AUTOSAR PduR User Manual

Document Change Histroy			
Date (YYYY-MM-DD)	**Ver.**	**Editor**	**Content**
2020-12-04	1.0.0.0	Autoever	PduR UM Initial Version
2021-07-08	1.0.1.0	PhucNHM	R44 migration release
2021-09-15	1.0.1.1	PhucNHM	Applying change of company name
2021-09-22	1.0.1.2	PhucNHM	ASPICE compliance update
2021-11-11	1.0.2.0	PhucNHM	Fix defects and update template
2022-03-04	1.0.3.0	PhucNHM	Support shared gateway buffer features
2022-06-27	1.0.3.1	PhucNHM	Editorial Changes of Work Products
2022-08-15	1.0.4.0	PhucNHM	Fix UNECE polyspace MISRA & CERT-C violations
2022-11-11	1.0.5.0	PhucNHM	Support routing for CDD_Router
2022-12-05	1.0.6.0	HaoTA4	Fix generate handle ID when turn on post build support for PduR
2022-12-14	1.0.7.0	HaoTA4	Fix no define Start section and Duplicated Start Sections MemMap error when applying variant for PduR
2022-12-16	1.0.8.0	HaoTA4	Fix SW version in PduR_Version.h
2023-08-22	1.0.9.0	HaoTA4	Add validation rules for detect gateway N:1 without configure buffer Update UM template
2024-05-24	1.0.9.0_HF1	MJ Kim	Shared buffer not supported Add validation rules to detect PduRDestPduRef and duRSrcPduRef unique or not. Add validation rules to detect that the SharedGatewayIfBuffer is false but the buffer is not unique.

Table of Contents

1. Overview

It is written based on Autosar standard SRS / SWS, and if more detailed functional description is needed when using the module, refer to the reference document below.

The interpretation of the category related to setting is as follows.

- Changeable (C) : Item that can be set by the user
- Fixed (F) : Item that can not be changed by the user
- Not Supported (N) : Deprecated item

2. Reference

Sl. No.	Title	Version
1	AUTOSAR_SWS_PDURouter.pdf	4.4.0
2		
3		

3. AUTOSAR System

3.1 Communication Stack

In the Hyundai AUTOEVER AUTOSAR platform, the Communication Stack consists of the detailed module below

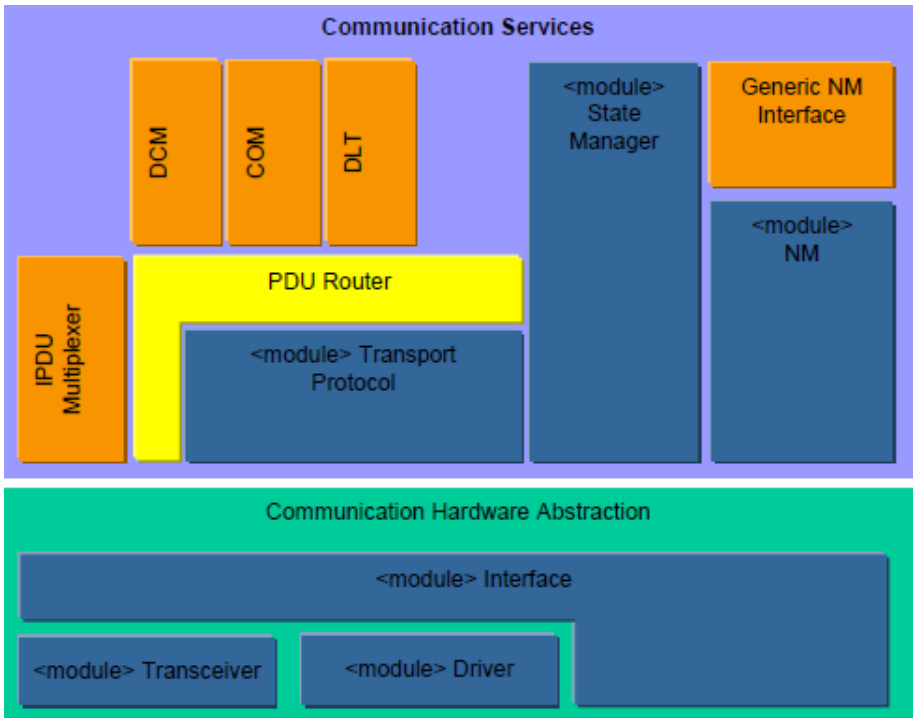
- CanIf : Sending and receiving CAN messages
- PduR : In charge of PDU transmission between communication modules
- IpduM : Reponsible for sending and receiving multiplxed PDUs
- CanTp : Respinsible for large scale data transmission and reception based on transport protocol
- Com : Sending and receiving I-PDU to Pdu Router, handle content of I-PDU and provides signal to RTE
- Dcm : Sending and receiving I-PDU to Pdu Router, handle diagnostics messages

3.2 PduR Module

The PDU Router module provides services for routing of I-PDUs (Interaction Layer Protocol Data Units) using the following module types:

- Communication interface modules, that are modules that use the <Provider:Up> or <Provider:Lo> API, e.g. Com, IPduM, LinIf, CanIf, CanNm, FrIf and FrNm

- Transport Protocol modules, that are modules using the <Provider:UpTp> or <Provider:LoTp> API, e.g. J1939Tp, LinTp (part of LinIf), CanTp, FrTp, COM, DCM



4. Limitations and Deviations

4.1 Limitations

PduRSharedGatewayIfBuffer is not supported.

4.2 Deviations

- PduRRxInterruptBased, PduRTxInterruptBased

  Feature added to process unnecessary interrupts

5. Configuration Guide

5.1 PduRBswModules

Parameter Name	Value	Category
[1] PduRCancelReceive	True/False	C
[2] PduRCancelTransmit	True/False	C
[3] PduRCommunicationInterface	True/False	C
[4] PduRLowerModule	True/False	C
[5] PduRRetransmitssion	True/False	C
[6] PduRTransportProtocol	True/False	C
[7] PduRTriggertransmit	True/False	C
[8] PduRTxConfirmation	True/False	C
[9] PduRUpperModule	True/False	C
[10] PduRUseTag	True/False	C
[11] PduRBswModuleRef	True/False	C

1) PduRCancelReceive:
   - Provides functions for processing cancel receive. (PduR_<User:Up>CancelReceive)

2) PduRCancelTransmit
   - Provides functions for processing cancel transmission. (PduR_<User:Up>CancelTransmit)

3) PduRCommunicationInterface
   - Provides functions for processing communication interface (PduR_<User:Up>Transmit, PduR_<User:Up>CancelTransmit, PduR_<User:Lo>RxIndication, PduR_<User:Lo>TxConfirmation)

4) PduRLowerModule
   - The PduRLowerModule will decide who will call the APIs and who will implement the APIs.

5) PduRRetransmitssion
   - If set to true this means that the destination transport protocol module will use the retransmission feature.
   - This parameter might be set to false if the retransmission feature is not used, even though the destination transport protocol is supporting it.

6) PduRTransportProtocol
   - Provides function for processing transport protocol.

7) PduRTriggertransmit
   - Provides function for processing trigger transmission. (PduR_<User:Lo>TriggerTransmit)

8) PduRTxConfirmation
   - Provides function for processing transmit confirmation (PduR_<User:Lo>TxConfirmation)

9) PduRUpperModule
   - The PduRUpperModule will decide who will call the APIs and who will implement the APIs.

