User Manual

for S32K3 SENT Driver

Document Number: UM34SENTASRR21-11 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 SENT for S32K3XX. SENT driver configuration parameters and deviations from the specification are described in Driver chapter of this document. SENT driver requirements and APIs are described in the SENT 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:

- s32k310_mqfp100
- s32k310_lqfp48
- s32k311_mqfp100 / MWCT2015S_mqfp100
- s32k311_lqfp48
- s32k312_mqfp100 / MWCT2016S_mqfp100
- s32k312_mqfp172 / MWCT2016S_mqfp172
- s32k314_mqfp172
- s32k314_mapbga257
- s32k322_mqfp100 / MWCT2D16S_mqfp100
- s32k322_mqfp172 / MWCT2D16S_mqfp172

- s32k324_mqfp172 / MWCT2D17S_mqfp172
- s32k324_mapbga257
- s32k341_mqfp100
- s32k341_mqfp172
- s32k342_mqfp100
- s32k342_mqfp172
- s32k344_mqfp172
- s32k344_mapbga257
- s32k394_mapbga289
- s32k396_mapbga289
- s32k358_mqfp172
- s32k358_mapbga289
- s32k328_mqfp172
- s32k328_mapbga289
- s32k338_mqfp172
- s32k338_mapbga289
- s32k348_mqfp172
- s32k348_mapbga289
- 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
AUTOSAR	Automotive Open System Architecture
ASM	Assembler
BSW	Basic Software
DEM	Diagnostic Event Manager
DET	Development Error Tracer
C/CPP	C and C++ Source Code
ECU	Electronic Control Unit
Sent	Single Edge Nibble Transmission
ISR	Interrupt Service Routine
N/A	Not Applicable
VLE	Variable Length Encoding

2.5 Reference List

#	Title	Version
1	S32K3XX Reference Manual	S32K3xx Reference Manual, Rev.6 Draft B, 01/2023
2	S32K396 Reference Manual	S32K39 and S32K37 Reference Manual, Rev. 2 Draft A, 11/2022
3	S32M27x Reference Manual	S32M27x Reference Manual, Rev.2, Draft A - 02/2023
4	S32K3XX Datasheet	S32K3xx Data Sheet, Rev. 6, Draft B. 01/2023
5	S32K396 Datasheet	S32K396 Data Sheet, Rev. 1.1, 08/2022
6	S32M2xx Datasheet	S32M2xx Data Sheet, Rev. 2 RC — 12/2022
7	S32K358 Errata	S32K358 Mask Set Errata for Mask 0P14E, Rev. 28, 9/2022
8	S32K311 Errata	S32K311 Mask Set Errata S32K311_0P98C, Rev. 6/March/2023, 3/2023
9	S32K396 Errata	SS32K396 Mask Set Errata for Mask 0P40E, Rev. DEC2022, 12/2022
10	S32K312 Errata	S32K312 Mask Set Errata for Mask 0P09C, Rev. 25/April/2022
11	S32K342 Errata	S32K342 Mask Set Errata for Mask 0P97C, Rev. 10 11/2022
12	S32K3x4 Errata	S32K3x4 Mask Set Errata for Mask 0P55A/1P55A, Rev. 14/Oct/2022

Chapter 3

Driver

3.1 Requirements

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

For CDD: Sent is a Complex Device Driver (CDD), so there are no AUTOSAR requirements regarding this module.

It has vendor-specific requirements and implementation.

3.2 Driver Design Summary

The Sent driver is implemented as an Autosar complex device driver. It uses the Flexio hardware peripheral which provides support for implementing the Sent protocol. The driver offers a hardware independent API to the upper layer that can be used to configure the Sent and initiate data transfers for both Fast messages and Serial messages (both Enhanced and Short). Hardware and software settings can be configured using an Autosar standard configuration tool. The information required for an Sent data transfer will be configured in a data structure that will be sent as parameter to the API of the driver. The driver reports errors to the error manager as defined in AUTOSAR.

