User Manual

for S32M27X LINTRCV Driver

Document Number: UM34LINTRCVASRR21-11 Rev0000R3.0.0 Rev. 1.0

Chapter 1

Revision History

Revision	Date	Author	Description
1.0	31.03.2023	NXP RTD Team	S32K3 Real-Time Drivers AUTOSAR 4.4 & R21-11 Version 3.0.0

Chapter 2

Introduction

This User Manual describes NXP Semiconductor AUTOSAR LINTRCV for S32K3XX. AUTOSAR LINTRCV driver configuration parameters and deviations from the specification are described in Driver chapter of this document. AUTOSAR LINTRCV driver requirements and APIs are described in the AUTOSAR LINTRCV driver software specification document.

2.1 Supported Derivatives

The software described in this document is intended to be used with the following microcontroller devices of NXP Semiconductors:

- s32m274_lqfp64
- s32m276_lqfp64

All of the above microcontroller devices are collectively named as S32K3.

Note: MWCT part numbers contain NXP confidential IP for Qi Wireless Power.

2.2 Overview

AUTOSAR (AUTomotive Open System ARchitecture) is an industry partnership working to establish standards for software interfaces and software modules for automobile electronic control systems.

AUTOSAR:

- paves the way for innovative electronic systems that further improve performance, safety and environmental friendliness.
- is a strong global partnership that creates one common standard: "Cooperate on standards, compete on implementation".
- is a key enabling technology to manage the growing electrics/electronics complexity. It aims to be prepared for the upcoming technologies and to improve cost-efficiency without making any compromise with respect to quality.
- facilitates the exchange and update of software and hardware over the service life of the vehicle.

2.3 About This Manual

This Technical Reference employs the following typographical conventions:

- **Boldface** style: Used for important terms, notes and warnings.
- *Italic* style: Used for code snippets in the text. Note that C language modifiers such "const" or "volatile" are sometimes omitted to improve readability of the presented code.

Notes and warnings are shown as below:

Note

This is a note.

Warning

This is a warning

2.4 Acronyms and Definitions

Term	Definition
API	Application Programming Interface
ASM	Assembler
BSMI	Basic Software Make file Interface
C/CPP	C and C++ Source Code
DEM	Diagnostic Event Manager
DET	Development Error Tracer
ECUM	ECU state Manager
ECU	Electronic Control Unit
ISR	Interrupt Service Routine
LIN	Local Interconnect Network
LINTRCV	Lin Transceiver Driver
SPI	Serial Peripheral Interface
ICU	Interrupt Control Unit
LSB	Least Signifigant Bit
MCU	Micro Controller Unit
MSB	Most Significant Bit
N/A	Not Applicable
RAM	Random Access Memory
SIU	Systems Integration Unit
SWS	Software Specification
VLE	Variable Length Encoding
XML	Extensible Markup Language

2.5 Reference List

#	Title	Version
1	Specification of LIN Transceiver Driver	AUTOSAR Release R21-11
2	S32M27x Reference Manual	S32M27x Reference Manual, Rev.2, Draft A, — 02/2023
3	Datasheet	S32M2xx Data Sheet, Rev. 2 RC — 12/2022

Chapter 3

Driver

3.1 Requirements

Requirements for this driver are detailed in the Autosar Driver Software Specification document (See Table Reference List).

3.2 Driver Design Summary

The LinTrcv Driver controls the LINPHY in AE of the S32M27x devices. It provides the following features:

- Compliant with LIN Physical Layer 2.2 specification.
- Configuration and initialization of the LINPHY.
- Support Normal and Standby modes.

3.3 Hardware Resources

Driver has one LINPHY channel in AE which is a companion die that is assembled along with an MCU die.

Note

The user must select properly the chip which supports LinPHY: S32M276

3.4 Deviations from Requirements

The driver deviates from the AUTOSAR LIN Driver software specification in some places. The table below identifies the AUTOSAR requirements that are not implemented, not fully implemented or out of scope for the LIN Driver.

Term	Definition
N/S	Out of scope
N/I	Not implemented
N/F	Not fully implemented

Below table identifies the AUTOSAR requirements that are not fully implemented, not implemented or out of scope for the LINTRCV driver.

Requirement	Status	Description	Notes
SWS_LinTrcv_00126	N/I	The function LinTrcv_GetBusWuReason shall return the reason for the wake up that the LIN transceiver has detected in the parameter Reason	Not implemented
SWS_LinTrcv_00066	N/I	Wakeup notification must be supported by Lin Transceiver driver, therefore LIN transceiver driver covers 2 wakeup modes, internal wakeup by an upper layer or external wakeup by LIN channel.	Not implemented
SWS_LinTrcv_00161	N/I	LinTrcv driver shall use the following APIs provided by ICU driver, to enable and disable the wakeup event notification: - Icu_EnableNotification - Icu_Disable↩ Notification	Not implemented
SWS_LinTrcv_00162	N/I	LinTrcv driver shall enable the ICU channels when the transceiver transmits to standby mode (LINTRCV_S↩ TANDBY)	Not implemented
SWS_LinTrcv_00163	N/I	LinTrcv driver shall disable the ICU channels when the transceiver transmits to Normal mode (LINTRCV_N↩ ORMAL)	Not implemented
SWS_LinTrcv_00171	N/I	The LIN Transceiver Driver shall use the Time service Tm_BusyWait1us16bit to realize the wait time for transceiver state changes.	Not implemented
SWS_LinTrcv_00126	N/I	The function LinTrcv_GetBusWuReason shall return the reason for the wake up that the LIN transceiver has detected in the parameter Reason	Not implemented

Requirement	Status	Description	Notes
SWS_LinTrcv_00127	N/I	If there is no/incorrect communication to the transceiver, the function LinTrcv_GetBusWuReason shall return E↩ _NOT_OK.	Not implemented

3.5 Driver Limitations

The limitations of this driver are:

- In External wakeup mode, Wakeup feature only support by polling, not interrupt.
- The LinTrcv_43_AE_GetBusWuReason function can't return the reason for the wake up that the Lin transceiver has detected.

3.6 Driver usage and configuration tips

3.6.1 How to configure the SPI sequence via AE module

Ae module is used to access LIN transceiver hardware connected via SPI. This SPI sequence shalln't configure in LINTRCV module, It have to configure in Ae module.

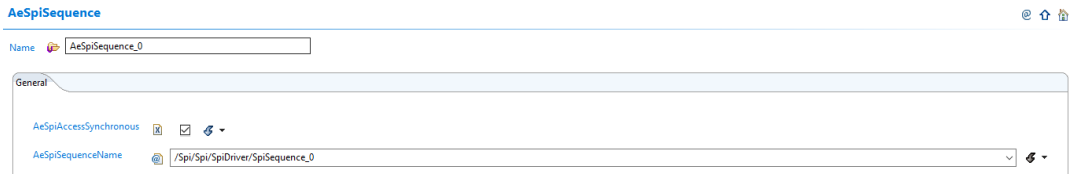

Figure 3.1 Reference to a Spi sequence configuration in Ae module over EB tresos.

