AUTOSAR ComM User Manual

Document Change History			
Date (YYYY-MM-DD)	Ver.	Editor	Content (before revision-> after revision)
2020-03-25	1.0.0.0	Jongsun Lim	ComM UM initial version
2021-09-06	1.0.1.0	Jongsun Lim	Scope of the Release Change Log
2021-10-29	1.0.2.0	ThuanVT	Scope of the Release Change Log
2022-02-24	1.0.3.0	ThuanVT	Scope of the Release Change Log Update Value and Category in Configuration Guide Update default value of ComMMainFunctionPeriod
2022-05-09	1.0.4.0	ThuanVT	Scope of the Release Change Log Add new parameter ComMBusloadDetectingApi Add new API ComM_GetChannelBusLoad
2022-06-20	1.0.4.1	SMKwon	Change Log
2022-08-16	1.1.0.0	ThuanVT	Scope of the Release Change Log
2022-09-15	1.1.1.0	ThuanVT	Scope of the Release Change Log
2022-10-31	1.1.2.0	NamNT1	Scope of the Release Change Log Error Message
2022-12-16	1.1.3.0	NamNT1	Scope of the Release

			Change Log
2023-03-07	1.1.4.0	NamNT1	Scope of the Release Change Log
2023-06-09	1.1.5.0	NamNT1	Scope of the Release Change Log Apply new UM template
2023-07-31	1.1.6.0	ThuanLD5	Scope of the Release Change Log
2024-01-31	1.1.7.0	MU Kim	Scope of the Release Change Log
2024-03-04	1.1.7.0_HF1	MU Kim	Change Log
2024-03-26	1.1.7.0_HF2	MU Kim	Apply new UM template

Table of Contents

1. Overview

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

The interpretation of the category related to setting is as follows

- Changeable (C) : Items that can be set by the user
- Fixed (F) : Items that cannot be changed by user
- Not Supported (N) : Items that not supported

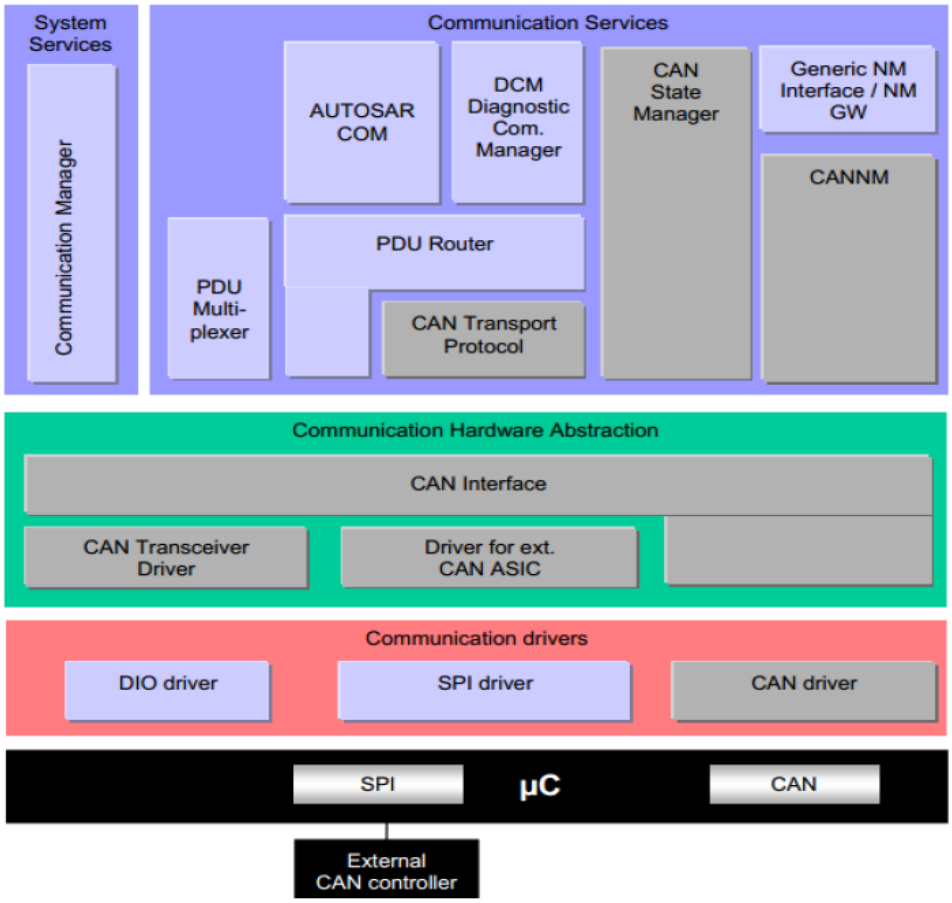
2. Reference

Sl. No.	Title	Version
1.	AUTOSAR_SWS_ COMManager.pdf	4.4.0

3. AUTOSAR System

3.1 ComM Module

Short Introduction and functional overview of module.
ComM module is a communication mode management for COM communication.



4. Limitations and Deviations

4.1 Limitations

- No support post-build for api ComM_Init

4.2 Deviations

- ComM_ResetRequestedChannel API for channel initialization is provided.

5. Configuration Guide

5.1 ComMGeneral

Parameter Name	Value	Category
ComM0PncVectorAvoidance	False	C
ComMDevErrorDetect	False	C
ComMDirectUserMapping	False	C
ComMEcuGroupClassification	3	C
ComMModeLimitationEnabled	False	C
ComMPncGatewayEnabled	False	C
ComMPncPrepareSleepTimer	Automated	C
ComMPncSupport	Automated	C
ComMResetAfterForcingNoComm	False	C
ComMSynchronousWakeUp	True	C
ComMTMinFullComModeDuration	5	C
ComMVersionInfoApi	False	C
ComMWakeupInhibitionEnabled	False	C
ComMCanCMSupport	False	C
ComMGlobalNvMBlockDescriptor	User Defined	C
ComMBusloadDetectingApi	False	C

1) ComM0PncVectorAvoidance
   - This parameter avoids sending of 0-PNC-Vectors in case ComMPncGatewayEnabled is enabled.
2) ComMDevErrorDetect
   - Switches the development error detection and notification on or off.
3) ComMDirectUserMapping
   - If this parameter is set to true the configuration tool shall automatically create a ComMUser per ComMPnc and a ComMUser per ComMChannel.
   - The shortName of the generated ComMUsers shall follow the following naming convention:
   - PNCUser_ComMPncId, e.g. PNCUser_13
   - ChannelUser_ComMChannelId, e.g. ChannelUser_25
   - Restriction: ComMUser, which are created due to this configuration parameter, shall not be used by SWCs (only available for BswM).
4) ComMEcuGroupClassification

- Defines whether a mode inhibition affects the ECU or not.
- Example) 000: No mode inhibition can be activated
  001: Wake up inhibition can be enabled

5) ComMModeLimitationEnabled
   - true if mode limitation functionality shall be enabled.
6) ComMPncGatewayEnabled
   - Enables or disables support of Partial Network Gateway.
7) ComMPncPrepareSleepTimer
   - Time in seconds the PNC state machine shall wait in COMM_PNC_PREPARE_SLEEP.
8) ComMPncSupport
   - Enables or disables support of partial networking.
9) ComMResetAfterForcingNoComm
   - ComM shall perform a reset after entering "No Communication" mode because of an active mode limitation to "No Communication" mode.
10) ComMSynchronousWakeUp
    - Wake up of one channel shall lead to a wake up of all channels if true.
11) ComMTMinFullComModeDuration
    - Minimum time duration in seconds, spent in the COMM_FULL_COMMUNICATION sub-state COMM_FULL_COM_NETWORK_REQUESTED.
    - When using the NM Light option, the minimum time setting to maintain the FULL COMM state after switching from FULL COMM to NOCOMM (during this time, the user's NO COMM request is not processed
12) ComMVersionInfoApi
    - Switches the possibility to read the published information with the service ComM_GetPublishedInformation().
13) ComMWakeupInhibitionEnabled
    - True if wake up inhibition functionality enabled.
14) ComMCanCMSupport
    - True if CanCM module support enabled.
15) ComMGlobalNvMBlockDescriptor
    - Reference to NVRAM block containing the none volatile data.
    - If this parameter is not configured it means that no NVRam is used at all.
16) ComMBusloadDetectingApi
    - Selects support for CAN Busload Detecting Api.
    - If True, CanIfBusloadDetectingSupport should be True

5.2 ComMConfigSet

Parameter Name	Value	Category
ComMPncEnabled	Automated	C

1) ComMPncEnabled
   - Defines whether in this configuration set the partial networking is enabled.