3.3 Hardware Resources

The hardware configured by the Sent driver is the same as for the Flexio IP.

The Flexio Channels that can be configured for this driver: CHANNEL_0,CHANNEL_1,CHANNEL_2,CHANN↩
EL_3,CHANNEL_4,CHANNEL_5,CHANNEL_6,CHANNEL_7

3.4 Deviations from Requirements

Sent is a Complex Device Driver (CDD), so there are no AUTOSAR requirements regarding this module. It has vendor-specific requirements and implementation.

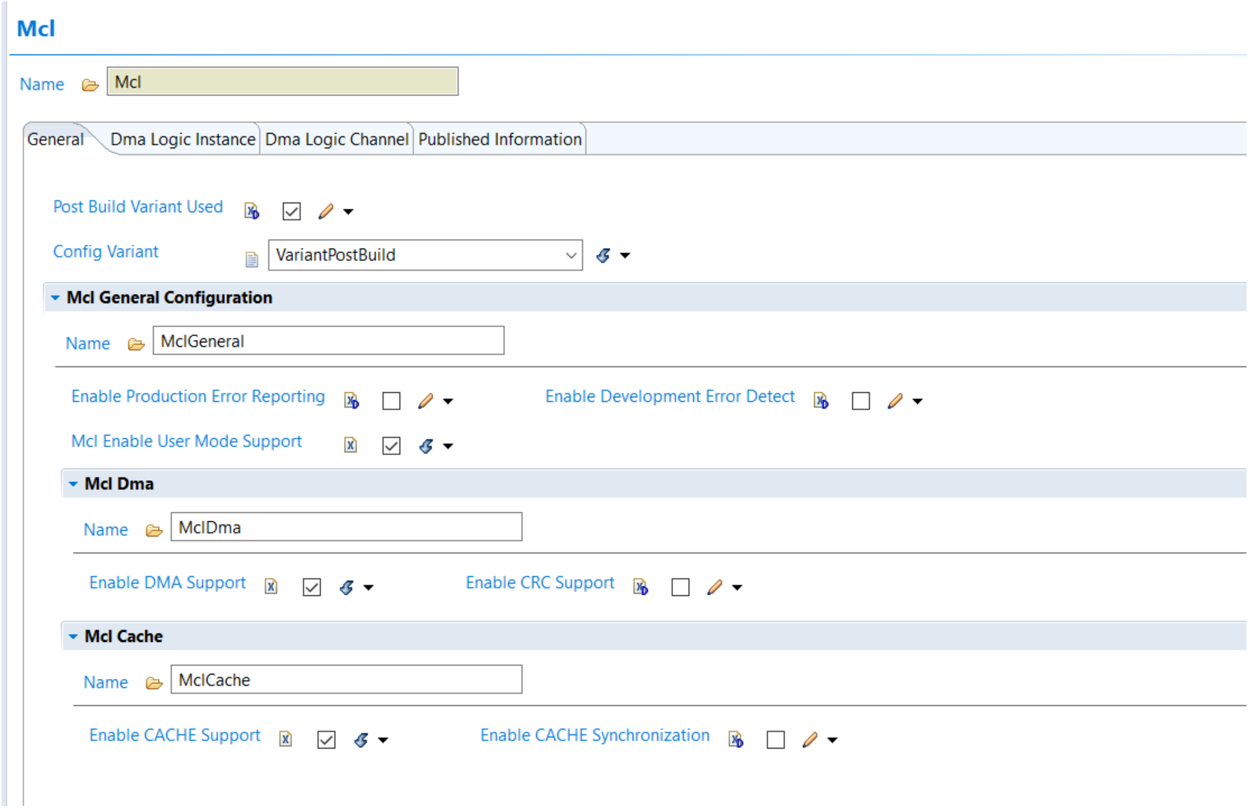
3.5 Driver Limitations

For S32K388 the receiving of data through FlexIO channel cannot be done in VDK due to the fact that FlexIo serial protocol emulation is not supported in VDK.