SPI Configuration:
Configuration SPI sequence:

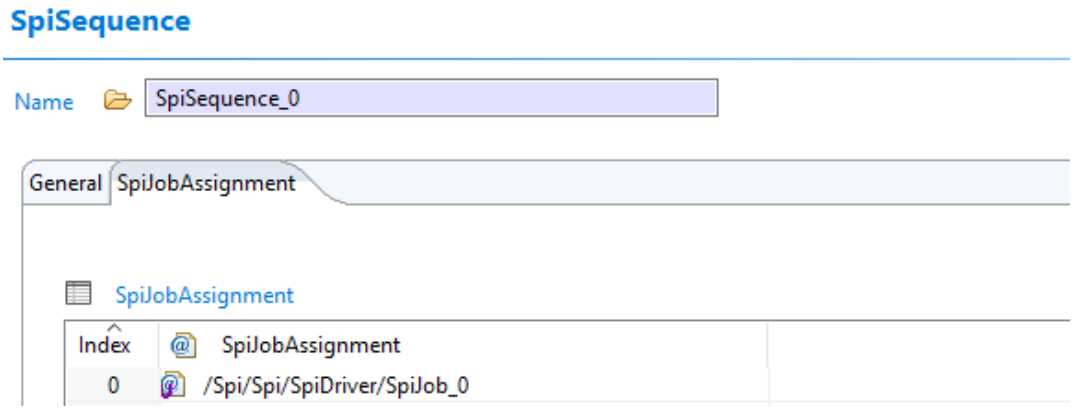

Figure 3.2 Configuration SPI sequence.

Configuration SPI Job:
In Spi Job tab, Need to configure the external device and asignment to Spi channel.

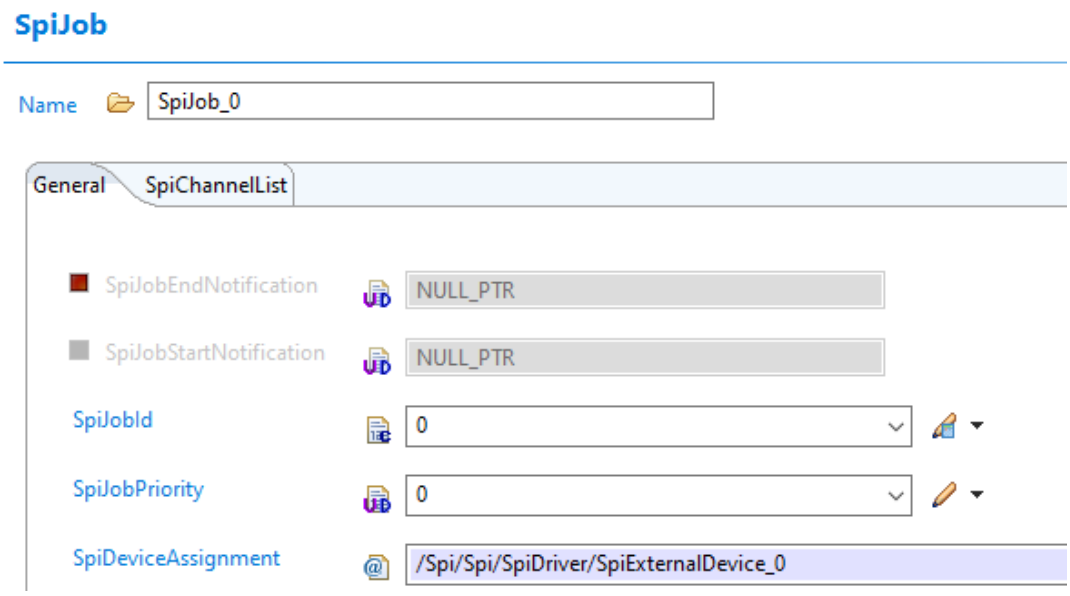

Figure 3.3 Configuration SPI job.

In Spi External Device tab:

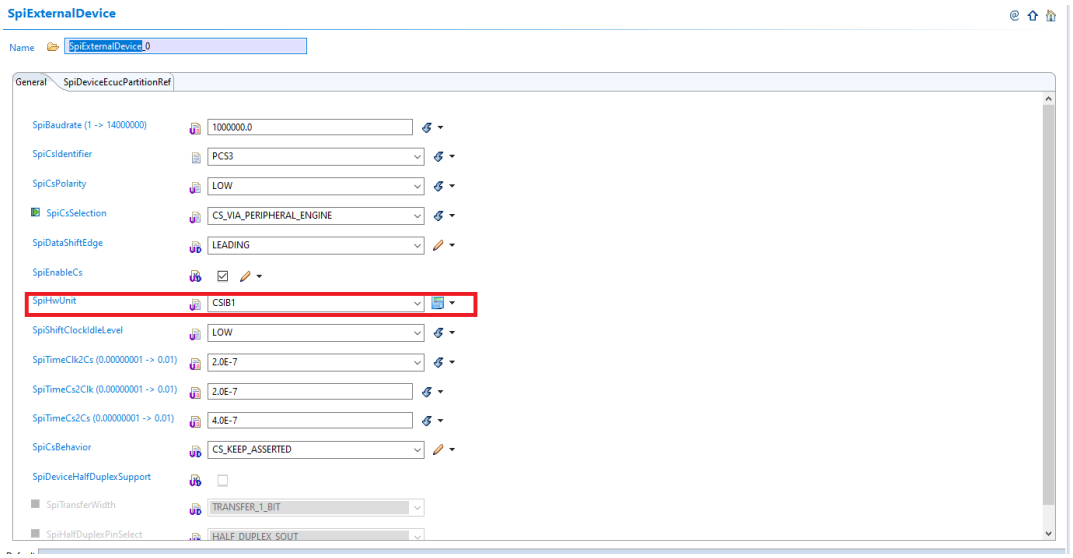

Figure 3.4 Configuration SPI External Device.

Need to make sure that the SpiHwUnit must refer to the Spi Phy Uint tab and the SpiPhyUnitMapping is LPSPI_1

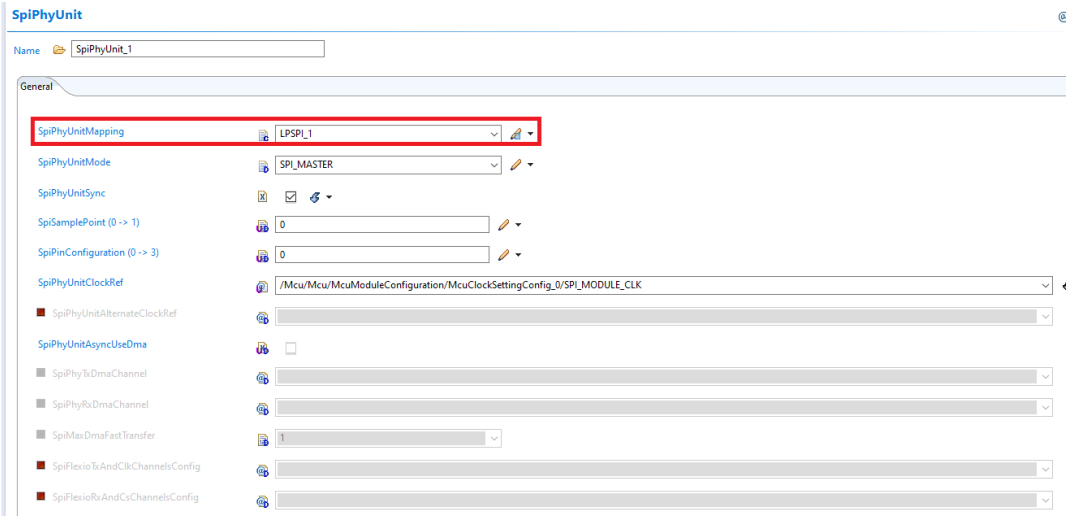

Figure 3.5 Configuration SPI channel.