5.3 ComMConfigSet-ComMUser

Parameter Name	Value	Category
ComMUserIdentifier	Automated	C
ComMUserEcucPartitionRef	User Defined	C

1) ComMUserIdentifier
   - An identifier that is needed to refer to a user in the system which is designated to request Communication Modes.
2) ComMUserEcucPartitionRef
   - Denotes in which "EcucPartition" the requester is executed. When the partition is stopped, the communication request shall be cancelled in the ComM

5.4 ComMConfigSet-ComMChannel

Parameter Name	Value	Category
ComMBusType	Automated	C
ComMCDDBusPrefix	User Defined	C
ComMChannelId	Automated	C
ComMFullCommRequestNotificationEnabled	False	C
ComMMainFunctionPeriod (1)	0.005	C
ComMNoCom	False	C
ComMNoWakeup	False	C
ComMNoWakeUpInhibitionNvmStorage	Automated	C
ComMPncGatewayType	User Defined	C
ComMManageReference	User Defined	C

1) ComMBusType

- Identifies the bus type of the channel.

2) ComMCDDBusPrefix
   - Prefix to be used for API calls to CDD.
3) ComMChannelId
   - Channel identification number of the corresponding channel.
4) ComMFullCommRequestNotificationEnabled
   - Defines if the optional SenderReceiver Port of Interface ComM_CurrentChannelRequest will be provided for this channel.
5) ComMMainFunctionPeriod
   - Specifies the period in seconds that the MainFunction has to be triggered with.
   - Setting the execution cycle of the main function (For the accuracy of time calculation, all setting values related to the time of the ComM module should be set in multiples of the main function cycle
   - ComM scheduling shall be at least as fast as the communication stack and a schedule longer than 100ms makes no sense for communication.
6) ComMNoCom
   - Not allowed to change state of ComM channel to COMM_SILENT_COMMUNICATION or COMM_FULL_COMMUNICATION.
   - Shall be possible to change parameter during runtime with ComM API's.
   - ECU/All channels: ComM_LimitECUToNoComMode().
   - Separate channels: ComM_LimitChannelToNoComMode().
7) ComMNoWakeup
   - Defines if an ECU is not allowed to wake-up the channel.
   - true: Enabled (not allowed to wake-up))
   - false: Disabled
   - This is the default/init value of a runtime variable that can be changed during runtime using ComM_PreventWakeUp().
8) ComMNoWakeUpInhibitionNvmStorage
   - If this parameter is set to "true", the NoWakeUp inhibition state of the channel shall be stored (in some implementation specific way) in the block pointed to by ComMGlobalNvmBlockDescriptor.
9) ComMPncGatewayType
   - Identifies the Partial Network Gateway behaviour of a ComMChannel.
10) ComMManageReference
   - Represenst the reference between a ComMChannel with role managing channel and a ComMChannel with role managed channel.

5.5 ComMConfigSet-ComMChannel-ComMNetworkManagement

Parameter Name	Value	Category
ComMNmLightTimeout	Automated	C
ComMNmVariant	Automated	C
ComMPncNmRequest	Automated	C

1) ComMNmLightTimeout
   - Defines the timeout (in seconds) after COMM_FULL_COMMUNICATION sub-state COMM_FULL_COM_READY_SLEEP is left.
   - The range shall be greater than 0.0 and less or equal to 255.0.
2) ComMNmVariant
   - Defines the functionality of the networkmanagement.
   - Shall be harmonized with NM configuration.
3) ComMPncNmRequest
   - If this parameter equals true then every time a FULL Communication is requested due to a change in the PNC state machine to COMM_PNC_REQUESTED Nm shall be called using the API Nm_NetworkRequest.

5.6 ComMConfigSet-ComMChannel-ComMUserPerChannel

Parameter Name	Value	Category
ComMUserChannel	Automated	C

1) ComMUserChannel
   - Reference to the ComMUser that corresponds to this channel user.
   - ImplementationType: COMM_UserHandleType

5.7 ComMConfigSet-ComMPnc

Parameter Name	Value	Category
ComMPncId	Automated	C
ComMChannelPerPnc	Automated	C
ComMPncEthIfSwitchPortGroupRef	User Defined	C
ComMUserPerPnc	Automated	C

1) ComMPncId
   - Partial network cluster identification number.
2) ComMChannelPerPnc
   - Reference to the ComMChannel that is required for this PNC.
   - ImplementationType: NetworkHandleType
3) ComMPncEthIfSwitchPortGroupRef
   - Reference to the PortGroups that correspond to this PNC.
   - This is only for documentation.
4) ComMUserPerPnc
   - Reference to the ComMUsers that correspond to this PNC.
   - ImplementationType: COMM_UserHandleType

5.8 ComMConfigSet-ComMPnc-ComMPncComSignal

Parameter Name	Value	Category
ComMPncComSignalDirection	Automated	C
ComMPncComSignalKind	Automated	C
ComMPncComSignalChannelRef	Automated	C
ComMPncComSignalRef	Automated	C

1) ComMPncComSignalDirection
   - Indicates the communication direction of this PncComSignal.
2) ComMPncComSignalKind
   - Indicates whether this PncComSignal represents EIRA or ERA PNC information.
   - This parameter ComMPncComSignalKind is optional and shall be ignored when ComMPncComSignalDirection equals TX.
3) ComMPncComSignalChannelRef
   - Reference to the ComMChannel which is used to determine whether this PncComSignal shall participate in the active or passive role (via the parameter ComMPncGatewayType of the ComMChannel).
4) ComMPncComSignalRef
   - Reference to the ComSignal which is used to transport the partial network channel request information.

6. Application Programming Interface (API)

6.1 Type Definitions

6.1.1 ComM_InitStatusType

Name	ComM_InitStatusType	
Type	Enumeration	
Range	COMM_UNINIT	0x00
	COMM_INIT	0x01

6.1.2 ComM_PncModeType

Name	ComM_PncModeType	
Type	Enumeration	
Range	COMM_PNC_REQUESTED	0x00
	COMM_PNC_READY_SLEEP	0x01
	COMM_PNC_PREPARE_SLEEP	0x02
	COMM_PNC_NO_COMMUNICATION	0x03

6.1.3 ComM_StateType

Name	ComM_ConfigType	
Type	uint8	
Range	COMM_NO_COM_NO_PENDING_REQUEST	0
	COMM_NO_COM_REQUEST_PENDING	1
	COMM_FULL_COM_NETWORK_REQUESTED	2
	COMM_FULL_COM_READY_SLEEP	3
	COMM_SILENT_COM	4

6.1.4 ComM_ConfigType

No supported

6.2 Macro Constants

None

6.3 Functions

6.3.1 ComM_Init

Function Name	ComM_Init	
Syntax:	FUNC(void, COMM_CODE) ComM_Init(P2CONST(ComM_ConfigType, AUTOMATIC, COMM_APPL_CONST) ConfigPtr)	
Service ID[hex]	0x01	
Sync/Async	Synchronous	
Reentrancy	Non Reentrant	
Parameters (In)	ConfigPtr	Pointer to post-build configuration data
Parameters (Inout)	None	
Parameters (Out)	None	
Return Value	None	
Description	Initializes the AUTOSAR Communication Manager and restarts the internal state machines. This function is used by BSW.	
Preconditions	None	
Configuration Dependency	None	

6.3.2 ComM_DeInit

Function Name	ComM_DeInit
Syntax:	FUNC(void, COMM_CODE) ComM_DeInit(void)
Service ID[hex]	0x02
Sync/Async	Synchronous
Reentrancy	Non Reentrant
Parameters (In)	None
Parameters (Inout)	None
Parameters (Out)	None
Return Value	None

Description	This API de-initializes the AUTOSAR Communication Manager. This function is used by BSW.
Preconditions	ComM shall be initialized.
Configuration Dependency	None

6.3.3 ComM_GetStatus

Function Name	ComM_GetStatus	
Syntax:	FUNC(Std_ReturnType, COMM_CODE) ComM_GetStatus (P2VAR (ComM_InitStatusType , AUTOMATIC, COMM_APPL_DATA) Status)	
Service ID[hex]	0x03	
Sync/Async	Synchronous	
Reentrancy	Non Reentrant	
Parameters (In)	Channel Identifier	
Parameters (Inout)	None	
Parameters (Out)	Status	COMM_UNINIT: The ComM is not initialized or not usable. Default value after startup or after ComM_DeInit() is called. COMM_INIT: The ComM is initialized and usable.
Return Value	Std_ReturnType	E_OK: Successfully return of initialization status E_NOT_OK: Return of initialization status failed
Description	Returns the initialization status of the AUTOSAR Communication Manager. After a call to ComM_DeInit() ComM should have status COMM_UNINIT, and a new call to ComM_Init needed to make sure ComM restart internal state machines to defailt values. This function is used by BSW.	
Preconditions	ComM shall be initialized.	
Configuration Dependency	None	