3.6 Driver usage and configuration tips

- Steps to configure Sent using DMA mode

1. Enable check "Enable Dma Support" from "General" tab of Mcl component.



Mcl

Name: Mcl

Post Build Variant Used ☑

Config Variant: VariantPostBuild

Mcl General Configuration

Name: MclGeneral

Enable Production Error Reporting □ Enable Development Error Detect □

Mcl Enable User Mode Support ☑

Mcl Dma

Name: MclDma

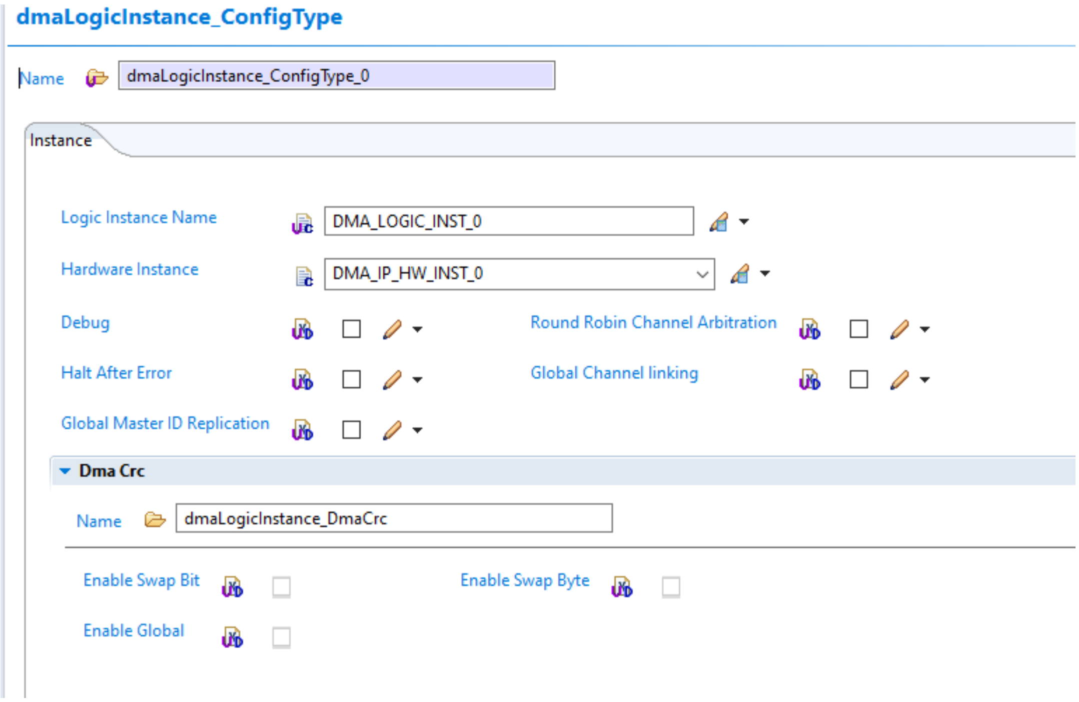
Enable DMA Support ☑ Enable CRC Support □

Mcl Cache

Name: MclCache

Enable CACHE Support ☑ Enable CACHE Synchronization □

2. Configure dma logic instance from "Dma logic instance" tab. Create two instances, one for transmit and one for receive transfers.