10) PduRUseTag
   - This parameter, if set to true, enables the usage of the tag (<up>) in the following API calls: * PduR_<Up>CancelReceive, * PduR_<Up>CancelTransmit

11) PduRBswModuleRef
   - This is a reference to one BSW module's configuration.

5.2 PduRGeneral

Parameter Name	Value	Category
[1] PduRDevErrorDetect	True/False	C
[2] PduRMetaDataSupport	True/False	C
[3] PduRVersionInfoApi	True/False	C
[4] PduRZeroCostOperation	True/False	C
[5] PduRTxInterruptBased	True/False	C
[6] PduRRxInterruptBased	True/False	C
[7] PduRHeaderFileInclusion	-	C
[8] PduRSharedGatewayIfBuffer	True/False	N

1) PduRDevErrorDetect
   - Enable Det Error detect feature.

2) PduRMetaDataSupport
   - Enable handle MetaData feature in I-PDU.

3) PduRVersionInfoApi
   - Provides PduR_GetVersionInfo() function

4) PduRZeroCostOperation
   - Enable Zero Cost feature.

5) PduRTxInterruptBased
6) PduRRxInterruptBased
   - A feature to remove unnecessary interrupts.

   - Set to false by default; when optimization is required due to the use of gateway functionality, may be set to true.

7) PduRHeaderFileInclusion
   - Define necessary header file to use CallOut function configured per PduRRoutingPath.

8) PduRSharedGatewayIfBuffer
   - Features developed for Memory Saving.

5.3 PduRRoutingPaths

Parameter Name	Value	Category
[1] PduRConfigurationId	0 … 65535	C
[2] PduRMaxRoutingPathCnt	0 … 65535	C
[3] PduRMaxRoutingPathGroupCnt	0 … 65535	C

1) PduRConfigurationId
   - Identification of the configuration of the PduR configuration.
   - This identification can be read using the PduR API.

2) PduRMaxRoutingPathCnt
   - Maximum number of RoutingPaths in all RoutingTables.
   - This parameter is needed only in case of post-build loadable implementation using static memory allocation.

3) PduRMaxRoutingPathGroupCnt

- Maximum number of RoutingPathGroups.
- This parameter is needed only in case of post-build loadable implementation using static memory allocation.

5.3.1 PduRDestPdu

Parameter Name	Value	Category
1) PduRDestPduDataProvision	-	C
2) PduRTransmissionConfirmation	True/False	C
3) PduRDestPduHandleId	0 … 65535	C
4) PduRDestPduRef	-	C

1) PduRDestPduDataProvision
   - Specifies how data are provided: direct (as part of the Transmit call) or via the TriggerTransmit callback function.
   - Only required for non-TP gatewayed I-PDUs.
   - PDUR_DIRECT : The PDU Router module shall call the transmit function in the destination module and not buffer the I-PDU
   - PDUR_TRIGGERTRANSMIT : The PDU Router module shall call the transmit function in the destination module. The destination module will request the I-PDU using the triggerTransmit function. The I-PDU is shall be buffered.

2) PduRTransmissionConfirmation
   - This parameter is only for communication interfaces.
   - Transport protocol modules will always call the TxConfirmation function.

3) PduRDestPduHandleId
   - PDU identifier assigned by PDU Router.
   - Used by communication interface and transport protocol modules for confirmation (PduR_<Lo>TxConfirmation) and for TriggerTransmit (PduR_<Lo>TriggerTransmit).

4) PduRDestPduRef
   - Destination PDU reference; reference to unique PDU identifier which shall be used by the PDU Router instead of the source PDU ID when calling the related function of the destination module.

5.3.2 PduRRoutingPath

Parameter Name	Value	Category
1) PduRRxIndUserCallOut	-	C
2) PduRQueueDepth	1 … 255	C
3) PduRTpThreshold	0 … 65535	C
4) PduRDestPduRRef	-	C
5) PduRDestTxBufferRef	-	C
6) PduRRoutingPathGroupRef	-	C
7) PduRSrcPduRRef	-	C

1) PduRRxIndUserCallOut
   - Name of the callout function to be called if PduLoRxIndication() or PduR_LoTpStartOfReception is called.

2) PduRQueueDepth
   - This parameter defines the queue depth for this routing path.

3) PduRTpThreshold
   - This parameter is only relevant for TP routings.

4) PduRDestPduRRef
   - Destination PduR reference.

5) PduRDestTxBufferRef
   - Reference to a buffer in the PduR.
   - This buffer is required for communication interface gatewaying, and for transport protocol gatewaying.

6) PduRRoutingPathGroupRef
   - Reference to routing path destinations.

7) PduRSrcPduRRef
   - Source PduR reference.

5.3.2.1 PduRDefaultValue

5.3.2.1.1 PduRDefaultValueElement

Parameter Name	Value	Category
[1] PduRDefaultValueElement	0 … 255	C
[1] PduRDefaultValueElementBytePosition	0 … 4294967294	C

1) PduRDefaultValueElement
   - The default value consists of a number of elements.
   - Each element is one byte long and the number of elements is specified by SduLength.
   - The position of this parameter in the container is specified by the PduRElementBytePosition parameter.

2) PduRDefaultValueElementBytePosition
   - This parameter specifies the byte position of the element within the default value.

5.3.3 PduRRoutingPathGroup

Parameter Name	Value	Category
[1] PduRIsEnabledAtInit	True/False	C
[2] PduRRoutingPathGroupId	0 … 65535	C

1) PduRIsEnabledAtInit
   - If set to true this routing path group will be enabled after initializing the PDU Router module (i.e. enabled in the PduR_Init function).

2) PduRRoutingPathGroupId
   - Identification of the routing group.

5.3.4 PduRSrcPdu

Parameter Name	Value	Category
[1] PduRSrcPduUpTxConf	True/False	C
[2] PduRSourcePduBlockSize	1 … 4294967295	C
[3] PduRSourcePduHandleId	0 … 65535	C
[4] PduRSrcPduRef	-	C

1) PduRSrcPduUpTxConf
   - When enabled, the TxConfirmation will be forwarded to the upper layer.
   - Prerequisites: Lower layer and upper layer support TxConfirmation.

2) PduRSourcePduBlockSize
   - Minimum amount of buffer space required by receiving transport protocol layer to continue reception.

3) PduRSourcePduHandleId
   - PDU identifier assigned by PDU Router.

4) PduRSrcPduRef
   - Source PDU reference; reference to unique PDU identifier which shall be used for the requested PDU Router operation.

5.3.5 PduRTxBuffer

Parameter Name	Value	Category
[1)] PduRPduMaxLength	1 … 4294967295	C

1) PduRPduMaxLength
   - Length of the Tx buffer in bytes.
   - This parameter limits the size of buffered routed PDUs.