6.3.4 ComM_GetVersionInfo

Function Name	ComM_GetVersionInfo	
Syntax:	FUNC(void, COMM_CODE) ComM_GetVersionInfo(P2VAR(Std_VersionInfoType, AUTOMATIC, COMM_APPL_DATA) Versioninfo)	
Service ID	0x10	
Sync/Async	Synchronous	
Reentrancy	Reentrant	
Parameters (In)	None	
Parameters (Inout)	None	
Parameters (Out)	Versioninfo	See STD_VersionInfo Type
Return Value	None	
Description	This function returns the published information including Major, Minor and Patch Version. This function is used by user. But it needs configuration. (It cannot be called directly by user)	
Preconditions	None	
Configuration Dependency	None	

6.3.5 ComM_RequestComMode

Function Name	ComM_RequestComMode	
Syntax:	FUNC(Std_ReturnType, COMM_CODE) ComM_RequestComMode(ComM_UserHandleType User, ComM_ModeType ComMode)	
Service ID	0x05	
Sync/Async	Synchronous	
Reentrancy	Reentrant	
Parameters (In)	User	
	ComMode	COMM_FULL_COMMUNICATION
		COMM_NO_COMMUNICATION
Parameters (Inout)	None	
Parameters (Out)	None	
Return Value	Std_ReturnType	E_OK: Successfully changed to the new mode
		E_NOT_OK: Changing to the new mode failed
		COMM_E_MODE_LIMITATION: Mode cannot be granted because of mode inhibition.

Description	Requesting of a communication mode by a user This function is used by user. But it needs configuration. (It cannot be called directly by user)
Preconditions	Previous mode request should be completed.
Configuration Dependency	None
In Communication with application SW-C	Rte_Call_ <RPort>_RequestComMode

6.3.6 ComM_GetMaxComMode

Function Name	ComM_GetMaxComMode	
Syntax:	Std_ReturnType ComM_GetMaxComMode(ComM_UserHandleType User, ComM_ModeType* ComMode)	
Service ID	0x06	
Sync/Async	Synchronous	
Reentrancy	Reentrant	
Parameters (In)	User	
Parameters (Inout)	None	
Parameters (Out)	ComMode	See ComM_ModeType
Return Value	Std_ReturnType	E_OK: Successfully returned maximum allowed Communication Mode
		E_NOT_OK: Return of maximum allowed Communication Mode failed
Description	Function to query the maximum allowed Communication Mode of the corresponding user. This function is used by user. But it needs configuration. (It cannot be called directly by user)	
Preconditions	None	
Configuration Dependency	None	
In Communication with application SW-C	Rte_Call_ <RPort>_GetMaxComMode	

6.3.7 ComM_GetRequestedComMode

Function Name	ComM_GetRequestedComMode	
Syntax:	FUNC(Std_ReturnType, COMM_CODE) ComM_GetRequestedComMode(ComM_UserHandleType User, P2VAR(ComM_ModeType, AUTOMATIC, COMM_APPL_DATA) ComMode)	
Service ID	0x06	
Sync/Async	Synchronous	
Reentrancy	Reentrant	
Parameters (In)	User	
Parameters (Inout)	None	
Parameters (Out)	ComMode	Name of the requested mode
Return Value	Std_ReturnType	E_OK: Successfully returned requested Communication Mode
		E_NOT_OK: Return of requested Communication Mode failed
Description	Function to query the currently requested Communication Mode of the corresponding user. This function is used by user. But it needs configuration. (It cannot be called directly by user)	
Preconditions	None	
Configuration Dependency	None	
In Communication with application SW-C	Rte_Call_ <RPort>_GetRequestedComMode	

6.3.8 ComM_GetCurrentComMode

Function Name	ComM_GetCurrentComMode
Syntax:	FUNC(Std_ReturnType, COMM_CODE) ComM_GetCurrentComMode(ComM_UserHandleType User, P2VAR(ComM_ModeType, AUTOMATIC, COMM_APPL_DATA) ComMode)
Service ID	0x08
Sync/Async	Synchronous
Reentrancy	Reentrant
Parameters (In)	User
Parameters (Inout)	None

Parameters (Out)	ComMode	See ComM_ModeType
Return Value	Std_ReturnType	
Description	Function to query the current communication mode. This function is used by user. But it needs configuration. (It cannot be called directly by user)	
Preconditions	ComM shall be initialized.	
Configuration Dependency	None	
In Communication with application SW-C	Rte_Call_ <RPort>_GetCurrentComMode	

6.3.9 ComM_PreventWakeUp

Function Name	ComM_PreventWakeUp	
Syntax:	FUNC(Std_ReturnType, COMM_CODE) ComM_PreventWakeUp(NetworkHandleType Channel, boolean Status)	
Service ID	0x09	
Sync/Async	Synchronous	
Reentrancy	Reentrant	
Parameters (In)	Channel	See NetworkHandleType
	Status	FALSE: Wake up inhibition is switched off TRUE: Wake up inhibition is switched on
Parameters (Inout)	None	
Parameters (Out)	None	
Return Value	Std_ReturnType	E_OK: Successfully changed wake up status for the channel
		E_NOT_OK: Change of wake up status for the channel failed
Description	Changes the inhibition status COMM_NO_WAKEUP for the corresponding channel This function is used by user. But it needs configuration. (It cannot be called directly by user)	
Preconditions	ComM shall be initialized.	
Configuration Dependency	ComMGeneral/ComMWakeupInhibitionEnabled set to True	

In Communication with application SW-C	Rte_Call_ <RPort>_PreventWakeUp

6.3.10 ComM_GetInhibitionStatus

Function Name	ComM_GetInhibitionStatus	
Syntax:	FUNC(Std_ReturnType, COMM_CODE) ComM_GetInhibitionStatus(NetworkHandleType Channel, P2VAR(ComM_InhibitionStatusType, AUTOMATIC, COMM_APPL_DATA) Status)	
Service ID	0x04	
Sync/Async	Synchronous	
Reentrancy	Reentrant	
Parameters (In)	Channel	See NetworkHandleType
Parameters (Inout)	None	
Parameters (Out)	Status	See ComM_InhibitionStatusType
Return Value	Std_ReturnType	E_OK: Successfully returned Inhibition status
		E_NOT_OK: return of Inhibition status failed
Description	Returns the inhibition status of a ComM channel This function is used by user. But it needs configuration. (It cannot be called directly by user)	
Preconditions	ComM shall be initialized.	
Configuration Dependency	ComMGeneral/ComMWakeupInhibitionEnabled or ComMGeneral/ ComMModeLimitationEnabled set to True	
In Communication with application SW-C	Rte_Call_ <RPort>_GetInhibitionStatus	

6.3.11 ComM_LimitChannelToNoComMode

Function Name	ComM_LimitChannelToNoComMode
Syntax:	FUNC(Std_ReturnType, COMM_CODE) ComM_LimitChannelToNoComMode(NetworkHandleType Channel, boolean Status)
Service ID	0x0b
Sync/Async	Synchronous

Reentrancy	Reentrant	
Parameters (In)	Channel	See NetworkHandleType
	Status	FALSE: Limit channel to COMM_NO_COMMUNICATION disabled TRUE: Limit channel to COMM_NO_COMMUNICATION enabled
Parameters (Inout)	None	
Parameters (Out)	None	
Return Value	Std_ReturnType	E_OK: Successfully changed inhibition status for the channel E_NOT_OK: Change of inhibition status for the channel failed, e.g. ComMEcuGroupClassification disables the functionality (see ECUC_ComM_00563)
Description	Changes the inhibition status for the channel for changing from COMM_NO_COMMUNICATION to a higher Communication Mode. This function is used by user. But it needs configuration. (It cannot be called directly by user)	
Preconditions	ComM shall be initialized.	
Configuration Dependency	ComMGeneral/ComMResetAfterForcingNoComm and ComMGeneral/ ComMModeLimitationEnabled are set to True	
In Communication with application SW-C	Rte_Call_ <RPort>_LimitChannelToNoComMode	

6.3.12 ComM_SetECUGroupClassification

Function Name	ComM_SetECUGroupClassification	
Syntax:	FUNC(Std_ReturnType, COMM_CODE) ComM_SetECUGroupClassification(ComM_InhibitionStatusType Status)	
Service ID	0x0f	
Sync/Async	Synchronous	
Reentrancy	None Reentrant	
Parameters (In)	Status	See ComM_InhibitionStatusType
Parameters (Inout)	None	
Parameters (Out)	None	
Return Value	Std_ReturnType	E_OK: Successfully change the ECU Group Classification Status
		E_NOT_OK: Change of the ECU Group Classification Status failed
Description	Changes the ECU Group Classification status This function is used by user.	