In SpiChannelList tab:
Need to asignment to the Spi Channel and configure for SpiChannel as below:

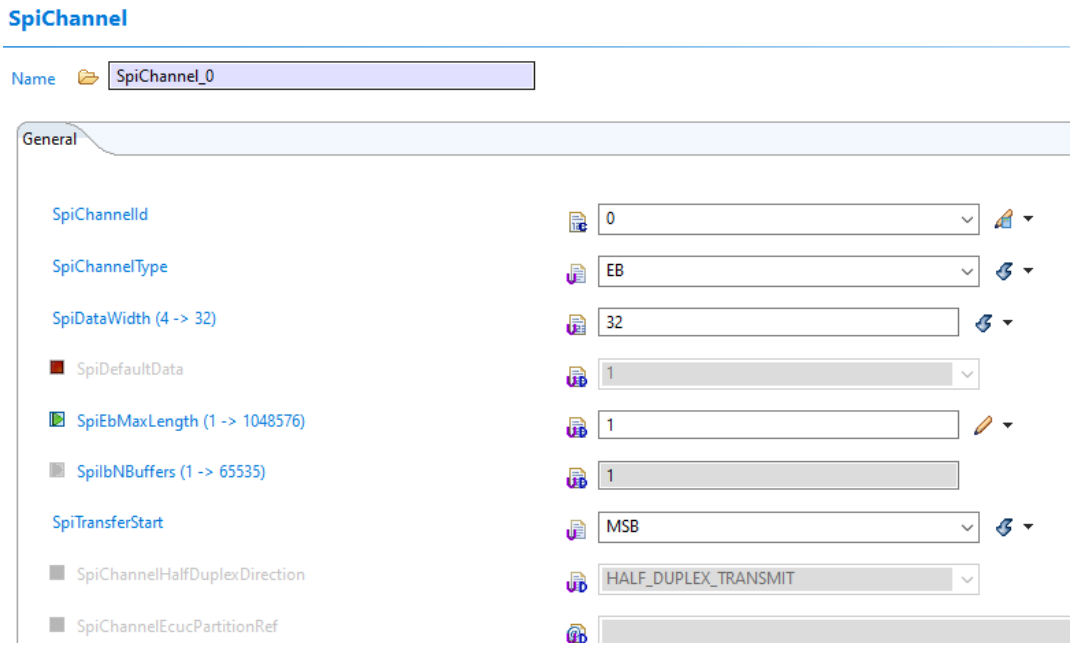

Figure 3.6 Configuration SPI channel.

Port Configuration:
Need to configure Pins for SPI in order communicates with LIN PHY interface.

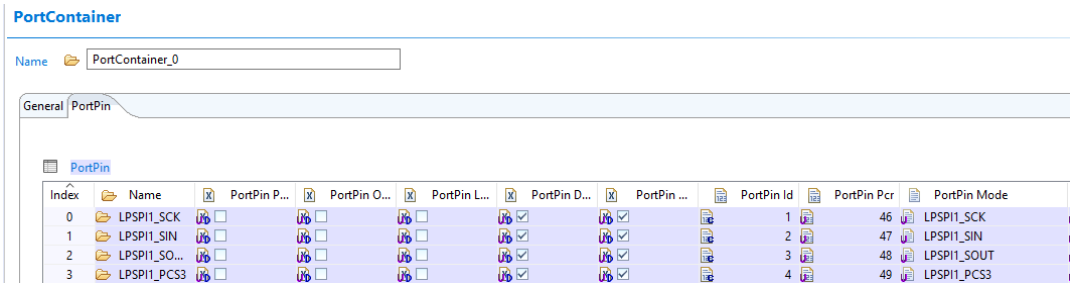

Figure 3.7 Configuration SPI Pins.

MCU Configuration:
Make sure that LPSPI_1 must enable clock.

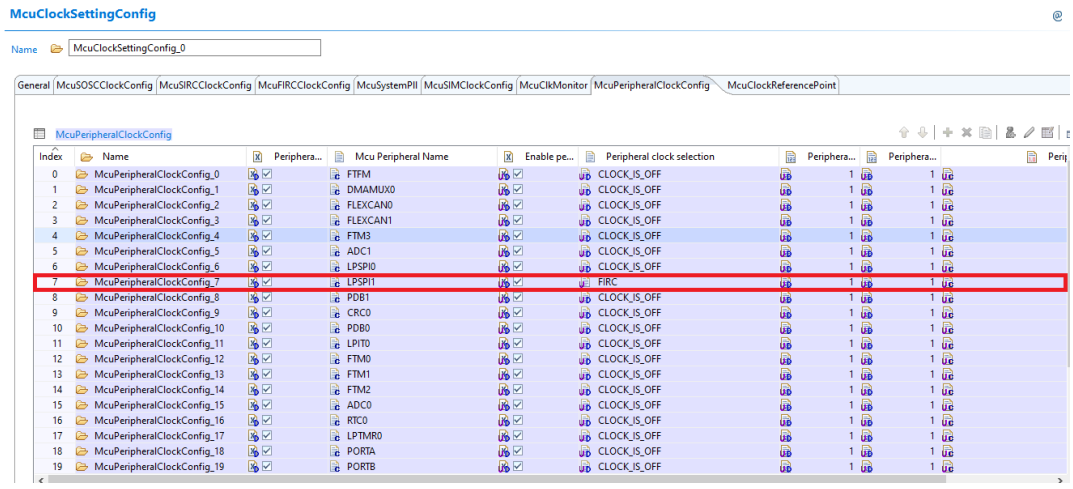

Figure 3.8 Enalbe the clock for LPSPI_1.

3.6.2 Approach about examples of LINTRCV

Please keep in mind that LinTrcv examples, which are Lin driver plugin, will only be available for the HLD layer. They are available for both configurators supported by the project: EB Tresos and Design Studio.

3.7 Runtime errors

None

3.8 Symbolic Names Disclaimer

All containers having symbolicNameValue set to TRUE in the AUTOSAR schema will generate defines like:

#define <Mip>Conf_<Container_ShortName>_<Container_ID>

For this reason it is forbidden to duplicate the names of such containers across the RTD configurations or to use names that may trigger other compile issues (e.g. match existing `#ifdefs` arguments).

Chapter 4

Tresos Configuration Plug-in

This chapter describes the Tresos configuration plug-in for the driver. All the parameters are described below.

- Module LinTrcv
  - Container LinTrcvGeneral
    - Parameter LinTrcvMulticoreSupport
    - Parameter LinTrcvDevErrorDetect
    - Parameter LinTrcvIndex
    - Parameter LinTrcvTimerType
    - Parameter LinTrcvVersionInfoApi
    - Parameter LinTrcvWaitTime
    - Parameter LinTrcvWakeUpSupport
    - Reference LinTrcvEcucPartitionRef
  - Container LinTrcvChannel
    - Parameter LinTrcvChannelId
    - Parameter LinTrcvHwChannel
    - Parameter LinTrcvChannelUsed
    - Parameter LinTrcvWakeupByBusUsed
    - Reference LinTrcvWakeupSourceRef
    - Reference LinTrcvChannelEcucPartitionRef
    - Reference LinTrcvIcuChannelRef
    - Container LinTrcvAccess
      - Container LinTrcvDioAccess
      - Container LinTrcvDioChannelAccess
      - Parameter LinTrcvHardwareInterfaceName
      - Reference LinTrcvDioSymRefName
      - Container LinTrcvSpiSequence
      - Reference LinTrcvSpiSequenceName
  - Container CommonPublishedInformation
    - Parameter ArReleaseMajorVersion
    - Parameter ArReleaseMinorVersion
    - Parameter ArReleaseRevisionVersion

* Parameter ModuleId
* Parameter SwMajorVersion
* Parameter SwMinorVersion
* Parameter SwPatchVersion
* Parameter VendorApiInfix
* Parameter VendorId

4.1 Module LinTrcv

Configuration of LIN Transceiver Driver module.