6. Application Programming Interface (API)

6.1 Type Definitions

6.1.1 PduR_PBConfigType

Name:	PduR_PBConfigType	
Type:	Structure	
Range:	--	implementation specific
Description:	Data structure containing post-build-time configuration data of the PDU Router.	
Available via:	PduR.h	

6.1.2 PduR_PBConfigIdType

Name:	PduR_PBConfigIdType
Type:	Uint16
Description:	Identification of the post-build configuration currently used for routing I-PDUs. An ECU may contain several configurations (post-build selectable), each have unique Id.
Available via:	PduR.h

6.1.3 PduR_RoutingPathGroupIdType

Name:	PduR_RoutingPathGroupIdType
Type:	Uint16
Description:	Identification of a Routing Table
Available via:	PduR.h

6.1.4 PduR_StateType

Name:	PduR_StateType		
Type	Enumeration		
Range:	PDUR_UNINIT PDUR_ONLINE	-- --	PDU Router not initialized PDU Router initialized successfully
Description:	States of the PDU Router.		
Available via:	PduR.h		

6.1.5 PduR_GwBufferStatus

Name:	PduR_GwBufferStatus		
Type	Structure		
Element:	uint8	ucNoOfRoutePdu	--
	uint8	aaPathStatus[15]	--
	uint8	aaBufStatus[15]	--
Description:	PDU Router gateway buffer status		

Available via:	PduR.h

6.2 Macro Constants

None

6.3 Functions

6.3.1 PduR_Init

Service name:	PduR_Init	
Syntax:	void PduR_Init(const PduR_ConfigTye* ConfigPtr)	
Service ID [hex]:	0xf0	
Sync/Async:	Synchronous	
Reentrancy:	Non-Reentrant	
Parameters (in):	ConfigPtr	Pointer to post build configuration
Parameters (inout):	None	
Parameters (out):	None	
Return Value:	None	
Description:	Initializes the PDU Router. This function is used by BSW.	
Available via:	PduR.h	

6.3.2 PduR_GetVersionInfo

Service name:	PduR_GetVersionInfo	
Syntax:	void PduR_GetVersionInfo(Std_VersionInfoType* versionInfo)	
Service ID[hex]:	0xf1	
Sync/Async:	Asynchronous	
Reentrancy:	Reentrant	
Parameters (in):	None	
Parameters (inout):	None	
Parameters (out):	versionInfo	Pointer to where to store the version information of this module.
Return Value:	None	
Description:	Returns the version information of this module. This function is used by user. But it needs configuration. (It cannot be called directly by user)	
Available via:	PduR.h	

6.3.3 PduR_GetConfigurationId

Service name:	PduR_GetConfigurationId
Syntax:	PduR_PBConfigIdType PduR_GetConfigurationId(void)
Service ID[hex]:	0xf2
Sync/Async:	Synchronous
Reentrancy:	Reentrant

Parameters (in):	None	
Parameters (inout):	None	
Parameters (out):	None	
Return Value:	PduR_PBConfigIdType	Identifier of the post-build time configuration
Description:	Returns the unique identifier of the post-build time configuration of the PDU Router. This function is used by user. But it needs configuration. (It cannot be called directly by user.)	
Available via:	PduR.h	

6.3.4 PduR_EnableRouting

Service name:	PduR_EnableRouting	
Syntax:	void PduR_EnableRouting(PduR_RoutingPathGroupIdType id)	
Service ID[hex]:	0xf3	
Sync/Async:	Synchronous	
Reentrancy:	Reentrant	
Parameters (in):	id	Identification of the routing path group. Routing path groups are defined in the PDU router configuration.
Parameters (inout):	None	
Parameters (out):	None	
Return Value:	None	
Description:	Enables a routing path group. This function is used by user. But it needs configuration. (It cannot be called directly by user)	
Available via:	PduR.h	

6.3.5 PduR_DisableRouting

Service name:	PduR_DisableRouting	
Syntax:	void PduR_DisableRouting(PduR_RoutingPathGroupIdType id, boolean initialize)	
Service ID[hex]:	0xf4	
Sync/Async:	Synchronous	
Reentrancy:	Reentrant	
Parameters (in):	id	Identification of the routing path group. Routing path groups are defined in the PDU router configuration.
	initialize	true: initialize single buffers to the default value false: retain current value of single buffers
Parameters (inout):	None	
Parameters (out):	None	
Return Value:	None	
Description:	Disables a routing path group. This function is used by user. But it needs configuration. (It cannot be called directly by user)	
Available via:	PduR.h	

6.3.6 PduR_<User:Up>Transmit

Service name:	PduR_<User:Up>Transmit
Syntax:	Std_ReturnType PduR_<User:Up>Transmit(PduIdType TxPduId, const PduInfoType* PduInfoPtr)
Service ID[hex]:	0x49

Sync/Async:	Synchronous	
Reentrancy:	Reentrant for different PduIds. Non reentrant for the same PduId.	
Parameters (in):	TxPduId	Identifier of the PDU to be transmitted
	PduInfoPtr	Length of and pointer to the PDU data and pointer to MetaData.
Parameters (inout):	None	
Parameters (out):	None	
Return Value:	Std_ReturnType	E_OK: Transmit request has been accepted. E_NOT_OK: Transmit request has not been accepted.
Description:	Requests transmission of a PDU. This function is used by BSW.	
Available via:	PduR_<module>.h	

6.3.7 PduR_<User:Up>CancelTransmit

Service name:	PduR_<User:Up>CancelTransmit	
Syntax:	Std_ReturnType PduR_<User:Up>CancelTransmit(PduIdType TxPduId)	
Service ID[hex]:	0x4a	
Sync/Async:	Synchronous	
Reentrancy:	Reentrant for different PduIds. Non reentrant for the same PduId.	
Parameters (in):	TxPduId	Identification of the PDU to be cancelled.
Parameters (inout):	None	
Parameters (out):	None	
Return Value:	Std_ReturnType	E_OK: Cancellation was executed successfully by the destination module. E_NOT_OK: Cancellation was rejected by the destination module.
Description:	Requests cancellation of an ongoing transmission of a PDU in a lower layer communication module. This function is used by BSW.	
Available via:	PduR_<module>.h	

6.3.8 PduR_<User:Up>CancelReceive

Service name:	PduR_<User:Up>CancelReceive	
Syntax:	Std_ReturnType PduR_<User:Up>CancelReceive(PduIdType RxPduId)	
Service ID[hex]:	0x4c	
Sync/Async:	Synchronous	
Reentrancy:	Non Reentrant	
Parameters (in):	RxPduId	Identification of the PDU to be cancelled.
Parameters (inout):	None	
Parameters (out):	None	
Return Value:	Std_ReturnType	E_OK: Cancellation was executed successfully by the destination module. E_NOT_OK: Cancellation was rejected by the destination module.
Description:	Requests cancellation of an ongoing reception of a PDU in a lower layer transport protocol module. This function is used by BSW.	
Available via:	PduR_<module>.h	

6.3.9 PduR_<User:Lo>RxIndication

Service name:	PduR_<User:Lo>RxIndication	
Syntax:	void PduR_<User:Lo>RxIndication(PduIdType RxPduId, const PduInfoType* PduInfoPtr)	
Service ID[hex]:	0x42	
Sync/Async:	Synchronous	
Reentrancy:	Reentrant for different PduIds. Non reentrant for the same PduId.	
Parameters (in):	RxPduId	ID of the received PDU.
	PduInfoPtr	Contains the length (SduLength) of the received PDU, a pointer to a buffer (SduDataPtr) containing the PDU, and the MetaData related to this PDU.
Parameters (inout):	None	
Parameters (out):	None	
Return Value:	None	
Description:	Indication of a received PDU from a lower layer communication interface module. This function is used by BSW.	
Available via:	PduR_<module>.h	

6.3.10 PduR_<User:Lo>TxConfirmation

Service name:	PduR_<User:Lo>TxConfirmation	
Syntax:	void PduR_<User:Lo>TxConfirmation(PduIdType TxPduId, Std_ReturnType result)	
Service ID[hex]:	0x40	
Sync/Async:	Synchronous	
Reentrancy:	Reentrant for different PduIds. Non reentrant for the same PduId	
Parameters (in):	TxPduId	ID of the PDU that has been transmitted.
	result	E_OK: The PDU was transmitted. E_NOT_OK: Transmission of the PDU failed.
Parameters (inout):	None	
Parameters (out):	None	
Return Value:	None	
Description:	The lower layer communication interface module confirms the transmission of a PDU, or the failure to transmit a PDU. This function is used by BSW.	
Available via:	PduR_<module>.h	