	But it needs configuration. (It cannot be called directly by user)
Preconditions	ComM shall be initialized.
Configuration Dependency	None
In Communication with application SW-C	Rte_Call_ <RPort>_SetECUGroupClassification

6.3.13 ComM_ResetInhibitCounter

Function Name	ComM_ResetInhibitCounter	
Syntax:	FUNC(Std_ReturnType, COMM_CODE) ComM_ResetInhibitCounter(void)	
Service ID	0x0e	
Sync/Async	Synchronous	
Reentrancy	None Reentrant	
Parameters (In)	None	
Parameters (Inout)	None	
Parameters (Out)	None	
Return Value	Std_ReturnType	E_OK: Successfully reset of Inhibit COMM_FULL_COMMUNICATION Counter
		E_NOT_OK: Reset of Inhibit COMM_FULL_COMMUNICATION Counter failed
Description	This function resets the Inhibited COMM_FULL_COMMUNICATION request Counter This function is used by user. But it needs configuration. (It cannot be called directly by user)	
Preconditions	ComM shall be initialized.	
Configuration Dependency	ComMGeneral/ComMModeLimitationEnabled is True and ComMGeneral/ComMGlobalNvMBlockDescriptor is configured	
In Communication with application SW-C	Rte_Call_ <RPort>_ResetInhibitCounter	

6.3.14 ComM_ReadInhibitCounter

Function Name	ComM_ReadInhibitCounter
Syntax:	FUNC(Std_ReturnType, COMM_CODE)

	ComM_ReadInhibitCounter(P2VAR(uint16, AUTOMATIC, COMM_APPL_DATA) CounterValue)	
Service ID	0x0d	
Sync/Async	Synchronous	
Reentrancy	None Reentrant	
Parameters (In)	None	
Parameters (Inout)	None	
Parameters (Out)	CounterValue	Amount of rejected COMM_FULL_COMMUNICATION user requests
Return Value	Std_ReturnType	E_OK: Successfully returned Inhibition Counter
		E_NOT_OK: Return of Inhibition Counter failed
Description	This function returns the amount of rejected COMM_FULL_COMMUNICATION user requests This function is used by user. But it needs configuration. (It cannot be called directly by user)	
Preconditions	ComM shall be initialized.	
Configuration Dependency	ComMGeneral/ComMModeLimitationEnabled is True and ComMGeneral/ComMGlobalNvMBlockDescriptor is configured	
In Communication with application SW-C	Rte_Call_ <RPort>_ReadInhibitCounter	

6.3.15 ComM_LimitECUToNoComMode

Function Name	ComM_LimitECUToNoComMode	
Syntax:	FUNC(Std_ReturnType, COMM_CODE) ComM_LimitECUToNoComMode(boolean Status)	
Service ID	0x0c	
Sync/Async	Synchronous	
Reentrancy	None Reentrant	
Parameters (In)	Status	FALSE: Limit ECU to COMM_NO_COMMUNICATION disabled
		TRUE: Limit ECU to COMM_NO_COMMUNICATION enabled
Parameters (Inout)	None	
Parameters (Out)	None	
Return Value	Std_ReturnType	E_OK: Successfully changed inhibition status for the ECU

		E_NOT_OK: Change of inhibition status for the ECU failed
Description	Changes the inhibition status for the ECU (=all channels) for changing from COMM_NO_COMMUNICATION to a higher Communication Mode This function is used by user. But it needs configuration. (It cannot be called directly by user)	
Preconditions	ComM shall be initialized.	
Configuration Dependency	ComMGeneral/ComMModeLimitationEnabled and ComMGeneral/ComMResetAfterForcingNoComm are set to True	
In Communication with application SW-C	Rte_Call_ <RPort>_LimitECUToNoComMode	

6.3.16 ComM_Mainfunction

Function Name	ComM_MainFunction	
Syntax:	FUNC(void, COMM_CODE) ComM_MainFunction (NetworkHandleType Channel)	
Service ID	0x60	
Sync/Async	Synchronous	
Reentrancy	Non Reentrant	
Parameters (In)	Channel	Identifier of the channel
Parameters (Inout)	None	
Parameters (Out)	None	
Return Value	None	
Description	This function shall perform the processing of the AUTOSAR ComM activities that are not directly initiated by the calls e.g. from the RTE. There shall be one dedicated Main Function for each channel of ComM. This function is used by BSW.	
Preconditions	ComM shall be initialized.	
Configuration Dependency	None	

6.3.17 Network Management Interface

Function Name	ComM_Nm_NetworkStartIndication	
Syntax:	FUNC(void, COMM_CODE) ComM_Nm_NetworkStartIndication(NetworkHandleType Channel)	
Service ID	0x15	
Sync/Async	Asynchronous	
Reentrancy	Reentrant	
Parameters (In)	Channel	See NetworkHandleType
Parameters (Inout)	None	
Parameters (Out)	None	
Return Value	None	
Description	Indication that a NM-message has been received in the Bus Sleep Mode, what indicates that some nodes in the network have already entered the Network Mode. This function is used by BSW.	
Preconditions	ComM shall be initialized.	
Configuration Dependency	None	

Function Name	ComM_Nm_NetworkMode	
Syntax:	FUNC(void, COMM_CODE) ComM_Nm_NetworkMode (NetworkHandleType Channel)	
Service ID	0x18	
Sync/Async	Asynchronous	
Reentrancy	Reentrant	
Parameters (In)	Channel	Channel
Parameters (Inout)	None	
Parameters (Out)	None	
Return Value	None	
Description	Notification that the network management has entered Network Mode. This function is used by BSW.	
Preconditions	ComM shall be initialized.	

Configuration Dependency	None

Function Name	ComM_Nm_PrepareBusSleepMode	
Syntax:	FUNC(void, COMM_CODE) ComM_Nm_PrepareBusSleepMode (NetworkHandleType Channel)	
Service ID	0x19	
Sync/Async	Asynchronous	
Reentrancy	Reentrant	
Parameters (In)	Channel	Channel
Parameters (Inout)	None	
Parameters (Out)	None	
Return Value	None	
Description	Notification that the network management has entered Prepare Bus-Sleep Mode. Reentrancy: Reentrant (but not for the same NM-Channel). This function is used by BSW.	
Preconditions	ComM shall be initialized.	
Configuration Dependency	None	

Function Name	ComM_Nm_BusSleepMode	
Syntax:	FUNC(void, COMM_CODE) ComM_Nm_BusSleepMode (NetworkHandleType Channel)	
Service ID	0x1a	
Sync/Async	Asynchronous	
Reentrancy	Reentrant	
Parameters (In)	Channel	Channel
Parameters (Inout)	None	
Parameters (Out)	None	
Return Value	None	
Description	Notification that the network management has entered Bus-Sleep Mode. This callback function should perform a transition of the hardware and transceiver	

	to bus-sleep mode. This function is used by BSW.
Preconditions	ComM shall be initialized.
Configuration Dependency	None

Function Name	ComM_Nm_RestartIndication	
Syntax:	FUNC(void, COMM_CODE) ComM_Nm_RestartIndication (NetworkHandleType Channel)	
Service ID	0x1b	
Sync/Async	Asynchronous	
Reentrancy	Reentrant	
Parameters (In)	Channel	Channel
Parameters (Inout)	None	
Parameters (Out)	None	
Return Value	None	
Description	If Nmlf has started to shut down the coordinated busses, AND not all coordinated busses have indicated bus sleep state, AND on at least on one of the coordinated busses NM is restarted, THEN the NM Interface shall call the callback function ComM_Nm_RestartIndication with the nmNetworkHandle of the channels which have already indicated bus sleep state. This function is used by BSW.	
Preconditions	ComM shall be initialized.	
Configuration Dependency	None	

6.3.18 Diagnostic Communication Manager Interface

Function Name	ComM_DCM_ActiveDiagnostic
Syntax:	FUNC(void, COMM_CODE) ComM_DCM_ActiveDiagnostic (NetworkHandleType Channel)
Service ID	0x1f