Included containers:

- LinTrcvGeneral
- LinTrcvChannel
- CommonPublishedInformation

Property	Value
type	ECUC-MODULE-DEF
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantSupport	false
supportedConfigVariants	VARIANT-POST-BUILD, VARIANT-PRE-COMPILE

4.2 Container LinTrcvGeneral

LinTrcvGeneral

Autosar Requirements: ECUC_LinTrcv_00090

Container gives LIN transceiver driver basic information.

Included subcontainers:

- None

Property	Value
type	ECUC-PARAM-CONF-CONTAINER-DEF
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A

4.3 Parameter LinTrcvMulticoreSupport

This parameter determine multi-core feature will be used in Lin driver.

If LinTrcvMulticoreSupport is disabled, then for all the variants no partition shall be defined.

If LinTrcvMulticoreSupport is enabled, at least one EcucPartition needs to be defined (in all variants).

Property	Value
type	ECUC-BOOLEAN-PARAM-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	false
valueConfigClasses	VARIANT-POST-BUILD: PRE-COMPILE
	VARIANT-PRE-COMPILE: PRE-COMPILE
defaultValue	false

4.4 Parameter LinTrcvDevErrorDetect

LinDevErrorDetect

Autosar Requirements: ECUC_LinTrcv_00001

Switches the Default Error Detection and Notification ON or OFF.

Property	Value
type	ECUC-BOOLEAN-PARAM-DEF
origin	AUTOSAR_ECUC
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	false
valueConfigClasses	VARIANT-POST-BUILD: PRE-COMPILE
	VARIANT-PRE-COMPILE: PRE-COMPILE
defaultValue	false

4.5 Parameter LinTrcvIndex

Autosar Requirements: ECUC_LinTrcv_00153

Specifies the InstanceId of this module instance. If only one instance is present it shall have the Id 0.

Note, this parameter is not used in the current implementation.

Property	Value
type	ECUC-INTEGER-PARAM-DEF
origin	AUTOSAR_ECUC
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	false
valueConfigClasses	VARIANT-POST-BUILD: PRE-COMPILE
	VARIANT-PRE-COMPILE: PRE-COMPILE
defaultValue	0
max	255
min	0

4.6 Parameter LinTrcvTimerType

LinTrcvTimerType

Type of the Time Service Predefined Timer.

Property	Value
type	ECUC-ENUMERATION-PARAM-DEF
origin	AUTOSAR_ECUC
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	false
valueConfigClasses	VARIANT-POST-BUILD: PRE-COMPILE
	VARIANT-PRE-COMPILE: PRE-COMPILE
defaultValue	None
literals	['None', 'Timer_1us16bit']

4.7 Parameter LinTrcvVersionInfoApi

LinTrcvVersionInfoApi

Autosar Requirements: ECUC_LinTrcv_00003

Switches the LinTrcvVersionInfo function ON or OFF.

Property	Value
type	ECUC-BOOLEAN-PARAM-DEF
origin	AUTOSAR_ECUC
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	false
valueConfigClasses	VARIANT-POST-BUILD: PRE-COMPILE
	VARIANT-PRE-COMPILE: PRE-COMPILE
defaultValue	false

4.8 Parameter LinTrcvWaitTime

LinTrcvWaitTime

Autosar Requirements: ECUC_LinTrcv_00160

Wait time for transceiver state changes in seconds.

Property	Value
type	ECUC-FLOAT-PARAM-DEF
origin	AUTOSAR_ECUC
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	false
valueConfigClasses	VARIANT-POST-BUILD: PRE-COMPILE
	VARIANT-PRE-COMPILE: PRE-COMPILE
defaultValue	0.0
max	2.55E-4
min	0.0

4.9 Parameter LinTrcvWakeUpSupport

LinTrcvWakeUpSupport

Autosar Requirements: ECUC_LinTrcv_00107

Informs whether wake up is supported or not. In case wake up is not supported by LIN transceiver hardware the setting shall be false. The wake up ability may be switched on or off for each channel of one LIN transceiver by LinTrcvWakeupSourceRef.

Property	Value
type	ECUC-BOOLEAN-PARAM-DEF
origin	AUTOSAR_ECUC
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	false
valueConfigClasses	VARIANT-POST-BUILD: PRE-COMPILE
	VARIANT-PRE-COMPILE: PRE-COMPILE
defaultValue	false

4.10 Reference LinTrcvEcucPartitionRef

Maps the Lin transceiver driver to zero or multiple ECUC partitions to make the modules API available in this partition. The Lin transceiver driver will operate as an independent instance in each of the partitions.

Property	Value
type	ECUC-REFERENCE-DEF
origin	AUTOSAR_ECUC
lowerMultiplicity	0
upperMultiplicity	Infinite
postBuildVariantMultiplicity	true
multiplicityConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: PRE-COMPILE
postBuildVariantValue	true
valueConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: PRE-COMPILE
requiresSymbolicNameValue	False
destination	/AUTOSAR/EcucDefs/EcuC/EcucPartitionCollection/EcucPartition

4.11 Container LinTrcvChannel

Container gives LIN transceiver driver information about a single LIN transceiver channel. Any LIN transceiver driver has such LIN transceiver channels"

Included subcontainers:

- LinTrcvAccess

Property	Value
type	ECUC-PARAM-CONF-CONTAINER-DEF
lowerMultiplicity	1
upperMultiplicity	Infinite
postBuildVariantMultiplicity	false
multiplicityConfigClasses	VARIANT-POST-BUILD: PRE-COMPILE
	VARIANT-PRE-COMPILE: PRE-COMPILE

4.12 Parameter LinTrcvChannelId

Unique identifier of the LIN Transceiver Channel.

Property	Value
type	ECUC-INTEGER-PARAM-DEF
origin	AUTOSAR_ECUC
symbolicNameValue	true
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	false
valueConfigClasses	VARIANT-POST-BUILD: PRE-COMPILE
	VARIANT-PRE-COMPILE: PRE-COMPILE
defaultValue	1
max	255
min	0

4.13 Parameter LinTrcvHwChannel

Selects the Lin Transceiver Hardware Channel.

Note:

This Parameter is an Implementation Specific Parameter.

Property	Value
type	ECUC-ENUMERATION-PARAM-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	false
valueConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: PRE-COMPILE
defaultValue	AELINPHY_0
literals	['AELINPHY_0']

4.14 Parameter LinTrcvChannelUsed

LinTrcvChannelUsed

Autosar Requirements: ECUC_LinTrcv_00004

Shall the related LIN transceiver channel be used?

Property	Value
type	ECUC-BOOLEAN-PARAM-DEF
origin	AUTOSAR_ECUC
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	false
valueConfigClasses	VARIANT-POST-BUILD: PRE-COMPILE
	VARIANT-PRE-COMPILE: PRE-COMPILE
defaultValue	true

4.15 Parameter LinTrcvWakeupByBusUsed

LinTrcvWakeupByBusUsed

Autosar Requirements: ECUC_LinTrcv_00006

Is wake up by bus supported? If LIN transceiver hardware does not support wake up by bus value is always FALSE. If LIN transceiver hardware supports wake up by bus value is TRUE or FALSE depending whether it is used or not

Property	Value
type	ECUC-BOOLEAN-PARAM-DEF
origin	AUTOSAR_ECUC
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	false
valueConfigClasses	VARIANT-POST-BUILD: PRE-COMPILE
	VARIANT-PRE-COMPILE: PRE-COMPILE
defaultValue	true

4.16 Reference LinTrcvWakeupSourceRef

LinTrcvWakeupByBusUsed

Autosar Requirements: ECUC_LinTrcv_00012

Reference to a wakeup source in the EcuM configuration. This reference is only needed if LinTrcvWakeupByBusUsed is true. Implementation Type: reference to EcuM_WakeupSourceType

Property	Value
type	ECUC-REFERENCE-DEF
origin	AUTOSAR_ECUC
lowerMultiplicity	0
upperMultiplicity	1
postBuildVariantMultiplicity	false
multiplicityConfigClasses	VARIANT-POST-BUILD: PRE-COMPILE
	VARIANT-PRE-COMPILE: PRE-COMPILE
postBuildVariantValue	false
valueConfigClasses	VARIANT-POST-BUILD: PRE-COMPILE
	VARIANT-PRE-COMPILE: PRE-COMPILE
requiresSymbolicNameValue	False
destination	/AUTOSAR/EcucDefs/EcuM/EcuMConfiguration/EcuMCommon↩ Configuration/EcuMWakeupSource

4.17 Reference LinTrcvChannelEcucPartitionRef

Maps one single Lin transceiver channel to zero or one ECUC partitions.