dmaLogicInstance_ConfigType

Name: dmaLogicInstance_ConfigType_0

Instance

Logic Instance Name: DMA_LOGIC_INST_0

Hardware Instance: DMA_IP_HW_INST_0

Debug □ Round Robin Channel Arbitration □

Halt After Error □ Global Channel linking □

Global Master ID Replication □

Dma Crc

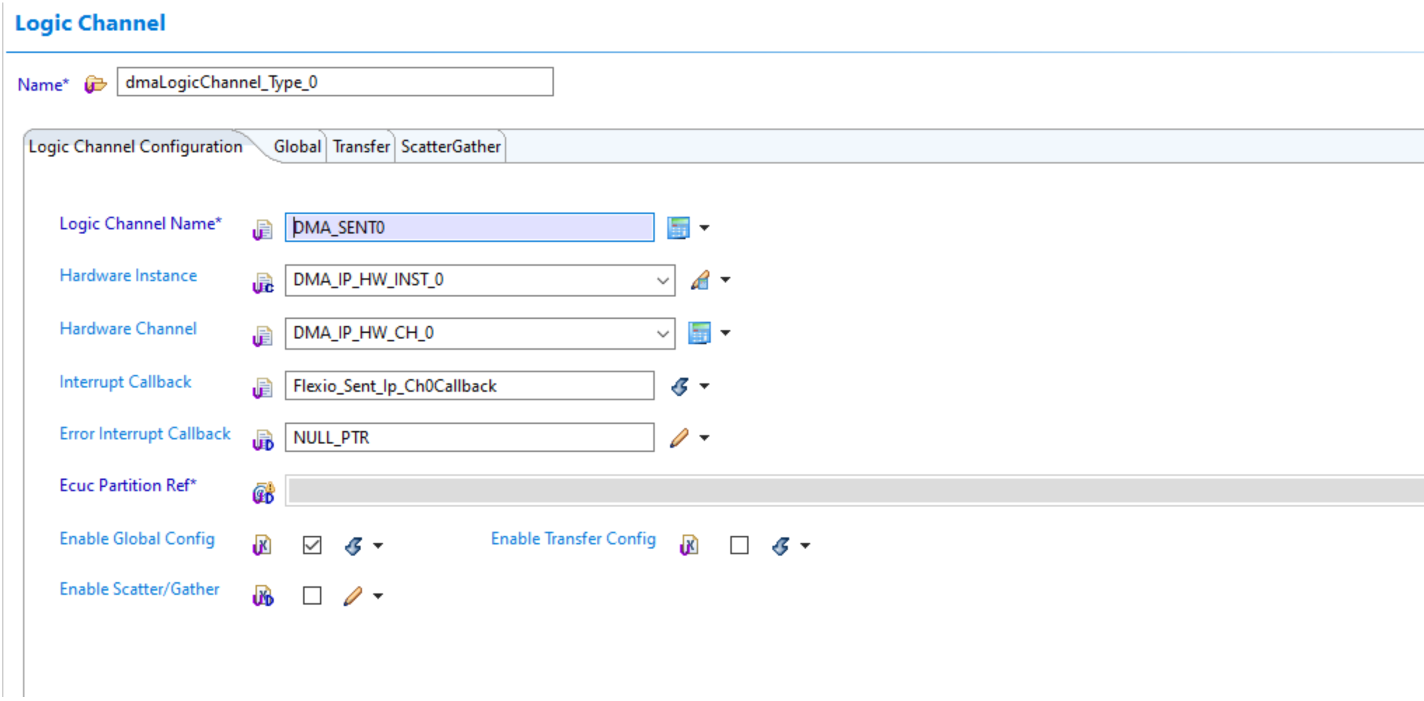
Name: dmaLogicInstance_DmaCrc

Enable Swap Bit □ Enable Swap Byte □

Enable Global □

3. In "Logic Channel Configuration" an entry should be added for each logical instance. Choose hardware instance, hardware channel, set the corresponding "Interrupt Callback" and enable the "Enable Global Config" check.

The "Interrupt Callback" should have one of the following values: Flexio_Sent_Ip_Ch0Callback, Flexio_↩ Sent_Ip_Ch1Callback, Flexio_Sent_Ip_Ch2Callback, Flexio_Sent_Ip_Ch3Callback, depending of the Sent instace that is used. For example, if the Dma channel is configured for Ch0Callback, the value of Interrupt Callback should be Flexio_Sent_Ip_Ch0Callback. This function is defined in Flexio_Sent_Ip.c file and it is called at the end of the dma transfer.