Sync/Async	Synchronous	
Reentrancy	Reentrant	
Parameters (In)	Channel	Channel needed for Diagnostic communication
Parameters (Inout)	None	
Parameters (Out)	None	
Return Value	None	
Description	Indication of active diagnostic by the DCM. This function is used by BSW.	
Preconditions	ComM shall be initialized.	
Configuration Dependency	None	

Function Name	ComM_DCM_InactiveDiagnostic	
Syntax:	FUNC(void, COMM_CODE) ComM_DCM_InactiveDiagnostic (NetworkHandleType Channel)	
Service ID	0x20	
Sync/Async	Synchronous	
Reentrancy	Reentrant	
Parameters (In)	Channel	Channel needed for Diagnostic communication
Parameters (Inout)	None	
Parameters (Out)	None	
Return Value	None	
Description	Indication of inactive diagnostic by the DCM. This function is used by BSW.	
Preconditions	ComM shall be initialized.	
Configuration Dependency	None	

6.3.19 ECU State Manager Interface

Function Name	ComM_EcuM_WakeUpIndication
Syntax:	FUNC(void, COMM_CODE) ComM_EcuM_WakeUpIndication

	(NetworkHandleType Channel)	
Service ID	0x2a	
Sync/Async	Synchronous	
Reentrancy	Reentrant	
Parameters (In)	Channel	Channel
Parameters (Inout)	None	
Parameters (Out)	None	
Return Value	None	
Description	Notification of a wake up on the corresponding channel. This function is used by BSW.	
Preconditions	ComM shall be initialized.	
Configuration Dependency	None	

Function Name	ComM_EcuM_PNCWakeUpIndication	
Syntax:	FUNC(void, COMM_CODE) ComM_EcuM_PNCWakeUpIndication (PNCHandleType PNCid)	
Service ID	0x37	
Sync/Async	Synchronous	
Reentrancy	Reentrant	
Parameters (In)	PNCid	Identifier of the partial network cluster
Parameters (Inout)	None	
Parameters (Out)	None	
Return Value	None	
Description	Notification of a wake up on the corresponding partial network cluster. This function is used by BSW.	
Preconditions	ComM shall be initialized.	
Configuration Dependency	None	

Function Name	ComM_CommunicationAllowed
Syntax:	FUNC(void, COMM_CODE) ComM_CommunicationAllowed

	(NetworkHandleType Channel, boolean Allowed)	
Service ID	0x35	
Sync/Async	Asynchronous	
Reentrancy	Non Reentrant	
Parameters (In)	Channel	Channel
	Allowed	TRUE: Communication is allowed FALSE: Communication is not allowed
Parameters (Inout)	None	
Parameters (Out)	None	
Return Value	None	
Description	EcuM or BswM shall indicate to ComM when communication is allowed. If EcuM/Flex is used: BswM. This function is used by BSW.	
Preconditions	ComM shall be initialized.	
Configuration Dependency	None	

6.3.20 Bus State Manager Interface

Function Name	ComM_BusSM_ModeIndication	
Syntax:	FUNC(void, COMM_CODE) ComM_BusSM_ModeIndication (NetworkHandleType Channel, ComM_ModeType ComMode)	
Service ID	0x33	
Sync/Async	Asynchronous	
Reentrancy	Reentrant	
Parameters (In)	Channel	See NetworkHandleType
	ComMode	See ComM_ModeType
Parameters (Inout)	None	
Parameters (Out)	None	
Return Value	None	
Description	Indication of the actual bus mode by the corresponding Bus State Manager. ComM shall propagate the indicated state to the users with means of the RTE and BswM. This function is used by BSW.	

Preconditions	ComM shall be initialized.
Configuration Dependency	None

Function Name	ComM_BusSM_BusSleepMode	
Syntax:	FUNC(void, COMM_CODE) ComM_BusSM_BusSleepMode (NetworkHandleType Channel)	
Service ID	0x34	
Sync/Async	Synchronous	
Reentrancy	Reentrant	
Parameters (In)	Channel	Identifier of the channel
Parameters (Inout)	None	
Parameters (Out)	None	
Return Value	None	
Description	Notification of the corresponding Bus State Manager that the actual bus mode is Bus-Sleep. Only applicable for LIN slave nodes. This function is used by BSW.	
Preconditions	ComM shall be initialized.	
Configuration Dependency	ComM_BusSM_BusSleepMode is true.	

7. Generator

7.1 Generator Message

7.1.1 Information Message

Non information message

7.1.2 Waning Message

Non warning message

7.1.3 Error Message

ERR0120001 if (output source files generated do not have uneven number of fields in the structures)

Unexpected Error Found. Please contact AUTRON AUTOSAR Support System

ERR0120002 if (Incorrect Configuration of paramater(s) and conatiner(s) in input file(s)).

Unexpected Error Found. This error may be due to the incorrect configuration of the element(s) 'Element Name'. If the error is not resolved, then please contact AUTRON AUTOSAR Support System

ERR0120003 if (none of the ECU Configuration Description File(s) contain the required component)

If none of the ECU Configuration Description File(s) contain the required component

ERR0120004 if (the references provided in the input file are empty)

The Reference path is empty for the parameter '$parameter' in the container '$container', having short name 'ShortName'

ERR0120005 if (The parameter 'Parameter Name' in the container 'Container Name' is not configured)

Parameter 'Parameter Name' in the container 'Container Name' should be configured

ERR0120006 if (AR-RELEASE-VERSION and SW-VERSION present in the Com Module Description template do not follow C syntax)

The value configured for the parameter 'AR-RELEASE-VERSION' and 'SW-VERSION' in the container 'BSW-IMPLEMENTATION' should follow the C Syntax

ERR0120007 if (value of the parameter ComMModeLimitationEnabled is true and ComMNoCom is false.)

Value of the parameter 'ComMNoCom' in the container 'ComMChannel' should not be configured as false, since value of the parameter 'ComMModeLimitationEnabled' in the container 'ComMChannel' is configured as true

ERR0120008 if (Incorrect reference path is configured for the parameter 'Parameter Name' in the container 'Container Name')

The reference path <Reference Path> provided for the parameter 'Parameter Name' in the container 'Container Name' having short name <Short Name> is incorrect.

ERR0120009 if ((Value of the structure element < Min value) || (Value of the structure element > Max value))

The value <Value> of the structure element 'Structure Element Name' in structure 'Structure Name' is not within the range. The value <Value> should be within the range of <Min Value> - <Max Value>, as its data type is <Type>.

ERR0120010 if (Value of the parameter ComMNmLightTimeout is not configured and ComMNmVariant is configured as LIGHT)

Parameter 'ComMNmLightTimeout' in the container 'ComMNetworkManagement' should be configured, since value of the parameter 'ComMNmVariant' in the container ' ComMNetworkManagement ' is configured as LIGHT.

ERR0120011 if (Value of the Parameter 'Parameter Name' in the container 'Container Name' is not unique)

Value configured for the parameter 'Parameter Name's should be unique in 'Container Name' container

ERR0120012 if (Value of the Parameter 'ComMNmVariant' is configured as <NONE> and 'ComMBusType' is

configured as <COMM_BUS_TYPE_INTERNAL>)

Value of the parameter 'ComMNmVariant' present in the container 'ComMNetworkManagement' should be configured as <NONE>, when the value of the parameter 'ComMBusType' present in the container 'ComMChannel' is configured as <COMM_BUS_TYPE_INTERNAL>.

ERR0120013 if (Value of the parameter ComMGlobalNvmBlockDescriptor in the container ComMChannel is true and the parameter ComMGlobalNvmBlockDescriptor in the container ComMGeneral is not configured)

If the parameter 'ComMNoWakeUpInhibitionNvmStorage' in the container 'ComMChannels' is true then the parameter 'ComMGlobalNvMBlockDescriptor' in the container 'ComMGeneral' should have a valid reference

ERR0120014 if (value of the parameter ComMUserChannel and parameter ComMChannelPerPnc references are same)

The reference of parameter 'ComMUserChannel' in the container 'ComMUserPerChannel' and reference of parameter 'ComMChannelPerPnc' in the container 'ComMPnc' should be unique.