The ECUC partition referenced is a subset of the ECUC partitions where the Lin transceiver driver is mapped to.

Property	Value
type	ECUC-REFERENCE-DEF
origin	AUTOSAR_ECUC
lowerMultiplicity	0
upperMultiplicity	1
postBuildVariantMultiplicity	true
multiplicityConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: PRE-COMPILE
postBuildVariantValue	true
valueConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: PRE-COMPILE
requiresSymbolicNameValue	False
destination	/AUTOSAR/EcucDefs/EcuC/EcucPartitionCollection/EcucPartition

4.18 Reference LinTrcvIcuChannelRef

Reference to the IcuChannel to enable/disable the interrupts for wakeups.

Property	Value
type	ECUC-REFERENCE-DEF
origin	AUTOSAR_ECUC
lowerMultiplicity	0
upperMultiplicity	1
postBuildVariantMultiplicity	true
multiplicityConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: PRE-COMPILE
postBuildVariantValue	true
valueConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: PRE-COMPILE
requiresSymbolicNameValue	False
destination	/AUTOSAR/EcucDefs/Icu/IcuConfigSet/IcuChannel

4.19 Container LinTrcvAccess

Container gives LIN transceiver driver access about a single LIN transceiver channel.

Included subcontainers:

- LinTrcvDioAccess
- LinTrcvSpiSequence

Property	Value
type	ECUC-PARAM-CONF-CONTAINER-DEF
lowerMultiplicity	0
upperMultiplicity	1
postBuildVariantMultiplicity	false
multiplicityConfigClasses	N/A

4.20 Container LinTrcvDioAccess

Container gives LIN transceiver driver information about accessing ports and port pins. In addition relation between LIN transceiver hardware pin names and Dio port access information is given. If a LIN transceiver hardware has no Dio interface, there is no instance of this container.

Included subcontainers:

- LinTrcvDioChannelAccess

Property	Value
type	ECUC-PARAM-CONF-CONTAINER-DEF
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A

4.21 Container LinTrcvDioChannelAccess

Container gives DIO channel access by single Lin transceiver channel.

Included subcontainers:

- None

Property	Value
type	ECUC-PARAM-CONF-CONTAINER-DEF
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A

4.22 Parameter LinTrcvHardwareInterfaceName

LIN transceiver hardware interface name. It is typically the name of a pin. From a Dio point of view it is either a port, a single channel or a channel group. Depending on this fact either LINTRCV_DIO_PORT_SYMBOLIC_NAME or LINTRCV_DIO_CHANNEL_SYMBOLIC_NAME or LINTRCV_DIO_CHANNEL_GROUP_SYMBOLIC_NAME shall reference a Dio configuration. The LIN transceiver driver implementation description shall list up this name for the appropriate LIN transceiver hardware.

Property	Value
type	ECUC-STRING-PARAM-DEF
origin	AUTOSAR_ECUC
symbolicNameValue	False
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	false
valueConfigClasses	VARIANT-POST-BUILD: PRE-COMPILE
	VARIANT-PRE-COMPILE: PRE-COMPILE
defaultValue	N/A

4.23 Reference LinTrcvDioSymRefName

Choice Reference to a DIO Port, DIO Channel or DIO Channel Group. This reference replaces the LINTRCV_DIO_PORT_SYM_NAME, LINTRCV_DIO_CHANNEL_SYM_NAME and LINTRCV_DIO_GROUP_SYM_NAM references in the Lin Trcv SWS.

Property	Value
type	ECUC-CHOICE-REFERENCE-DEF
origin	AUTOSAR_ECUC
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	false
valueConfigClasses	VARIANT-POST-BUILD: PRE-COMPILE
	VARIANT-PRE-COMPILE: PRE-COMPILE
requiresSymbolicNameValue	False
destinations	['/AUTOSAR/EcucDefs/Dio/DioConfig/DioPort/DioChannel', '/AUTOSA↩ R/EcucDefs/Dio/DioConfig/DioPort/DioChannelGroup', '/AUTOSAR/Ecuc↩ Defs/Dio/DioConfig/DioPort']

4.24 Container LinTrcvSpiSequence

Container gives LIN transceiver driver information about one SPI sequence.

One SPI sequence used by LIN transceiver driver is in exclusive use for it.

No other driver is allowed to access this sequence. LIN transceiver driver may use one sequence to access n LIN transceiver hardwares chips of the same type or n sequences are used to access one single LIN transceiver hardware chip.

If a LIN transceiver hardware has no SPI interface, there is no instance of this container.

Included subcontainers:

- None

Property	Value
type	ECUC-PARAM-CONF-CONTAINER-DEF
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A

4.25 Reference LinTrcvSpiSequenceName

Reference to a Spi sequence configuration container.

Property	Value
type	ECUC-REFERENCE-DEF
origin	AUTOSAR_ECUC
lowerMultiplicity	0
upperMultiplicity	1
postBuildVariantMultiplicity	false
multiplicityConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: PRE-COMPILE
postBuildVariantValue	true
valueConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: PRE-COMPILE
requiresSymbolicNameValue	False
destination	/AUTOSAR/EcucDefs/Spi/SpiDriver/SpiSequence

4.26 Container CommonPublishedInformation

Common container, aggregated by all modules. It contains published information about vendor and versions.

Included subcontainers:

- None

Property	Value
type	ECUC-PARAM-CONF-CONTAINER-DEF
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A

4.27 Parameter ArReleaseMajorVersion

Major version number of AUTOSAR specification on which the appropriate implementation is based on.

Property	Value
type	ECUC-INTEGER-PARAM-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	false
valueConfigClasses	VARIANT-POST-BUILD: PUBLISHED-INFORMATION
	VARIANT-PRE-COMPILE: PUBLISHED-INFORMATION
defaultValue	4
max	4
min	4

4.28 Parameter ArReleaseMinorVersion

Minor version number of AUTOSAR specification on which the appropriate implementation is based on.

Property	Value
type	ECUC-INTEGER-PARAM-DEF

Property	Value
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	false
valueConfigClasses	VARIANT-POST-BUILD: PUBLISHED-INFORMATION
	VARIANT-PRE-COMPILE: PUBLISHED-INFORMATION
defaultValue	7
max	7
min	7

4.29 Parameter ArReleaseRevisionVersion

Revision version number of AUTOSAR specification on which the appropriate implementation is based on.

Property	Value
type	ECUC-INTEGER-PARAM-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	false
valueConfigClasses	VARIANT-POST-BUILD: PUBLISHED-INFORMATION
	VARIANT-PRE-COMPILE: PUBLISHED-INFORMATION
defaultValue	0
max	0
min	0

4.30 Parameter ModuleId

Module ID of this module from Module List.