6.3.11 PduR_<User:Lo>TriggerTransmit

Service name:	PduR_<User:Lo>TriggerTransmit	
Syntax:	Std_ReturnType PduR_<User:Lo>TriggerTransmit(PduIdType TxPduId, PduInfoType* PduInfoPtr)	
Service ID[hex]:	0x41	
Sync/Async:	Synchronous	
Reentrancy:	Reentrant for different PduIds. Non reentrant for the same PduId.	
Parameters (in):	TxPduId	ID of the SDU that is requested to be transmitted.
Parameters (inout):	PduInfoPtr	Contains a pointer to a buffer (SduDataPtr) to where the SDU data shall be copied, and the available buffer size in SduLengh. On return, the service will indicate the length of the copied SDU data in SduLength.
Parameters (out):	None	
Return Value:	Std_ReturnType	E_OK: SDU has been copied and SduLength indicates the number of copied bytes. E_NOT_OK: No SDU data has been

		copied. PduInfoPtr must not be used since it may contain a NULL pointer or point to invalid data.
Description:	Within this API, the upper layer module (called module) shall check whether the available data fits into the buffer size reported by PduInfoPtr->SduLength. If it fits, it shall copy its data into the buffer provided by PduInfoPtr->SduDataPtr and update the length of the actual copied data in PduInfoPtr->SduLength. If not, it returns E_NOT_OK without changing PduInfoPtr. This function is used by BSW.	
Available via:	PduR_<module>.h	

6.3.12 PduR_<User:LoTp>CopyRxData

Service name:	PduR_<User:LoTp>CopyRxData	
Syntax:	BufReq_ReturnType PduR_<User:LoTp>CopyRxData(PduIdType id, const PduInfoType* info, PduLengthType* bufferSizePtr)	
Service ID[hex]:	0x44	
Sync/Async:	Synchronous	
Reentrancy:	Reentrant	
Parameters (in):	id	Identification of the received I-PDU.
	info	Provides the source buffer (SduDataPtr) and the number of bytes to be copied (SduLength). An SduLength of 0 can be used to query the current amount of available buffer in the upper layer module. In this case, the SduDataPtr may be a NULL_PTR.
Parameters (inout):	None	
Parameters (out):	bufferSizePtr	Available receive buffer after data has been copied.
Return Value:	BufReq_ReturnType	BUFREQ_OK: Data copied successfully BUFREQ_E_NOT_OK: Data was not copied because an error occurred.
Description:	This function is called to provide the received data of an I-PDU segment (N-PDU) to the upper layer. Each call to this function provides the next part of the I-PDU data. The size of the remaining buffer is written to the position indicated by bufferSizePtr. This function is used by BSW.	
Available via:	PduR_<module>.h	

6.3.13 PduR_<User:LoTp>RxIndication

Service name:	PduR_<User:LoTp>RxIndication	
Syntax:	void PduR_<User:LoTp>RxIndication(PduIdType id, Std_ReturnType result)	
Service ID[hex]:	0x45	
Sync/Async:	Synchronous	
Reentrancy:	Reentrant	
Parameters (in):	Id	Identification of the received I-PDU.
	result	Result of the reception.
Parameters (inout):	None	
Parameters (out):	None	
Return Value:	None	
Description:	Called after an I-PDU has been received via the TP API, the result indicates whether the transmission was successful or not.	

	This function is used by BSW.
Available via:	PduR_<module>.h

6.3.14 PduR_<User:LoTp>StartOfReception

Service name:	PduR_<User:LoTp>StartOfReception	
Syntax:	BufReq_ReturnType PduR_<User:LoTp>StartOfReception(PduIdType id, const PduInfoType* info, PduLengthType TpSduLength, PduLengthType* bufferSizePtr)	
Service ID[hex]:	0x46	
Sync/Async:	Synchronous	
Reentrancy:	Reentrant	
Parameters (in):	Id	Identification of the I-PDU.
	info	Pointer to a PduInfoType structure containing the payload data (without protocol information) and payload length of the first frame or single frame of a transport protocol I-PDU reception, and the MetaData related to this PDU. If neither first/single frame data nor MetaData are available, this parameter is set to NULL_PTR.
	TpSduLength	Total length of the N-SDU to be received.
Parameters (inout):	None	
Parameters (out):	bufferSizePtr	Available receive buffer in the receiving module. This parameter will be used to compute the Block Size (BS) in the transport protocol module.
Return Value:	BufReq_ReturnType	BUFREQ_OK: Connection has been accepted. bufferSizePtr indicates the available receive buffer; reception is continued. If no buffer of the requested size is available, a receive buffer size of 0 shall be indicated by bufferSizePtr. BUFREQ_E_NOT_OK: Connection has been rejected; reception is aborted. bufferSizePtr remains unchanged. BUFREQ_E_OVFL: No buffer of the required length can be provided; reception is aborted. bufferSizePtr remains unchanged.
Description:	This function is called at the start of receiving an N-SDU. The N-SDU might be fragmented into multiple N-PDUs (FF with one or more following CFs) or might consist of a single N-PDU (SF). The service shall provide the currently available maximum buffer size when invoked with TpSduLength equal to 0. This function is used by BSW.	
Available via:	PduR_<module>.h	

6.3.15 PduR_<User:LoTp>CopyTxData

Service name:	PduR_<User:LoTp>CopyTxData	
Syntax:	BufReq_ReturnType PduR_<User:LoTp>CopyTxData(PduIdType id, const PduInfoType* info, const RetryInfoType* retry, PduLengthType* availableDataPtr)	
Service ID[hex]:	0x43	
Sync/Async:	Synchronous	
Reentrancy:	Reentrant	
Parameters (in):	Id	Identification of the transmitted I-PDU.
	info	Provides the destination buffer (SduDataPtr) and the number of bytes to be copied (SduLength). If not enough transmit data

		is available, no data is copied by the upper layer module and BUFREQ_E_BUSY is returned. The lower layer module may retry the call. An SduLength of 0 can be used to indicate state changes in the retry parameter or to query the current amount of available data in the upper layer module. In this case, the SduDataPtr may be a NULL_PTR.
	retry	This parameter is used to acknowledge transmitted data or to retransmit data after transmission problems. If the retry parameter is a NULL_PTR, it indicates that the transmit data can be removed from the buffer immediately after it has been copied. Otherwise, the retry parameter must point to a valid RetryInfoType element. If TpDataState indicates TP_CONFPENDING, the previously copied data must remain in the TP buffer to be available for error recovery. TP_DATACONF indicates that all data that has been copied before this call is confirmed and can be removed from the TP buffer. Data copied by this API call is excluded and will be confirmed later. TP_DATARETRY indicates that this API call shall copy previously copied data in order to recover from an error. In this case TxTpDataCnt specifies the offset in bytes from the current data copy position.
Parameters (inout):	None	
Parameters (out):	availableDataPtr	Indicates the remaining number of bytes that are available in the upper layer module's Tx buffer. availableDataPtr can be used by TP modules that support dynamic payload lengths (e.g. FrIsoTp) to determine the size of the following CFs.
Return Value:	BufReq_ReturnType	BUFREQ_OK: Data has been copied to the transmit buffer completely as requested. BUFREQ_E_BUSY: Request could not be fulfilled, because the required amount of Tx data is not available. The lower layer module may retry this call later on. No data has been copied. BUFREQ_E_NOT_OK: Data has not been copied. Request failed.
Description:	This function is called to acquire the transmit data of an I-PDU segment (N-PDU). Each call to this function provides the next part of the I-PDU data unless retry->TpDataState is TP_DATARETRY. In this case the function restarts to copy the data beginning at the offset from the current position indicated by retry->TxTpDataCnt. The size of the remaining data is written to the position indicated by availableDataPtr. This function is used by BSW.	
Available via:	PduR_<module>.h	