ERR0120015 if (Value of the parameter ComMPncComSignalChannelRef is not configured and parameter ComMPncGatewayEnabled is true)

If the parameter 'ComMPncGatewayEnabled ' configured as 'true' in the container 'ComMGeneral' and the Parameter 'ComMPncComSignalChannelRef' in the 'ComMPncComSignal' container should be configured

ERR0120016 if (value of the parameter 'ComMNmVariant' is configured as PASSIVE and the parameter 'ComMModeLimitationEnabled' is configured as false)

Value of the parameter 'ComMNmVariant' in the container 'ComMNetworkManagement' should not be configured as PASSIVE, since value of the parameter 'ComMModeLimitationEnabled' in the container 'ComMGeneral' is configured as false

ERR0120017 if ((ComMPncGatewayType = False) && (ComMPncGatewayType is configured))

The parameter 'ComMPncGatewayType' in the container 'ComMChannels' should not be configured when the parameter 'ComMPncGatewayEnabled' in the container 'ComMGeneral' is False

ERR0120018 if (value of the parameter 'ComMPncGatewayEnabled' is configured as true and the parameter 'ComMPncEnabled' is configured as false)

Value of the parameter 'ComMPncGatewayEnabled' in the container 'ComMGeneral' should not be configured as true, since value of the parameter 'ComMPncEnabled' in the container 'ComMConfigSet' is configured as false.

ERR0120019 if (channel 'ComMManageReference' is configured as managed channel and the parameter ComMConfigSet/ComMChannel/ComMNetworkManagement/ComMPncGatewayType.{value} ' of this channel is configured as COMM_GATEWAY_TYPE_ACTIVE or COMM_GATEWAY_TYPE_PASSIVE)

The managed channel cannot config as COMM_GATEWAY_TYPE_ACTIVE or COMM_GATEWAY_TYPE_PASSIVE type.

ERR0120020 if (channel 'ComMManageReference' is configured as managing channel and the parameter ComMConfigSet/ComMChannel/ComMNetworkManagement/ComMNmVariant.{value} ' of this channel is

difference NM_VARIANT_FULL)

The ComMNmVariant of a managing channel is set to FULL.

ERR0120021 if (channel 'ComMManageReference' is configured as managed channel and the parameter ComMConfigSet/ComMChannel/ComMNetworkManagement/ComMNmVariant.{value} ' of this channel is difference NM_VARIANT_LIGHT)

The ComMNmVariant of a managing channel is set to LIGHT.

ERR0120022 if(ComMPncEnable and ComM_PncSupport are STD_ON and Parameter ComM/ComMConfigSet/ComMChannel/ComMNetworkManagement/ComMBusType.{value} is difference Can or FlexRay)

PNC only support on bus type Can or FlexRay

ERR0120023 If (ComMConfigSet/ComMPnc/ComMPncEthIfSwitchPortGroupRef is True and check all channel of this Pnc if exist the parameter of Struct ComM_Channel.ChannelUserCount > 0)

ComM channel's that are referenced by a PNC are not allowed to be referenced by any ComMUsers

ERR0120024 If((COMM_CHANNEL_SEQUENTIAL is STD_OFF) and (ComM/ComMConfigSet/ComMChannel/ComMChannelId for more than one channel have the same value))

Channel Id of <Channel Short name> and <Channel Short name> should be unique

ERR0120025 If((COMM_USER_SEQUENTIAL is STD_OFF) and (ComM/ComMConfigSet/ComMUser/ComMUserIdentifier for more than one channel have the same value))

Channel Id of <User Short name> and <User Short name> should be unique

ERR0120026 If(ComM/ComMConfigSet/ComMPnc/ComMPncId for more than one channel have the same value)

Channel Id of <Pnc Short name> and <Pnc Short name> should be unique

ERR0120027 Check all channel mapping to Pnc then get the bus type of channel ComM/ComMConfigSet/ComMChannel/ComMBusType.{value}) Channel mapping to this Pnc id is Can please use this condition if (ComMPncId/8 < CanNmPnInfo/CanNmPnInfoOffset.{value}). Channel mapping to this Pnc id is Flexray please use this condition if (ComMPncId/8 < FrNmPnInfo/FrNmPnInfoOffset.{value})

Pnc vector information offset of <BusTypeName> on the Channel id <Channel Id> incorrect.

ERR0120034 If (Post-Build supported and /EcucDefs/EcuC/EcucPostBuildVariants is not configured or /EcucDefs/EcuC is not existed)

The ComM module supported Post-Build but There are no variants configured in ECUC

ERR0120035 If (Post-Build supported and ComM module wasn't included all EcuC post-build variants.)

Mismatch post-build variant with EcuC module, the module's post-build variants should include all EcuC post-build variants.

8. SWP Error Code

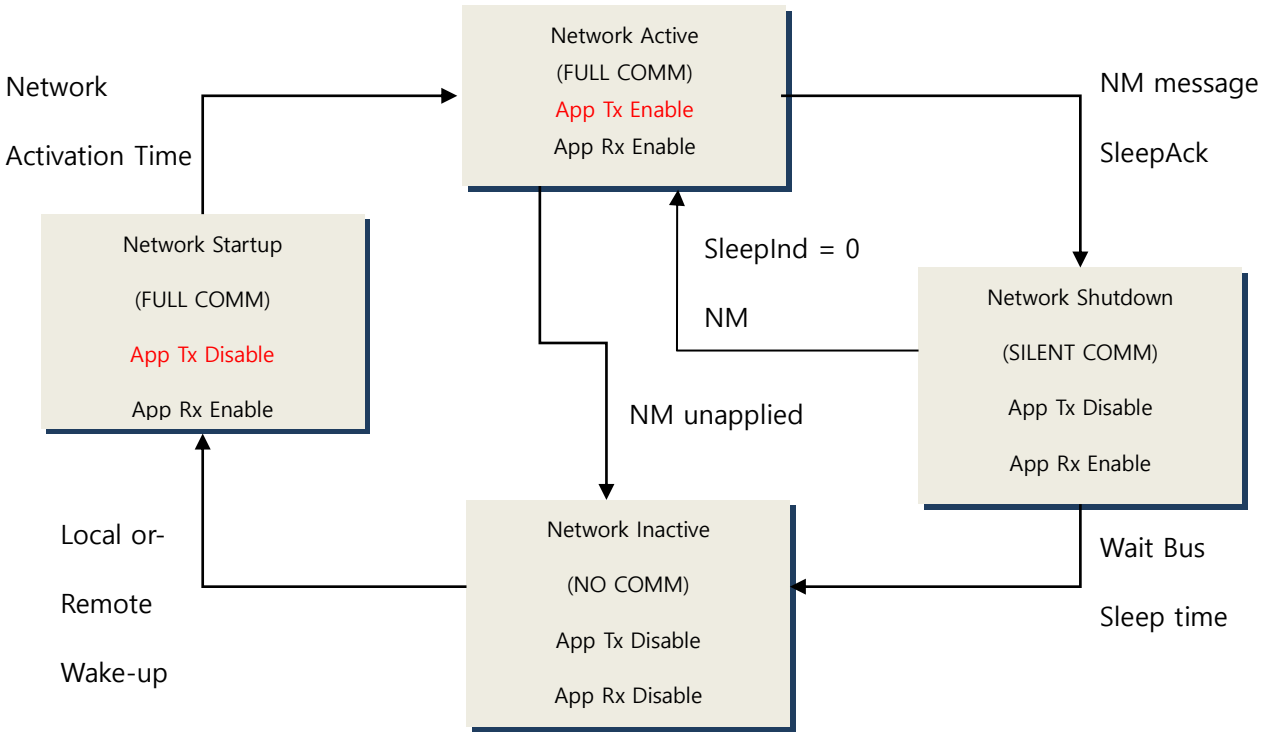
None

9. Appendix

9.1 Setting Guide

9.1.1 Precautions when switching communication mode

A request to switch NO COMMUNICATION for a CAN communication channel must be performed with the channel active. When the Tx Enable point of the Com module is the time when the associated CAN channel is switched to Network Active, refer to the Com UM for how to receive notification.