Property	Value
type	ECUC-INTEGER-PARAM-DEF
origin	NXP
symbolicNameValue	false

Property	Value
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	false
valueConfigClasses	VARIANT-POST-BUILD: PUBLISHED-INFORMATION
	VARIANT-PRE-COMPILE: PUBLISHED-INFORMATION
defaultValue	64
max	64
min	64

4.31 Parameter SwMajorVersion

Major version number of the vendor specific implementation of the module. The numbering is vendor specific.

Property	Value
type	ECUC-INTEGER-PARAM-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	false
valueConfigClasses	VARIANT-POST-BUILD: PUBLISHED-INFORMATION
	VARIANT-PRE-COMPILE: PUBLISHED-INFORMATION
defaultValue	3
max	3
min	3

4.32 Parameter SwMinorVersion

Minor version number of the vendor specific implementation of the module. The numbering is vendor specific.

Property	Value
type	ECUC-INTEGER-PARAM-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1

Property	Value
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	false
valueConfigClasses	VARIANT-POST-BUILD: PUBLISHED-INFORMATION
	VARIANT-PRE-COMPILE: PUBLISHED-INFORMATION
defaultValue	0
max	0
min	0

4.33 Parameter SwPatchVersion

Patch level version number of the vendor specific implementation of the module. The numbering is vendor specific.

Property	Value
type	ECUC-INTEGER-PARAM-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	false
valueConfigClasses	VARIANT-POST-BUILD: PUBLISHED-INFORMATION
	VARIANT-PRE-COMPILE: PUBLISHED-INFORMATION
defaultValue	0
max	0
min	0

4.34 Parameter VendorApiInfix

In driver modules which can be instantiated several times on a single ECU, BSW00347 requires that the name of APIs is extended by the VendorId and a vendor specific name.

This parameter is used to specify the vendor specific name. In total, the implementation specific name is generated as follows:

<ModuleName>_>VendorId>_<VendorApiInfix>.

E.g. assuming that the VendorId of the implementor is 123 and the implementer chose a VendorApiInfix of "v11r456" a api name Can_Write defined in the SWS will translate to Can_123_v11r456Write.

This parameter is mandatory for all modules with upper multiplicity > 1. It shall not be used for modules with upper multiplicity =1.

Property	Value
type	ECUC-STRING-PARAM-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	0
upperMultiplicity	1
postBuildVariantMultiplicity	false
multiplicityConfigClasses	VARIANT-POST-BUILD: PUBLISHED-INFORMATION
	VARIANT-PRE-COMPILE: PUBLISHED-INFORMATION
postBuildVariantValue	false
valueConfigClasses	VARIANT-POST-BUILD: PUBLISHED-INFORMATION
	VARIANT-PRE-COMPILE: PUBLISHED-INFORMATION
defaultValue	AE

4.35 Parameter VendorId

Vendor ID of the dedicated implementation of this module according to the AUTOSAR vendor list.

Property	Value
type	ECUC-INTEGER-PARAM-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	false
valueConfigClasses	VARIANT-POST-BUILD: PUBLISHED-INFORMATION
	VARIANT-PRE-COMPILE: PUBLISHED-INFORMATION
defaultValue	43
max	43
min	43

Chapter 5

Module Index

5.1 Software Specification

Here is a list of all modules:

Chapter 6

Module Documentation

6.1 AELINPHY_LINTRCV_IP

6.1.1 Detailed Description

Function Reference

- AeLinPhy_LinTrcv_Ip_StatusType AeLinPhy_LinTrcv_Ip_Init (const uint8 Instance, const AeLinPhy_LinTrcv_Ip_Use
  *LinTrcvUserConfig)
- AeLinPhy_LinTrcv_Ip_StatusType AeLinPhy_LinTrcv_Ip_Deinit (const uint8 Instance)
- AeLinPhy_LinTrcv_Ip_StatusType AeLinPhy_LinTrcv_Ip_SetMode (const uint8 Instance, const AeLinPhy_LinTrcv_Ip_TransceiverModeType OpMode)
- AeLinPhy_LinTrcv_Ip_StatusType AeLinPhy_LinTrcv_Ip_GetMode (const uint8 Instance, AeLinPhy_LinTrcv_Ip_Tra
  *TransceiverMode)
- AeLinPhy_LinTrcv_Ip_StatusType AeLinPhy_LinTrcv_Ip_SetWakeupMode (const uint8 Instance, AeLinPhy_LinTrcv_Ip_TrcvWakeupModeType TrcvWakupMode)
- AeLinPhy_LinTrcv_Ip_StatusType AeLinPhy_LinTrcv_Ip_CheckWakeupFlag (const uint8 Instance)

6.1.2 Function Reference

6.1.2.1 AeLinPhy_LinTrcv_Ip_Init()

```
AeLinPhy_LinTrcv_Ip_StatusType AeLinPhy_LinTrcv_Ip_Init (
            const uint8 Instance,
            const AeLinPhy_LinTrcv_Ip_UserConfigType * LinTrcvUserConfig )
```

AeLinPhy_LinTrcv_Ip_Init

6.1.2.2 AeLinPhy_LinTrcv_Ip_Deinit()

```
AeLinPhy_LinTrcv_Ip_StatusType AeLinPhy_LinTrcv_Ip_Deinit (
            const uint8 Instance )
```

AeLinPhy_LinTrcv_Ip_Deinit

6.1.2.3 AeLinPhy_LinTrcv_Ip_SetMode()

```
AeLinPhy_LinTrcv_Ip_StatusType AeLinPhy_LinTrcv_Ip_SetMode (
            const uint8 Instance,
            const AeLinPhy_LinTrcv_Ip_TransceiverModeType OpMode )
```

AeLinPhy_LinTrcv_Ip_SetMode

6.1.2.4 AeLinPhy_LinTrcv_Ip_GetMode()

```
AeLinPhy_LinTrcv_Ip_StatusType AeLinPhy_LinTrcv_Ip_GetMode (
            const uint8 Instance,
            AeLinPhy_LinTrcv_Ip_TransceiverModeType * TransceiverMode )
```

AeLinPhy_LinTrcv_Ip_GetMode

6.1.2.5 AeLinPhy_LinTrcv_Ip_SetWakeupMode()

```
AeLinPhy_LinTrcv_Ip_StatusType AeLinPhy_LinTrcv_Ip_SetWakeupMode (
            const uint8 Instance,
            AeLinPhy_LinTrcv_Ip_TrcvWakeupModeType TrcvWakupMode )
```

AeLinPhy_LinTrcv_Ip_SetWakeupMode

6.1.2.6 AeLinPhy_LinTrcv_Ip_CheckWakeupFlag()

```
AeLinPhy_LinTrcv_Ip_StatusType AeLinPhy_LinTrcv_Ip_CheckWakeupFlag (
            const uint8 Instance )
```

AeLinPhy_LinTrcv_Ip_CheckWakeupFlag

6.2 AeLinPhy LinTrcv IPL

6.2.1 Detailed Description

Data Structures

- struct AeLinPhy_LinTrcv_Ip_UserConfigType
  User configuration structure of the LINTRCV driver. *More...*

Enum Reference

- enum AeLinPhy_LinTrcv_Ip_TransceiverModeType
  Enum containing the events related to a mode of Lin Tracv.
- enum AeLinPhy_LinTrcv_Ip_StatusType
  AELINPHY status type.
- enum AeLinPhy_LinTrcv_Ip_TrcvWakeupModeType
  Enum containing the events related to a wakeup mode of Lin Tracv.

