shall be processed without handling the overflow case. As it is not considered normal operation and it would increase complexity, in case of an overflow the newest container would be dequeued before the oldest one.

  Requirements:

  SWS_IpduM_00190

- Max value of IpduMContainerQueueSize

  Description:

  The upper limit of the configuration parameter IpduMContainerQueueSize is 254 due to the fact that the number of instances is stored in a uint8, for both RX and TX.

  Requirements:

  ECUC_IpduM_00185

- `PduR_IpduMTriggerTransmit` transmit data for each contained

  Description:

  The container is triggered based on `PduLength` of the contained PDUs and when preparing for transmission data for each contained is obtained from `PduR` (through `PduR_IpduMTriggerTransmit` ).

  Requirements:

  SWS_IpduM_00231

- MultiplexConfigurationChanges

  Description:

  Changes to the Multiplexer configuration parameters in R20-11 that are not in the implementation.

Rationale:

Specified in SWS R20-11 but not implemented.

Requirements:

ECUC_IpduM_00170, ECUC_IpduM_00160, ECUC_IpduM_00171, ECUC_IpduM_00169, ECUC_IpduM_00168, ECUC_IpduM_00165,SWS_IpduM_00224

- ReliableTxConfirmation

  Description:

  Reliable Tx Confirmation is not implemented. Instead the old behavior from ASR 4.0.3 is used.

  Rationale:

  Feature not needed at the moment

  Requirements:

  SWS_IpduM_00189, SWS_IpduM_00044

- MultiplexFunctionalChange

  Description:

  Functional Requirement changes in R20-11

  Rationale:

  Changed or new requirements

  Requirements:

  SWS_IpduM_00173

- IpduM_Transmit_ServiceId

  Description:

  The service Id for API IpduM_Transmit has changed in R20-11

  Rationale:

  Changed or new requirements

  Requirements:

  SWS_IpduM_00043

- IpduMContainedTxPduHandleId_PB

  Description:

  IpduMContainedTxPduHandleId is postbuild.

  Rationale:

  IpduMContainedTxPduHandleId is part of the postbuild strategy.

  Requirements:

  ECUC_IpduM_00179
- IpduMContainerRxHandleId_PB

  Description:

  Parameter IpduMContainerRxHandleId shall be postbuild.

  Rationale:

  IpduMContainerRxHandleId is part of the postbuild strategy.

  Requirements:

  ECUC_IpduM_00187
- IpduMContainerTxHandleId_PB

  Description:

  Parameter IpduMContainerTxHandleId shall be postbuild.

  Rationale:

  IpduMContainerTxHandleId is part of the postbuild strategy.

  Requirements:

  ECUC_IpduM_00191
- IpduMContainedPduHeaderId_Multiplicity

  Description:

  Lower Multiplicity for parameter IpduMContainedPduHeaderId has changed from 1 to 0.

  Rationale:

  Changed or new requirements

Requirements:

ECUC_IpduM_00172

- InitFailed

  Description:

  Error code in R20-11 not implemented.

  Rationale:

  Not implemented.

  Requirements:

  SWS_IpduM_00174

- ContainerHeaderId

  Description:

  R20-11 feature regarding Header Ids which are part of the ContainerPdu when the ContainedPdu is inside.

  Rationale:

  Feature not implemented.

  Requirements:

  ECUC_IpduM_00203, ECUC_IpduM_00202

- Metadata

  Description:

  Metadata functionality which is not yet part of the implementation.

  Rationale:

  Feature not implemented.

  Requirements:

  SWS_IpduM_00228, SWS_IpduM_00226, SWS_IpduM_00225, SWS_IpduM_00230

- Partly implemented functionalities

  Description:

  The listed changed requirements are partly not support and can be requested on demand.

Rationale:

Requirements:

SWS_IpduM_00190, SWS_IpduM_00231, ECUC_IpduM_00158, ECUC_IpduM_00162, ECUC_IpduM_00142

- IpduMMaxTxBufferSize

  Description:

  The functionality behind the parameter IpduMMaxTxBufferSize was already implemented by EB but in a different form.

  Rationale:

  Functionality already implemented through parameter IpduMDataMemSize

  Requirements:

  IpduM.ECUC_IpduM_00166

- IpduMMetaDataSupport lower multiplicity

  Description:

  The lower multiplicity of parameter IpduMMetaDataSupport is set to 1.