9.1.2 The communication mode request service settings

When the Harmonization of the ComM module is executed after CAN DB Import, an OperationInvokedEvent for each channel and the provided service associated with it is generated for each channel as shown in the figure below:

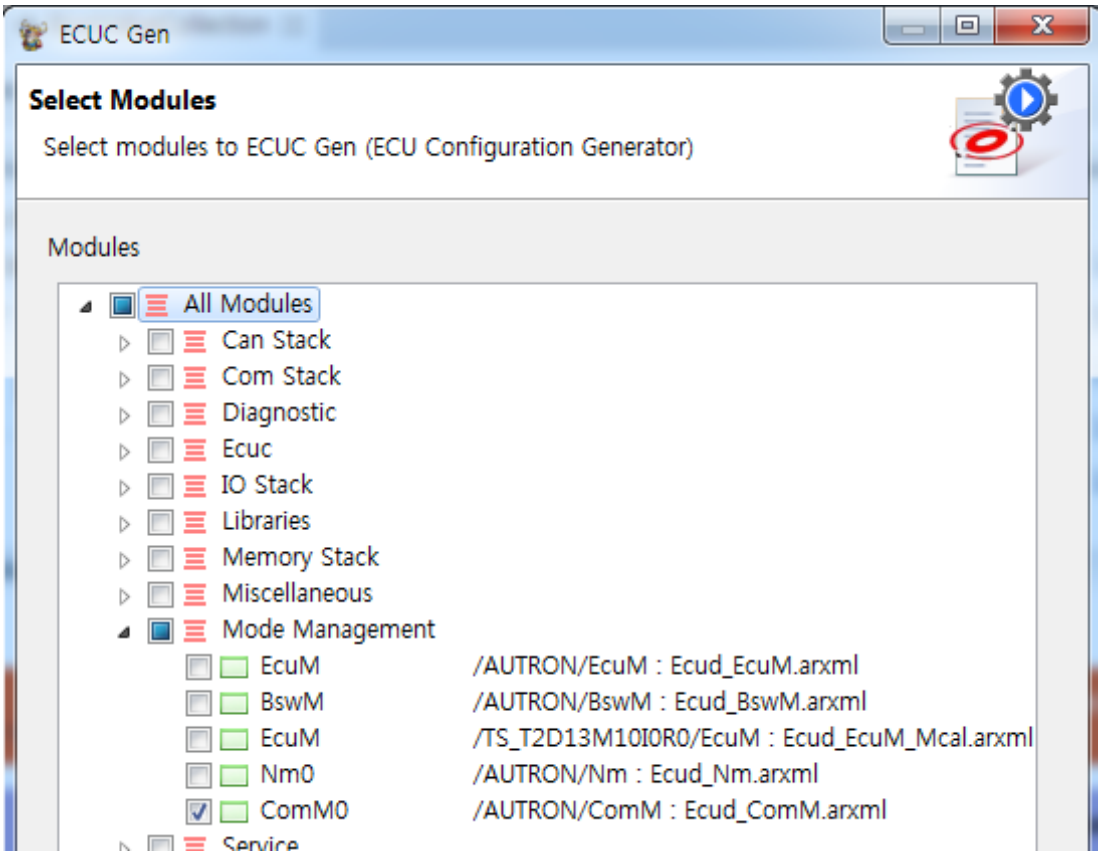


Figure 1: Select ComM module when harmonizing

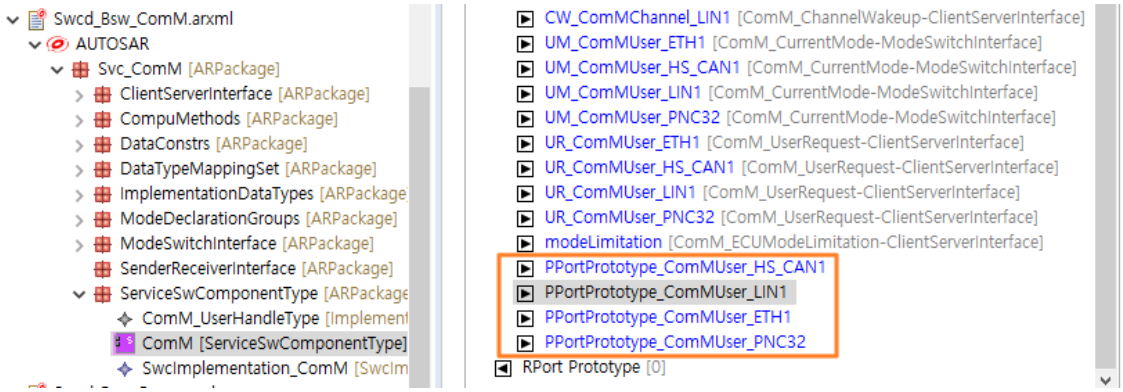


Figure 2: Automatic PPort creation for each harmonized channel

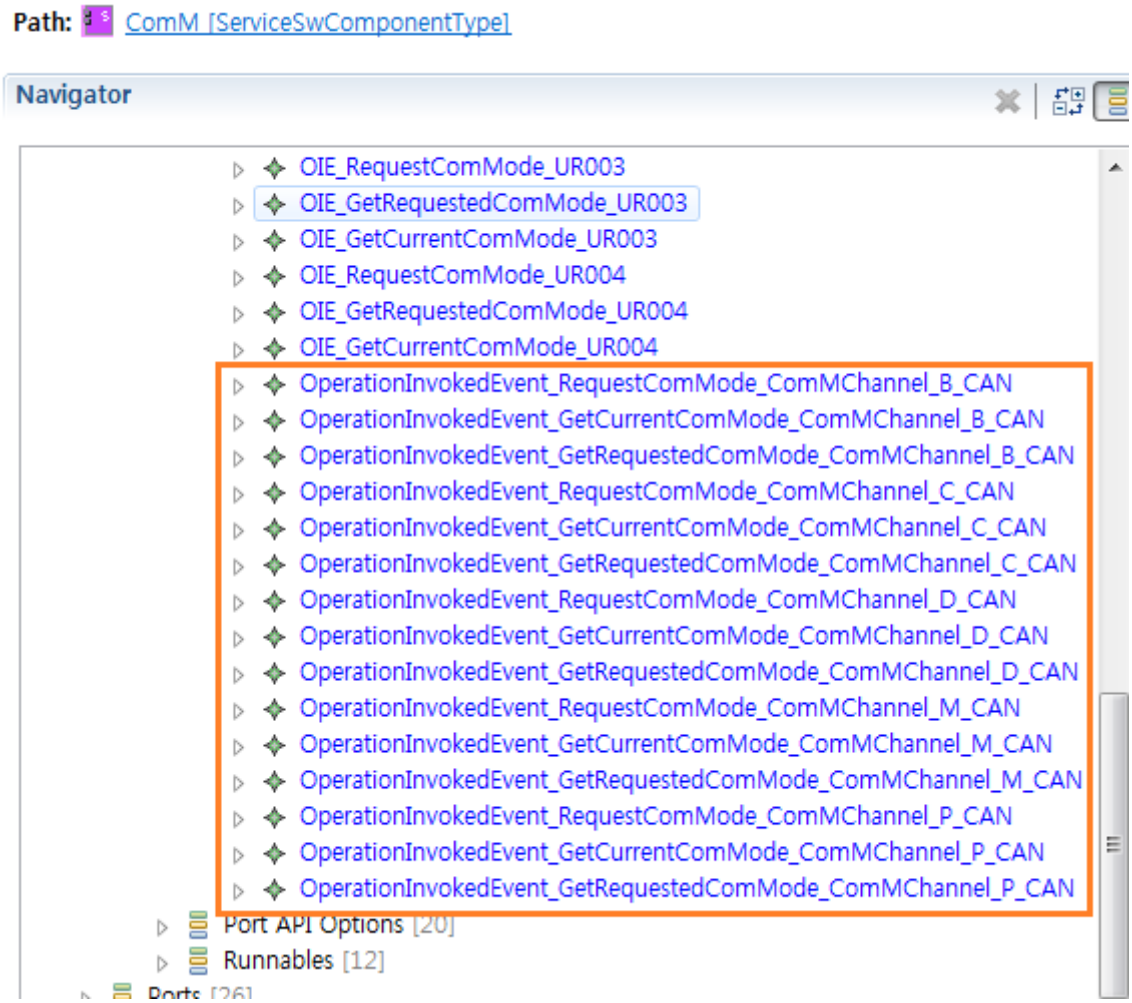


Figure 3: Automatic generation of OperationInvokedEvent that is combined by service and port provided by ComM