6.2.2 Data Structure Documentation

6.2.2.1 struct AeLinPhy_LinTrcv_Ip_UserConfigType

User configuration structure of the LINTRCV driver.

Definition at line 127 of file AeLinPhy_LinTrcv_Ip_Types.h.

Data Fields

- uint8 Instance
  Hardware Instance.

6.2.2.1.1 Field Documentation

6.2.2.1.1.1 Instance `uint8 Instance`

Hardware Instance.

Definition at line 129 of file AeLinPhy_LinTrcv_Ip_Types.h.

6.2.3 Enum Reference

6.2.3.1 AeLinPhy_LinTrcv_Ip_TransceiverModeType

`enum AeLinPhy_LinTrcv_Ip_TransceiverModeType`

Enum containing the events related to a mode of Lin Tracv.

This enum defines types for an enumerating event related to mode of Lin Tracv.

Enumerator

AELINPHY_LINTRCV_IP_OFF_MODE	The LIN Physical Layer is fully disabled. No wake-up functionality is available.
AELINPHY_LINTRCV_IP_RECV_ONLY_MODE	Entering this mode disables the transmitter and immediately stops any on-going transmission, The receiver is running in full performance mode.
AELINPHY_LINTRCV_IP_NORMAL_MODE	The full functionality is available. Both receiver and transmitter are enabled.
AELINPHY_LINTRCV_IP_STANDBY_MODE	The transmitter of the LIN Physical Layer is disabled and the receiver enters a low power mode in order to be able to detect valid wake-up pulses on the LIN bus.

Definition at line 83 of file AeLinPhy_LinTrcv_Ip_Types.h.

6.2.3.2 AeLinPhy_LinTrcv_Ip_StatusType

enum AeLinPhy_LinTrcv_Ip_StatusType

AELINPHY status type.

Enumerator

AELINPHY_LINTRCV_IP_STATUS_ERROR	No error. General error, command failed to execute task successfully.

Definition at line 95 of file AeLinPhy_LinTrcv_Ip_Types.h.

6.2.3.3 AeLinPhy_LinTrcv_Ip_TrcvWakeupModeType

enum AeLinPhy_LinTrcv_Ip_TrcvWakeupModeType

Enum containing the events related to a wakeup mode of Lin Tracv.

This enum defines types for an enumerating event related to wakeup mode of Lin Tracv.

Enumerator

AELINPHY_LINTRCV_IP_WUMODE_ENABLE	The notification for wakeup events is enabled on the addressed network.
AELINPHY_LINTRCV_IP_WUMODE_DISABLE	The notification for wakeup events is disabled on the addressed network.
AELINPHY_LINTRCV_IP_WUMODE_CLEAR	A stored wakeup event is cleared on the addressed network.

Definition at line 110 of file AeLinPhy_LinTrcv_Ip_Types.h.

6.3 LINTRCV Driver

6.3.1 Detailed Description

Macros

- #define LINTRCV_43_AE_NOT_ACTIVE

  LINTRCV driver states.
- #define LINTRCV_43_AE_ACTIVE

  LINTRCV driver states.
- #define LINTRCV_43_AE_INIT_ID

  API service ID for LinTrcv_43_AE_Init() function.
- #define LINTRCV_43_AE_SETOPMODE_ID

  API service ID for LinTrcv_43_AE_SetOpMode() function.
- #define LINTRCV_43_AE_GETOPMODE_ID

  API service ID for LinTrcv_43_AE_GetOpMode() function.
- #define LINTRCV_43_AE_GETBUSWUREASON_ID

  API service ID for LinTrcv_43_AE_GetBusWuReason() function.
- #define LINTRCV_43_AE_GETVERSIONINFO_ID

  API service ID for LinTrcv_43_AE_GetVersionInfo() function.
- #define LINTRCV_43_AE_CHECKWAKEUP_ID

  API service ID for LinTrcv_43_AE_CheckWakeup() function.
- #define LINTRCV_43_AE_SETWAKEUPMODE_ID

  API service ID for LinTrcv_43_AE_SetWakeupMode() function.

Function Reference

- void LinTrcv_43_AE_Init (const LinTrcv_43_AE_ConfigType *ConfigPtr)

  Initializes the LINTRCV module.
- Std_ReturnType LinTrcv_43_AE_SetOpMode (uint8 LinNetwork, LinTrcv_TrcvModeType OpMode)

  The internal state of the LIN transceiver driver is switched to mode given in the parameter OpMode.
- Std_ReturnType LinTrcv_43_AE_GetOpMode (uint8 LinNetwork, LinTrcv_TrcvModeType *OpMode)

  API detects the actual software state of LIN transceiver driver.
- Std_ReturnType LinTrcv_43_AE_GetBusWuReason (uint8 LinNetwork, LinTrcv_TrcvWakeupReasonType *Reason)

  This API provides the reason for the wakeup that the LIN transceiver has detected in the parameter "Reason". The ability to detect and differentiate the possible wakeup reasons depends strongly on the LIN transceiver hardware.
- Std_ReturnType LinTrcv_43_AE_CheckWakeup (uint8 LinNetwork)

  Notifies the calling function if a wakeup is detected.
- Std_ReturnType LinTrcv_43_AE_SetWakeupMode (uint8 LinNetwork, LinTrcv_TrcvWakeupModeType TrcvWakupMode)

  This API enables, disables and clears the notification for wakeup events on the addressed network.

6.3.2 Macro Definition Documentation

6.3.2.1 LINTRCV_43_AE_NOT_ACTIVE

```
#define LINTRCV_43_AE_NOT_ACTIVE
```

LINTRCV driver states.

The state NOT_ACTIVE means that the LinTrcv module has not been initialized yet and cannot be used.

Definition at line 208 of file LinTrcv_43_AE.h.

6.3.2.2 LINTRCV_43_AE_ACTIVE

```
#define LINTRCV_43_AE_ACTIVE
```

LINTRCV driver states.

The ACTIVE state indicates that the LINTRCV driver has been initialized, making each available channel ready for service.

Definition at line 217 of file LinTrcv_43_AE.h.

6.3.2.3 LINTRCV_43_AE_INIT_ID

```
#define LINTRCV_43_AE_INIT_ID
```

API service ID for LinTrcv_43_AE_Init() function.

Parameters used when raising an error or exception.

Definition at line 227 of file LinTrcv_43_AE.h.

6.3.2.4 LINTRCV_43_AE_SETOPMODE_ID

```
#define LINTRCV_43_AE_SETOPMODE_ID
```

API service ID for LinTrcv_43_AE_SetOpMode() function.

Parameters used when raising an error or exception.

Definition at line 237 of file LinTrcv_43_AE.h.

6.3.2.5 LINTRCV_43_AE_GETOPMODE_ID

```
#define LINTRCV_43_AE_GETOPMODE_ID
```

API service ID for LinTrcv_43_AE_GetOpMode() function.

Parameters used when raising an error or exception.

Definition at line 245 of file LinTrcv_43_AE.h.

6.3.2.6 LINTRCV_43_AE_GETBUSWUREASON_ID

```
#define LINTRCV_43_AE_GETBUSWUREASON_ID
```

API service ID for LinTrcv_43_AE_GetBusWuReason() function.

Parameters used when raising an error or exception.

Definition at line 253 of file LinTrcv_43_AE.h.

6.3.2.7 LINTRCV_43_AE_GETVERSIONINFO_ID

```
#define LINTRCV_43_AE_GETVERSIONINFO_ID
```

API service ID for LinTrcv_43_AE_GetVersionInfo() function.

Parameters used when raising an error or exception.

Definition at line 261 of file LinTrcv_43_AE.h.