6.3.16 PduR_<User:LoTp>TxConfirmation

Service name:	PduR_<User:LoTp>TxConfirmation	
Syntax:	void PduR_<User:LoTp>TxConfirmation(PduIdType id, Std_ReturnType result)	
Service ID[hex]:	0x48	
Sync/Async:	Synchronous	
Reentrancy:	Reentrant	
Parameters (in):	id	Identification of the transmitted I-PDU.

	result	Result of the transmission of the I-PDU.
Parameters (inout):	None	
Parameters (out):	None	
Return Value:	None	
Description:	This function is called after the I-PDU has been transmitted on its network, the result indicates whether the transmission was successful or not. This function is used by BSW.	
Available via:	PduR_<module>.h	

6.3.17 PduR_GetGwBufferStatus

Service name:	PduR_GetGwBufferStatus	
Syntax:	void PduR_GetGwBufferStatus(PduIdType LddPduRSrcPduId, PduR_GwBufferStatus * LpGwStatus)	
Service ID[hex]:	0xf5	
Sync/Async:	Synchronous	
Reentrancy:	Reentrant	
Parameters (in):	LddPduRSrcPduId	Identification of the received gateway I-PDU.
Parameters (inout):	None	
Parameters (out):	LpGwStatus	Provide gateway buffer status
Return Value:	None	
Description:	This function is called by the user in case of gateway to check buffer is available. This function Is used by user. But it needs configuration. (It cannot be called directly by user)	
Available via:	PduR.h	

6.4 Scheduled functions

None

7. Generator

Option	Description
-G,--Generation	Symbolic parameters to be used for fore generation (skip validation).
-H,--Help	Display this help message.
-I,--Input <I>	ECU description file path of the module for which generation tool need to run.
-L,--Log	Symbolic parameters to be used for generation error log.
-M,--Module <M>	Specify module name and version to be generated code for.
-T,--Top_path <T>	Symbolic parameters to be used for set path of module.
-V,--Validate	Symbolic parameters to be used for invoking validation checks.
-O, --Output <O>	Project-relative path to location where the generated code is to be placed.

7.1 Generator Message

7.1.1 Error Messages

7.1.1.1 ERR051051: Container 'PduRRoutingPath' should be configured.
This error occurs, if container PduRRoutingPath is not configured.

7.1.1.2 ERR051052: Destination path 'path' configured for the parameter 'PduRDestPduRef' in the container 'PduRRoutingPath's name' is not matching with the destination module within the configuration set <'PduRRoutingPaths's name'>
This error occurs, if path of PduRDestPduRef is configured wrong, can not match with path of destination module.

7.1.1.3 ERR051053: Destination path 'path' configured for the parameter 'PduRSrcPduRef' in the container 'PduRRoutingPath's name' is not matching with the source module within the configuration set <'PduRRoutingPaths's name'>
This error occurs, if path of PduRSrcPduRef is configured wrong, can not match with path of source module.

7.1.1.4 ERR051057: Handle Id(s) < 'List PduRSourcePduHandleId' > configured for the parameter 'PduRSourcePduHandleId' in the container ' PduRSrcPdu's name' for the Source Module(s) <'List source module'> as Tp RoutingPath sequence should be unique within the configuration set <'PduRRoutingPaths's Name'>.
This error occurs, if PduRSourcePduHandleId of PduRSrcPdu is not configured unique in list LoTp module.

7.1.1.5 ERR051058: Handle Id(s) <'List PduRSourcePduHandleId'> configured for the parameter 'PduRSourcePduHandleId' in the container 'PduRSrcPdu's name' for the Source Module(s) <'List source module'> should start with <0> and should be sequential within the configuration set <'PduRRoutingPaths's Name'>.
This error occurs, if PduRSourcePduHandleId of PduRSrcPdu is not configured sequential in list LoTp.

7.1.1.6 ERR051060: Handle Id(s) <'List PduRSourcePduHandleId'> configured for the parameter 'PduRSourcePduHandleId' in the container 'PduRSrcPdu's name' for the Source Module(s) <'List source module'> should start with <0> and should be sequential within the configuration set <'PduRRoutingPaths's Name'>.
This error occurs, if PduRSourcePduHandleId of PduRSrcPdu is not configured sequential in list LoIf.

7.1.1.7 ERR051063: Handle Id(s) <'List PduRSourcePduHandleId'> configured for the parameter 'PduRSourcePduHandleId' in the container 'PduRSrcPdu's name' for the Source Module(s) <'List source module'> as UpperLayer Modules(s) either the RoutingPath is Interface or Tp the Id(s) should be unique within the configuration set <'PduRRoutingPaths's Name'>.
This error occurs, if PduRSourcePduHandleId of PduRSrcPdu is not configured unique in list UpIf and UpTp.

7.1.1.8 ERR051064: Handle Id(s) <'List PduRSourcePduHandleId'> configured for the parameter 'PduRSourcePduHandleId' in the container "PduRSrcPdu's name ' for the Source Module(s) <'List source module'> should be sequential within the configuration set <'PduRRoutingPaths's Name'>.
This error occurs, if PduRSourcePduHandleId of PduRSrcPdu is not configured sequential in list UpIf and UpTp.

7.1.1.9 ERR051065: Handle Id(s) <'List PduRDestPduHandleId'> configured for the parameter ' DestPduHandleId ' in the container 'PduRDestPdu's name' for the Destination Module(s) <'List dest module'> as Tp RoutingPath sequence should be unique within the configuration set <'PduRRoutingPaths's Name'>.
This error occurs, if PduRDestPduHandleId of PduRDestPdu is not configured unique in list LoTp.

7.1.1.10 ERR051066: Handle Id(s) <'List PduRDestPduHandleId'> configured for the parameter 'DestPduHandleId' in the container 'PduRDestPdu's name' for the Destination Module(s) <'List dest module'> should start with <0> and should be sequential within the configuration set <'PduRRoutingPaths's Name'>.
This error occurs, if PduRDestPduHandleId of PduRDestPdu is not configured sequential in list LoTp.

7.1.1.11 ERR051067: Handle Id(s) <'List PduRDestPduHandleId'> configured for the parameter 'DestPduHandleId' in the container <'List dest module'> for the Destination Module(s) <'List dest module'> as Interface RoutingPath sequence should be unique with in the configuration set <'PduRRoutingPaths's Name'>.
This error occurs, if PduRDestPduHandleId of PduRDestPdu is not configured unique in list LoIf.

7.1.1.12 ERR051068: Handle Id(s) <'List PduRDestPduHandleId'> configured for the parameter 'DestPduHandleId' in the container <'List dest module'> for the Destination Module(s) <'List dest module'> should start with <0> and should be sequential within the configuration set <'PduRRoutingPaths's Name'>.
This error occurs, if PduRDestPduHandleId of PduRDestPdu is not configured sequential in list LoIf.