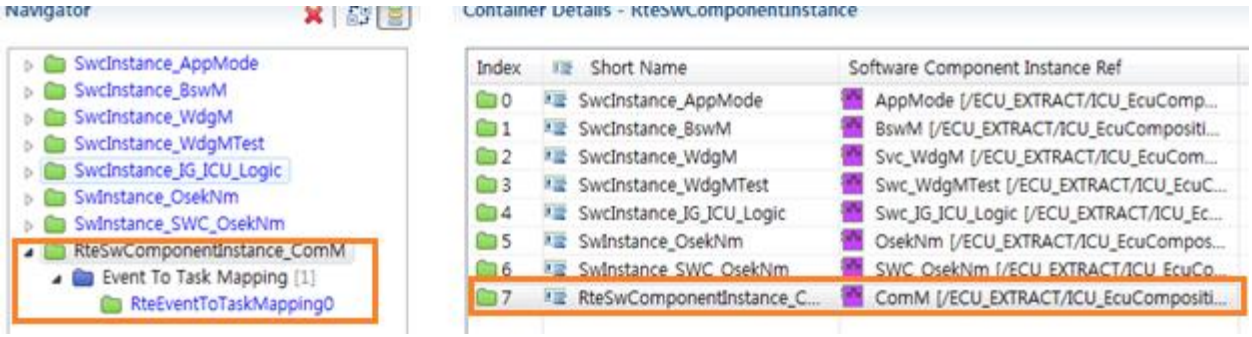


Figure 4: Add ComM SWC Instance to Ecud_Rte.arxml

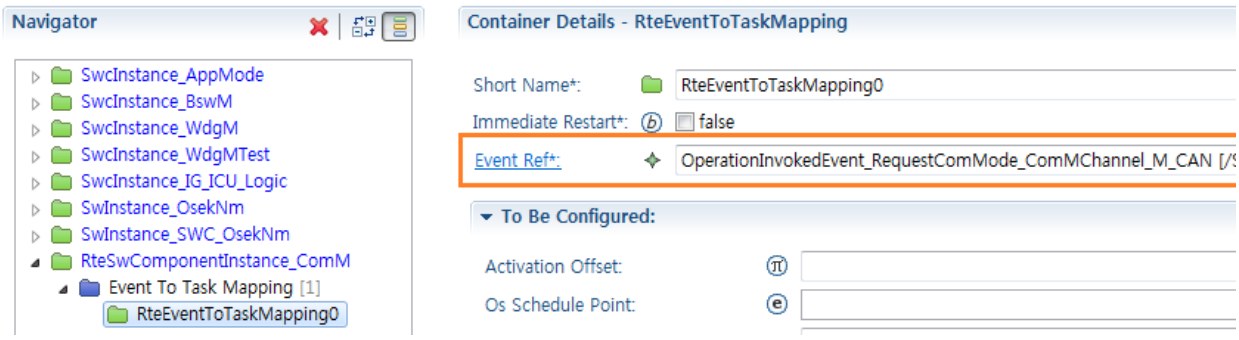


Figure 5: Add the OperationInvokedEvent connected to the added PPort

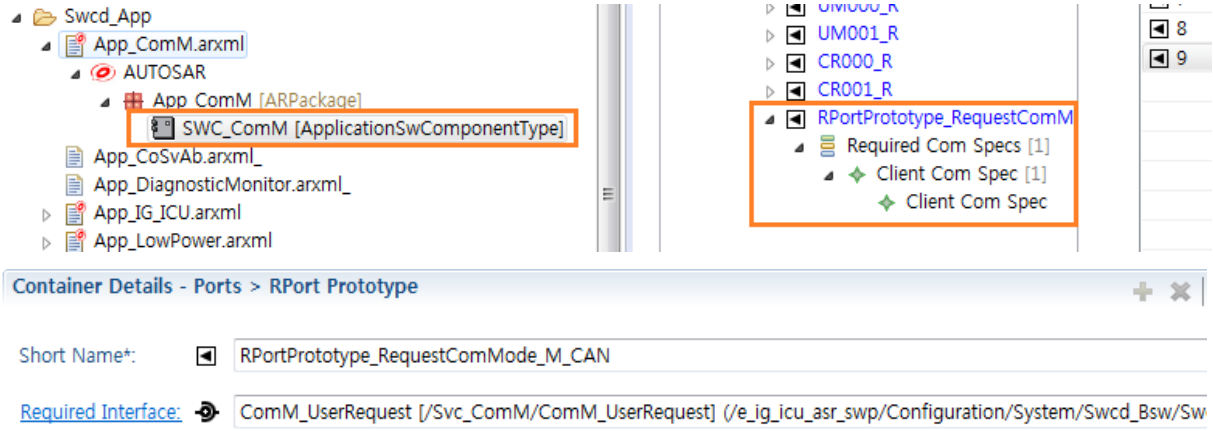


Figure 6: Create RPort in ApplicationSwComponentType to use service

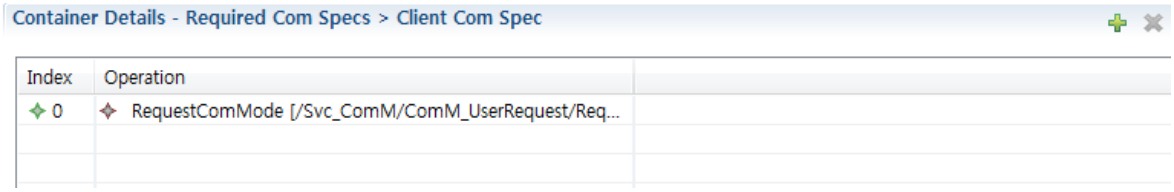


Figure 7: Add the operation you want to call

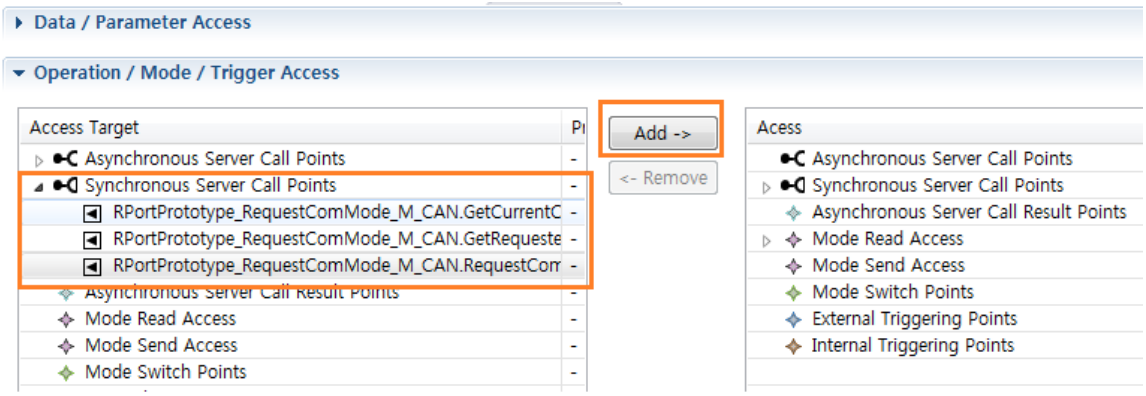


Figure 8: Add Server Call Point to Runnable

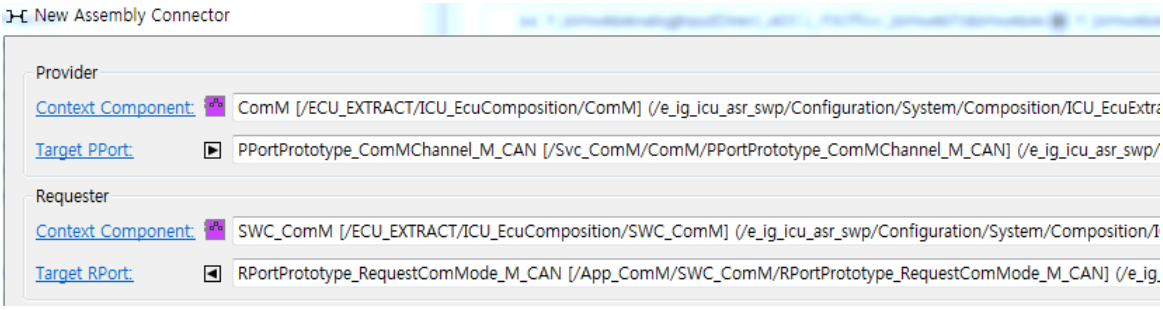


Figure 9: Connect two ports in Ecu Composition

9.2 Use case setting guide

Note that the guides described in this chapter are sample codes and settings

9.2.1 Request Com mode guide

The communication mode request code should not be executed on the interrupt.

The communication mode request code should not be executed in another task that can preempt the task in which the ComM periodic function is executed.

```
#include “Rte_SWC_ComM.h”
/ * Request FULL COMM on M_CAN * /
Rte_Call_RPortPrototype_RequestComMode_M_CAN_RequestComMode
(RTE_MODE_ComMMode_COMM_FULL_COMMUNICATION);
/ * Request NO COMM on M_CAN * /
Rte_Call_RPortPrototype_RequestComMode_M_CAN_RequestComMode
(RTE_MODE_ComMMode_COMM_NO_COMMUNICATION);
```

9.2.2 Troubleshoot Request Com mode problem

This Guide is not to eliminate the cause of the mode change, but rather to eliminate the cause of the mode change, it is a method for mode control. If the request com mode cannot be change because of an error.

1. Initialize the internal state of ComM channel by calling ComM_ResetRequestChannel API [6.3.1.1].
2. No Communication request.
3. After confirming the completion of No Communication mode conversion, request to switch to Full Communication mode
4. Confirm the completion of switching to Full Communication mode.