6.3.2.8 LINTRCV_43_AE_CHECKWAKEUP_ID

```
#define LINTRCV_43_AE_CHECKWAKEUP_ID
```

API service ID for LinTrcv_43_AE_CheckWakeup() function.

Parameters used when raising an error or exception.

Definition at line 270 of file LinTrcv_43_AE.h.

6.3.2.9 LINTRCV_43_AE_SETWAKEUPMODE_ID

```
#define LINTRCV_43_AE_SETWAKEUPMODE_ID
```

API service ID for LinTrcv_43_AE_SetWakeupMode() function.

Parameters used when raising an error or exception.

Definition at line 278 of file LinTrcv_43_AE.h.

6.3.3 Function Reference

6.3.3.1 LinTrcv_43_AE_Init()

```
void LinTrcv_43_AE_Init (
            const LinTrcv_43_AE_ConfigType * ConfigPtr )
```

Initializes the LINTRCV module.

Parameters

in	*ConfigPtr*	- Pointer to the selected configuration set.

Returns

void

Precondition

-

6.3.3.2 LinTrcv_43_AE_SetOpMode()

```
Std_ReturnType LinTrcv_43_AE_SetOpMode (
            uint8 LinNetwork,
            LinTrcv_TrcvModeType OpMode )
```

The internal state of the LIN transceiver driver is switched to mode given in the parameter OpMode.

Parameters

in	*LinNetwork*	LIN network to wich API call has to be applied
in	*OpMode*	The parameter says to which operation mode the change shall be performed.

Returns

Std_ReturnType.

Return values

E_OK	will be returned if the transceiver state has been changed to the requested mode.
E_NOT_OK	will be returned if the transceiver state change is not accepted or has failed or the parameter is out of the allowed range.

6.3.3.3 LinTrcv_43_AE_GetOpMode()

```
Std_ReturnType LinTrcv_43_AE_GetOpMode (
            uint8 LinNetwork,
            LinTrcv_TrcvModeType * OpMode )
```

API detects the actual software state of LIN transceiver driver.

Parameters

in	*LinNetwork*	LIN network to wich API call has to be applied
out	*OpMode*	Pointer to operation mode of the bus the API is applied to.

Returns

Std_ReturnType.

Return values

E_OK	will be returned if the operation mode is detected.
E_NOT_OK	will be returned, if service request is failed due to development errors or the operation mode is not detected.

6.3.3.4 LinTrcv_43_AE_GetBusWuReason()

```
Std_ReturnType LinTrcv_43_AE_GetBusWuReason (
            uint8 LinNetwork,
            LinTrcv_TrcvWakeupReasonType * Reason )
```

This API provides the reason for the wakeup that the LIN transceiver has detected in the parameter "Reason". The ability to detect and differentiate the possible wakeup reasons depends strongly on the LIN transceiver hardware.

Parameters

in	*LinNetwork*	LIN network to wich API call has to be applied
out	*Reason*	Pointer to wakeup reason of the bus the API is applied to.

Returns

Std_ReturnType.

Return values

E_OK	will be returned if the wake up reason is detected.
E_NOT_OK	will be returned, if service request is failed due to development errors or the wakeup reason is not detected..

6.3.3.5 LinTrcv_43_AE_CheckWakeup()

```
Std_ReturnType LinTrcv_43_AE_CheckWakeup (
            uint8 LinNetwork )
```

Notifies the calling function if a wakeup is detected.

Parameters

in	*LinNetwork*	LIN network to which API call has to be applied.

Returns

Std_ReturnType.

Return values

E_NOT_OK	Will be returned, if no wakeup has been detected
E_OK	Will be returned, if a wakeup has been detected.

Precondition

: LinTrcv_43_AE_Init function must be called before this API.

6.3.3.6 LinTrcv_43_AE_SetWakeupMode()

```
Std_ReturnType LinTrcv_43_AE_SetWakeupMode (
            uint8 LinNetwork,
            LinTrcv_TrcvWakeupModeType TrcvWakupMode )
```

This API enables, disables and clears the notification for wakeup events on the addressed network.

Parameters

in	*LinNetwork*	LIN network to which API call has to be applied.
in	*TrcvWakupMode*	Requested transceiver wakup reason.

Returns

Std_ReturnType.

Return values

E_NOT_OK	will be returned, if service request is failed due to development errors or the wakeup mode is not set
E_OK	will be returned if the transceiver state has been changed to the requested mode..

Precondition

: LinTrcv_43_AE_Init function must be called before this API.

6.4 LINTRCV Driver

6.4.1 Detailed Description

Data Structures

- struct LinTrcv_43_AE_ChannelConfigType
  LINTRCV channel configuration type structure. *More...*
- struct LinTrcv_43_AE_ConfigType
  LINTRCV driver configuration type structure. *More...*

6.4.2 Data Structure Documentation

6.4.2.1 struct LinTrcv_43_AE_ChannelConfigType

LINTRCV channel configuration type structure.

This is the type of the external data structure containing the overall initialization data for one LINTRCV Channel. A pointer to such a structure is provided to the LINTRCV channel initialization routine for configuration of the LINTRCV hardware channel.

Definition at line 120 of file LinTrcv_43_AE_Types.h.

Data Fields

Type	Name	Description
const uint8	LinTrcvChannelID	
const LinTrcv_43_AE_HwConfigType *	ChannelConfigPtr	Lin Channel ID. !<
const uint32	ChannelCoreId	LINTRCV Hardware configuration pointer. !<
const boolean	AllocatedPartition	LINTRCV Channel core id. !<
const boolean	LinTrcvChannelUsed	LINTRCV Channel is allocated partition or not. !<

6.4.2.2 struct LinTrcv_43_AE_ConfigType

LINTRCV driver configuration type structure.

This is the type of the pointer to the external data LINTRCV Channels. A pointer of such a structure is provided to the LINTRCV driver initialization routine for configuration of the LINTRCV hardware channel.

Definition at line 150 of file LinTrcv_43_AE_Types.h.

Data Fields

- const LinTrcv_43_AE_ChannelConfigType * LinTrcv_43_AE_ChannelPtr [LINTRCV_43_AE_HW_M↩ AX_MODULES]

  Partition core id is assigned for this configuration.

6.4.2.2.1 Field Documentation

6.4.2.2.1.1 LinTrcv_43_AE_ChannelPtr `const LinTrcv_43_AE_ChannelConfigType* LinTrcv_43_AE_↩ ChannelPtr[LINTRCV_43_AE_HW_MAX_MODULES]`

Partition core id is assigned for this configuration.

!<

Hardware channel.

Constant pointer of the constant external data structure containing the overall initialization data for all the configured LINTRCV Channels.

Definition at line 159 of file LinTrcv_43_AE_Types.h.

How to Reach Us:

Home Page:
nxp.com

Web Support:
nxp.com/support

Information in this document is provided solely to enable system and software implementers to use NXP products. There are no express or implied copyright licenses granted hereunder to design or fabricate any integrated circuits based on the information in this document. NXP reserves the right to make changes without further notice to any products herein.

NXP makes no warranty, representation, or guarantee regarding the suitability of its products for any particular purpose, nor does NXP assume any liability arising out of the application or use of any product or circuit, and specifically disclaims any and all liability, including without limitation consequential or incidental damages. "Typical" parameters that may be provided in NXP data sheets and/or specifications can and do vary in different applications, and actual performance may vary over time. All operating parameters, including "typicals," must be validated for each customer application by customer's technical experts. NXP does not convey any license under its patent rights nor the rights of others. NXP sells products pursuant to standard terms and conditions of sale, which can be found at the following address: nxp.com/SalesTermsandConditions.