7.1.1.13 ERR051070: In case of communication interfaces and gateway operation, the value of the Parameter 'DataProvision' in the container <PduRDestPdu> should be configured when 'PduRDestTxBufferRef' is not empty and 'PduRQueueDepth' configured greater than <1>.
This error occurs when PduRRoutingPath is configured with PduRDestTxBufferRef but PduRDestPduDataProvision is not configured in case of gateway operation interfaces and 'PduRQueueDepth' configured greater than <1>.

7.1.1.14 ERR051073: The Module(s) <'Module's Name'> should be configured in the container 'PduRBswModules' the Module(s) <'Module's Name'> is/are configured in the 'PduRRoutingPath'.
This error occurs when the module is used as src or dest of PduR but PduRBswModuleRef of PduRBswModules is not configured or not refer to it.

7.1.1.15 ERR051074: The ShortName of the Container 'PduRRoutingPathGroup', 'PduRDestPdu' and 'PduRSrcPdu' should be unique within the Configuration Set <'PduRRoutingPaths's Name'>.
This error occurs, if short name of 'PduRRoutingPathGroup' or 'PduRDestPdu' or 'PduRSrcPdu' is not unique

7.1.1.16 ERR051075: RoutPathGrp Id(s) <'List Id'> configured for the parameter 'PduRRoutingPathGroupId' in the container 'PduRRoutingPath' should be unique within the configuration set <'PduRRoutingPaths'>.
This error occurs while PduRRoutingPathGroupId is not unique in list PduRRoutingPathGroup

7.1.1.17 ERR051076: Handle Id(s) <'List PduRSourcePduHandleId'> configured for the parameter 'PduRSourcePduHandleId' in the container 'PduRSrcPdu's name' for the Source Module(s) <'List source module'> as Interface RoutingPath sequence should be unique with in the configuration set <'PduRRoutingPaths's Name'>.
This error occurs, if PduRSourcePduHandleId of PduRSrcPdu is not configured unique in list LoIf.

7.1.1.18 ERR051077: The Ref of the Container 'PduRSrcPdu' and 'PduRDestPdu' should be unique within the Configuration Set.
This error occurs, if PduRSrcPduRef and PduRDestPduRef is not configured unique.

7.1.1.19 ERR051078: When the value of the Parameter 'PduRSharedGatewayIfBuffer' is configured false, TxBuffer Reference(s) configured for the parameter 'PduRDestTxBufferRef' in the container 'PduRRoutingPath' should be unique within the Configuration Set.
This error occurs, if the PduRShared Gateway If Buffer is false and the PduRTxBuffer is not unique to the PduRoutingPath.

7.1.1.20 ERR051079: The Value of the Parameter 'ComIPduType' in the container 'ComIPdu's name' should be Configured as 'NORMAL', when the respective 'Pdu' is interacting with interface layers in the Configuration set ('PduRRoutingTables')
This error occurs when Pdu is interacting with interface layers but ComIPduType is configured as 'TP'

7.1.1.21 ERR051080: The Value of the Parameter 'ComIPduType' in the container 'ComIPdu's name' should be Configured as 'TP', when the respective 'Pdu' is interacting with interface layers in the Configuration set ('PduRRoutingTables')
This error occurs when Pdu is interacting with interface layers but ComIPduType is configured as 'NORMAL'

7.1.1.22 ERR051081: In Multicasted Gateway, all routing paths should be use common buffer that configured by 'PduRDestTxBufferRef' and common 'PduRQueueDepth'. Routing path (PduRRoutingPath) have value of parameter 'PduRDestTxBufferRef' or value of parameter 'PduRQueueDepth' is not correct
This error occurs, if PduRRoutingPath are multicasted gateway but they aren't configured same PduRDestTxBufferRef and same PduRQueueDepth

7.1.1.23 ERR051082: In Gateway, N:1 is support, all routing paths should be use common buffer that configured by PduRDestTxBufferRef and common queuedepth. Routing path (PduRRoutingPath) have value of parameter 'PduRDestTxBufferRef' or value of 'queuedepth' is not correct.
This error occurs, if PduRRoutingPath are N:1 gateway but they aren't configured same PduRDestTxBufferRef and same PduRQueueDepth

7.1.1.24 ERR051083: All routing path in N:1, multicast is not support. Routing path ('PduRRoutingPath') have parameter 'PduRDestPduRRef' or 'PduRSrcPduRRef' is not correct.
This error occurs when PduRRoutingPath is configured with multicast and N:1 at the same time.

7.1.1.25 ERR051085: Source ('PduRSrcPduRef') and destination ('PduRDestPduRef') should be configured same MetaDataTypes in routing path ('path') when value of parameter 'PduRMetaDataSupport' is configured as (true/1)
This error occurs when src and dest of PduRRoutingPath is not configured same MetaDataTypes

7.1.1.26 ERR051087: When the value of the parameter 'PduRDestPduDataProvision' is configured as (PDUR_TRIGGERTRANSMIT) and the value of the parameter 'PduRDestTxBufferRef' having reference to 'PduRTxBuffer' with the 'PduRTxBufferDepth' configured as (1). The value of the parameter 'PduRPduMaxLength' in the container ('PduRTxBuffer') should be equal value of the parameter 'PduLength' of container ('Pdu') in EcuC.
This error occurs when PduRDestPduDataProvision is configured as 'PDUR_TRIGGERTRANSMIT', PduRDestTxBufferRef is configured, PduRTxBufferDepth is configured as 1/ true but PduRPduMaxLength in PduRTxBuffer which is configured for PduRRoutingPath is not configured as same as PduLength of Pdu

7.1.1.27 ERR051088: The PduR Routing Path ('PduRRoutingPath') is configured isn't correct, this routing path is forwarding when destination of this routing path belong N:1 fashion. PduR isn't support for combined forwarding and gatewaying in n:1 fashion.
This error occurs when PduRRoutingPath is configured as forwarding (Lo module -> Up module) but PduRRoutingPath is also configured as N:1

7.1.1.28 ERR051090: The parameter 'PduRTpThreshold' of PduRRoutingPath ('PduRRoutingPath') is configured

incorrect. While using gatewaying on-the-fly, only one destination transport protocol module is allowed. In case multicast to more than one module, one destination module is configured with 'PduRTpThreshold'.
This error occurs when using gatewaying on-the-fly and multicast to more than one module but more than one destination module is configured with PduRTpThreshold

7.1.1.29 ERR051091: When configuration is Post-build, handle Id(s) <List all handle Id> configured for the parameter 'PduRSourcePduHandleId' in the container <PduRSrcPdu> for the Source Module(s) <LoIf or LoTp module as Lower Layer> as Interface RoutingPath sequence should be unique with in the configuration set <'PduRRoutingPaths'>.
This error occurs when the parameter 'PduRSourcePduHandleId' in the container <PduRSrcPdu> for the Source Module(s) <LoIf or LoTp module as Lower Layer> as Interface RoutingPath sequence is not unique with in the configuration set <'PduRRoutingPaths'>.

7.1.1.30 ERR051092: When configuration is Post-build, handle Id(s) <List all handle Id> configured for the parameter 'PduRSourcePduHandleId' in the container <PduRSrcPdu> for the Source Module(s) <LoIf or LoTp module as Lower Layer> should start with <0> and should be sequential within the configuration set <'PduRRoutingPaths'>.
This error occurs when the parameter 'PduRSourcePduHandleId' in the container <PduRSrcPdu> for the Source Module(s) <LoIf or LoTp module as Lower Layer> isn't start with <0> and isn't sequential within the configuration set <'PduRRoutingPaths'>.