  Rationale:

  This is a general parameter and always present.

  Requirements:

  IpduM.ECUC_IpduM_00205

- EB implementation on flex processing and multicore distribution

  Description:

  The flexible processing and multicore distribution was implemented in IpduM before it was specified in the SWS. Therefore, while the core functionality respects the SWS, some details are not exactly identical (ex. name of config. parameters).

  Rationale:

  Functionality already implemented.

  Requirements:

IpduM.ECUC_IpduM_00211, IpduM.ECUC_IpduM_00212, IpduM.ECUC_IpduM_00213, IpduM.ECUC_IpduM_00214, IpduM.ECUC_IpduM_00215, IpduM.ECUC_IpduM_00216 IpduM.ECUC_IpduM_00217, IpduM.ECUC_IpduM_00218

- IpduMRxSelectorValue upper multiplicity

  Description:

  The upper multiplicity of parameter IpduMRxSelectorValue is set to 255.

  Rationale:

  Memory consumption improvement.

  Requirements:

  IpduM.ECUC_IpduM_00113

## 2.5. Limitations

This chapter lists the limitations of the module. Refer to the module references chapter *Integration notes*, subsection *Integration requirements* for requirements on integrating this module.

- IpduM Handle ID assignment

  Description:

  - `IpduMConfig/IpduMRxPathway/*/IpduMRxIndication/IpduMRxHandleId` has to be assigned zero-based and dense.
  - `IpduMConfig/IpduMTxPathway/*/IpduMTxRequest/IpduMTxStaticPart/IpduMTxStaticHandleId` has to be assigned zero-based and dense.
  - `IpduMConfig/IpduMTxPathway/*/IpduMTxRequest/IpduMTxDynamicPart/*/IpduMTxDynamicHandleId` has to be assigned dense and start with `max(IpduMTxStaticHandleId)+1`.
  - `IpduMConfig/IpduMContainedTxPdu/*/IpduMContainedTxPduHandleId` has to be assigned dense and start from the last `IpduMTxPathway/IpduMTxRequest/IpduMTxDynamicPart/IpduMTxDynamicHandleId`.
  - `IpduMConfig/IpduMContainerTxPdu/*/IpduMContainerTxHandleId` has to be assigned dense and start from the number of `IpduMTxPathways`.

  - IpduMConfig/IpduMContainerRxPdu/*/IpduMContainerRxHandleId has to be assigned dense and start from the last IpduMRxPathway/IpduMRxHandleId, IpduMContainerRxPdus with IpduMContainerPduProcessing configured as IPDUM_PROCESSING_DEFERRED first.

  Rationale:

  Code size reduction and run-time improvement.

- IpduM module expects restricted multiplicity of container PduRRoutingTable

  Description:

  The IpduM module expects the upper multiplicity of container PduRRoutingTable restricted to 1.

  Rationale:

  The post-build time loadable and selectable concepts are supported through variant handling, where multiple routing tables are not applied.

- Configuration separation

  Description:

  In the current version the configuration is generated in a monolithic manner, not separately for each core.

- Uniqueness of contained PDU header IDs

  Description:

  The IpduMContainedPduHeaderIds must be unique for contained Tx PDUs in the context of the associated IpduMContainerTxPdu.

  The IpduMContainedPduHeaderIds must be unique for contained Rx PDUs if IpduMGeneral/IpduMRxContainerAcceptAllNoRefOnly is FALSE.

- Rx/Tx PathWay PduLengthType

  Description:

  The PduLengthType is limited to uint16 for Rx and Tx PathWays. Even though the PduLengthType can be set to uint32 the IpduM module does not offer support for Rx and Tx PathWay PDUs of size greater than uint16.

- Contained PDUs with length 0 not forwarded

  Description:

  The IpduM module silently drops Contained PDUs with an SduLength of 0.

  Rationale:

The SWS does not define how to handle PDUs with a length of 0. During transmission the call to `IpduM_Transmit()` is ignored and the value `E_OK` is returned.

- Metadata on Container Transmission

  Description:

  On the container transmission side, only metadata of type CAN_ID_32 is currently supported.

  Rationale:

  Currently limited for efficiency reasons.

## 2.6. Open-source software

IpduM does not use open-source software.