7.1.1.31 ERR051093: When configuration is Post-build, handle Id(s) <List all handle Id> configured for the parameter 'DestPduHandleId' in the container <PduRDestPdu> for the Destination Module(s) <LoIf or LoTp module as Lower Layer> as Interface RoutingPath sequence should be unique with in the configuration set <'PduRRoutingPaths'>.
This error occurs when the parameter 'DestPduHandleId' in the container <PduRDestPdu> for the Destination Module(s) <LoIf or LoTp module as Lower Layer> as Interface RoutingPath sequence isn't unique with in the configuration set <'PduRRoutingPaths'>.

7.1.1.32 ERR051094: When configuration is Post-build, handle Id(s) <List all handle Id> configured for the parameter 'DestPduHandleId' in the container <PduRDestPdu> for the Destination Module(s) <LoIf or LoTp module as Lower Layer> should start with <0> and should be sequential within the configuration set <'PduRRoutingPaths'>.
This error occurs when the parameter 'DestPduHandleId' in the container <PduRDestPdu> for the Destination Module(s) <LoIf or LoTp module as Lower Layer> isn't start with <0> and isn't sequential within the configuration set <'PduRRoutingPaths'>.

7.1.1.33 ERR051095: RoutPathGrp Id(s) <List all path group Id> configured for the parameter 'PduRRoutingPathGroupId' in the container <PduRRoutingPathGroup> should start with <0> and should be sequential within the configuration set <'PduRRoutingPaths'>.
This error occurs when the parameter 'PduRRoutingPathGroupId' in the container <PduRRoutingPathGroup> isn't start with <0> and isn't sequential within the configuration set <'PduRRoutingPaths'>.

7.1.1.34 ERR051096: Configuration Id(s) <List all ConfigurationId> configured for the parameter 'PduRConfigurationId' in the container <PduRRoutingPaths> should be unique in the PduR configuration.
This error occurs when the parameter 'PduRConfigurationId' in the container <PduRRoutingPaths> isn't unique in the PduR configuration.

7.1.1.35 ERR051097: When variant of the Module is 'PRE-COMPILE', container 'PduRRoutingPaths' shouldn't be configured more than one.
This error occurs when variant of the Module is 'PRE-COMPILE', container 'PduRRoutingPaths' are configured more than one.

7.1.1.36 ERR051099: When configuration is post-build, in the container 'PduRDestPdu', parameter 'PduRTransmissionConfirmation' with same PduRDestPduHandleId of each variant should be configured same value
This error occurs when configuration is post-build, parameter 'PduRTransmissionConfirmation' with same PduRDestPduHandleId of each variant isn't configured same value

7.1.1.37 ERR051100: Mismatch between post-build variants collection defined in EcucPostBuildVariants and the list of variation points that was applied in PduR module
This error occurs when Mismatch between post-build variants collection defined in EcucPostBuildVariants and the list of variation points that was applied in PduR module

7.1.1.38 ERR051101:PduRDestTxBufferRef was not found when configured gateway N:1 in the 'PddRRoutingPath': {0}.
This error occurs when configure gateway N:1 without configure PduRDestTxBufferRef

7.1.1.39 ERR051102: PduRDestTxBufferRef must be the same on each routing path which have same dest when configured gateway N:1 'PddRRoutingPath': {0}.
This error occurs when configure PduRDestTxBufferRef is not a same for each routing path in gateway N:1

7.1.1.40 ERR051103 : Gateway N:1 must be FIFO, PduRQueueDepth have to greater than 1 in those 'PddRRoutingPaths' below: {0}.
This error occurs when configure PduRQueueDepth is not greater than 1 in gateway N:1

7.1.1.41 ERR051104 : PduRSharedGatewayIfBuffer is not supported.
This error occurs when PduRSharedGatewayIfBuffer option is true

7.1.2 Warning Messages

1) WRN051051: The Handle Id's configured for the Modules <PduRDestination Handle Id Source module Destination module PduRSource Handle Id>' should be same for the RoutingPath 'PduRRoutingPath's name', when the parameter 'PduRZeroCostOperation' in the container 'PduRGeneral' is configured as <true/1>.
This warning occurs, if the Handle Id's configured for the Modules <PduRDestination Handle Id Source module Destination module PduRSource Handle Id > are not same for the RoutingPath.

2) WRN051052: The Variant of the PduR Module should be 'PRE-COMPILE' when the parameter 'PduRZeroCostOperation' in the container 'PduRGeneral' is configured as <true/1>.
This warning occurs, if PduRZeroCostOperation in container PduRGeneral is configured as true/1 but the variant of module is not configured as 'PRE-COMPILE'

3) WRN051053: The PduR Routing Path should be Single Cast(i.e., one source and one Destination) and source and destination module is exclusively(i.e., Source is CanIf and Destination is Com, not exist Source is LinIf and Destination is Com), when the parameter 'PduRZeroCostOperation' the container 'PduRGeneral' is configured as <true/1>.
This warning occurs, if the PduR Routing Path is not Single Cast when the parameter 'PduRZeroCostOperation' in the container 'PduRGeneral' is configured as 'TRUE'.

4) WRN051054: Combined forwarding and gatewaying in n:1 fashion is not supported. Routing path <'PduRRoutingPath's name'> is configured is not correct
This warning occurs, if PduRRoutingPath is combined forwarding and gatewaying at the same time in case N:1

7.1.3 Information Messages

INF051051: The parameter 'PduRZeroCostOperation' in the container 'PduRGeneral' is configured as <false/0>, but the configuration satisfies all the necessary conditions to support Zero-cost Operation feature

1. Variant of the Module is 'PRE-COMPILE'

2. There is only single cast Routing Path(s) (One source and One destination)

3. There is only one bus used for communication(CAN or LIN or FLEXRAY or Ethernet).

Since all condtions are satisfied, the Zero-cost Opertaion feature is enabled.

This information occurs, if the parameter 'PduRZeroCostOperation' in the Container 'PduRGeneral' is not configured as 'TRUE'

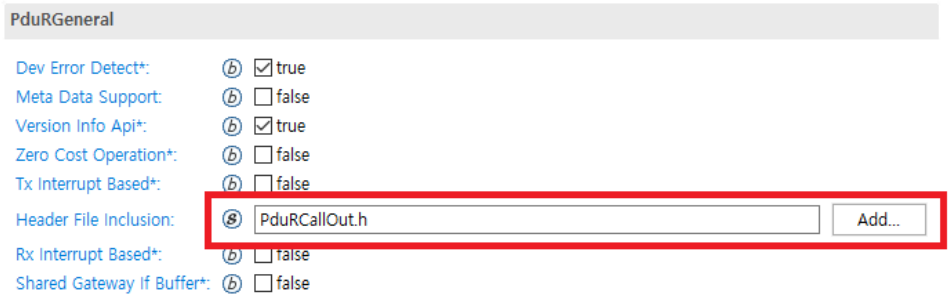
8. Appendix

8.1 A Guide to Configuring a Routing Path Callout

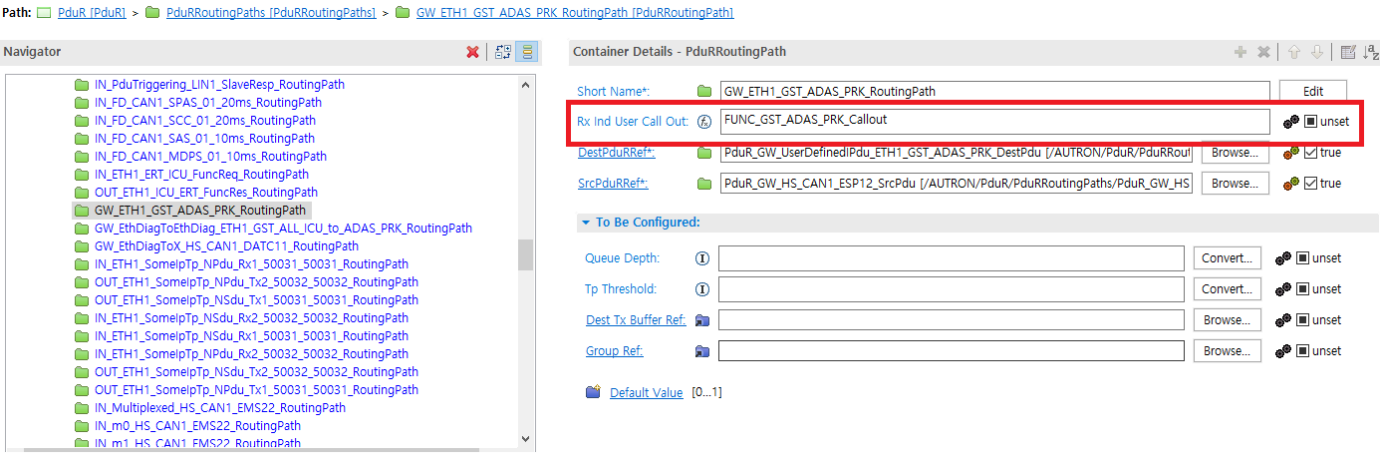
This is to be implemented by the user if necessary.

8.1.1 Configuration Guide

8.1.1.1 Add a Header File



8.1.1.2 Configure a Callout for PduRRoutingPath



8.1.1.3 Configure a Callout for PduRRoutingPath

Code can be developed based on the user's purpose. This callout is basically designed with Boolean as a return type; if its return value is "True or 1," routing will proceed, and if it's "False or 0." routing will be skipped.

```
/*Typedefine for Rx Indication Callout function*/
typedef P2FUNC(boolean, PDUR_APPL_CODE, PduRRxIndCallOut)
  (PduIdType PduId, P2CONST(PduInfoType, AUTOMATIC, PDUR_CONST) PduInfoPtr);
```

The following example is given as sample code. The user should implement this callout function keeping the type in mind and check the build for error and if it operates correctly.

PduR_CallOut.h

```
#ifndef PDUR_CALLOUT_H
#define PDUR_CALLOUT_H

#include "ComStack_Types.h"

extern boolean PduR_RxInd_CallOut_Func(PduIdType PduId, P2CONST(PduInfoType, AUTOMATIC, PDUR_CONST) PduInfoPtr);

#endif
```

PduR_CallOut.c

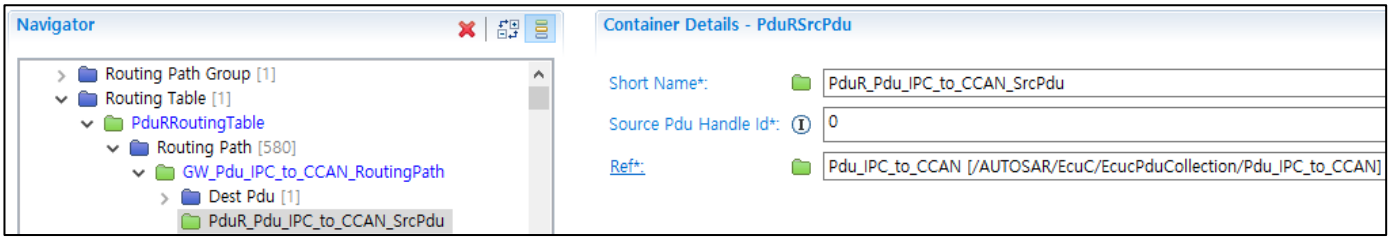
```
#include "PduR_CallOut.h"

boolean PduR_RxInd_CallOut_Func(PduIdType PduId, P2CONST(PduInfoType, AUTOMATIC, PDUR_CONST) PduInfoPtr)
{
  /* Add User Logc */
  return 1;
}
```

8.2 PduR API to support getting gateway buffer status

When PduRBuffer is set, it is an item that User must implement for the purpose of preventing buffer overflow. The example case is when using the buffer of PduR when ‘delivering data received from IPC from Cdd Router to PduR.

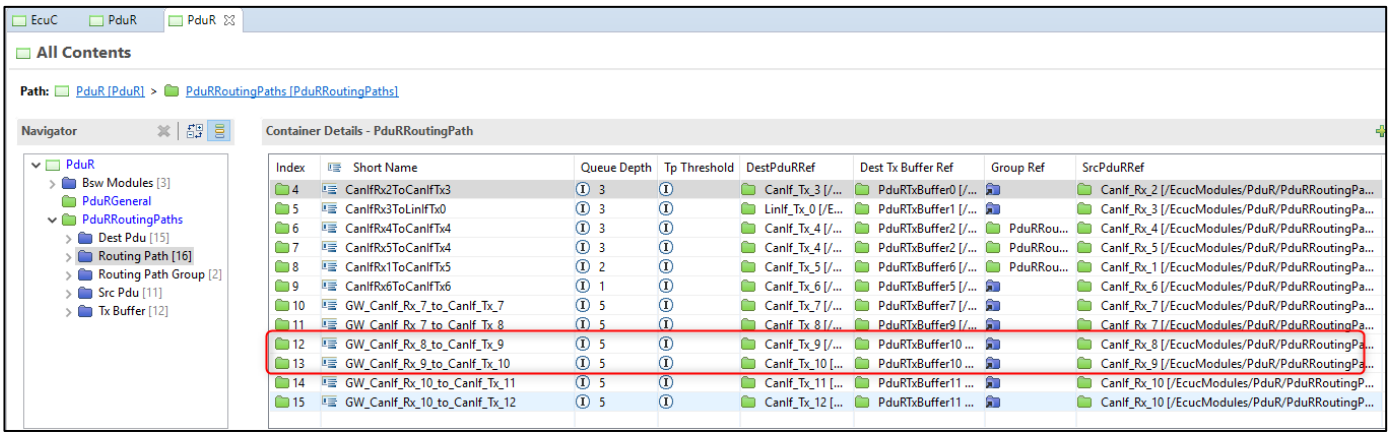
FUNC(void, PDUR_CODE) PduR_GetGwBufferStatus(PduIdType LddPduRSrcPduId,
  P2VAR(PduR_GwBufferStatus, AUTOMATIC, PDUR_VAR) LpGwStatus)
- LddPduRSrcPduId is the SourcePduHandleId of the PduR connected to the Cdd Router,
- Return is the routing path set by the user.
1) Count of DestPdu
2) Whether to set Buffer for each DestPdu
3) Whether the corresponding buffer is usable (if 1, it can be used).
(*Check before calling RxIndication)



8.3 PduR share gateway buffer for 1:1 gateway routing paths

PduRSharedGatewayIfBuffer is not supported.

For saving memory, users can configure multiple 1:1 gateway routing paths for using a single TxBuffer.
E.g. At below figure, GW_CanIf_Rx_8_to_CanIf_Tx_9 (1:1) and GW_CanIf_Rx_9_to_CanIf_Tx_10 (1:1) routing paths are configured with same buffer (PduRTxBuffer10).



Note that this configuration is only applicable for 1:1 gateway routing paths, and not applicable for 1:N or N:1 cases.