Logic Channel

Name* dmaLogicChannel_Type_0

Logic Channel Name* DMA_SENT0

Hardware Instance DMA_IP_HW_INST_0

Hardware Channel DMA_IP_HW_CH_0

Interrupt Callback Flexio_Sent_Ip_Ch0Callback

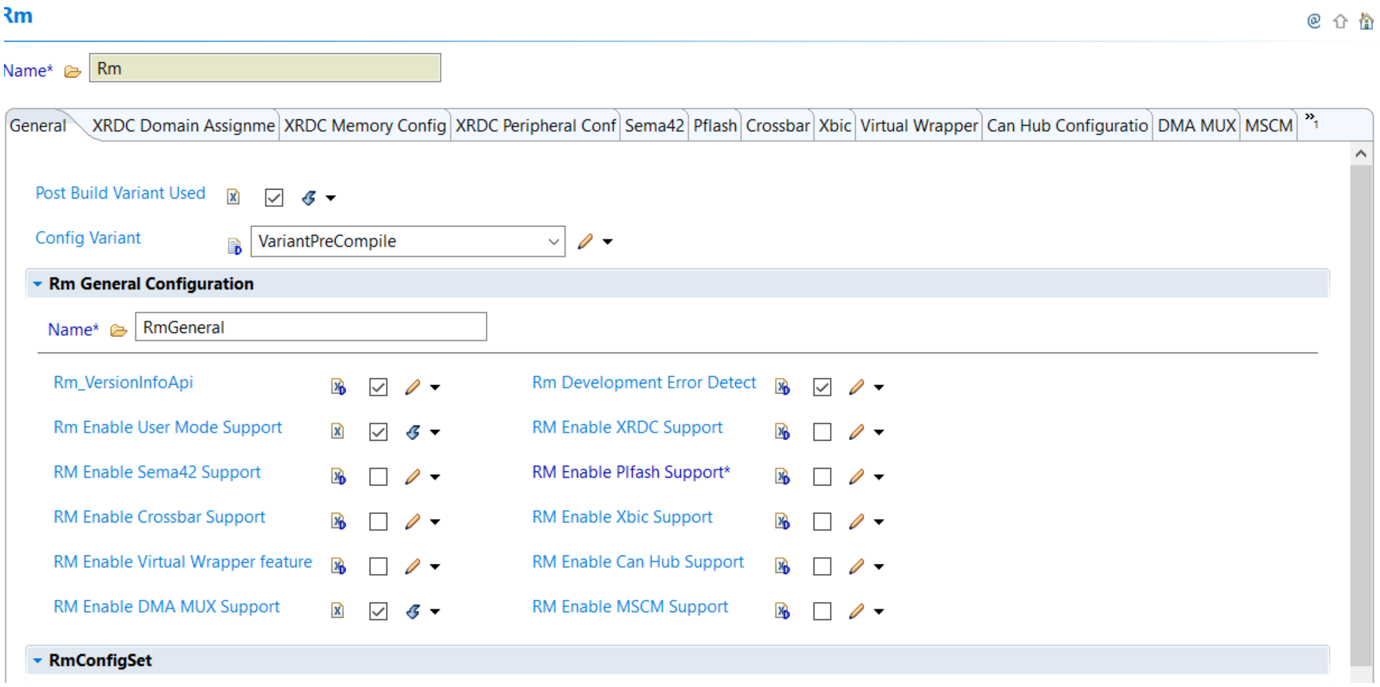
Error Interrupt Callback NULL_PTR

Ecuc Partition Ref*

Enable Global Config ☑ Enable Transfer Config ☐

Enable Scatter/Gather ☐

4. In RM module, enable DMA mux support. The "Enable DMAMUX Source" and "Enable DMA Request" checks should be enabled in the corresponding DMA channel.



Rm

Name* Rm

Post Build Variant Used ☑

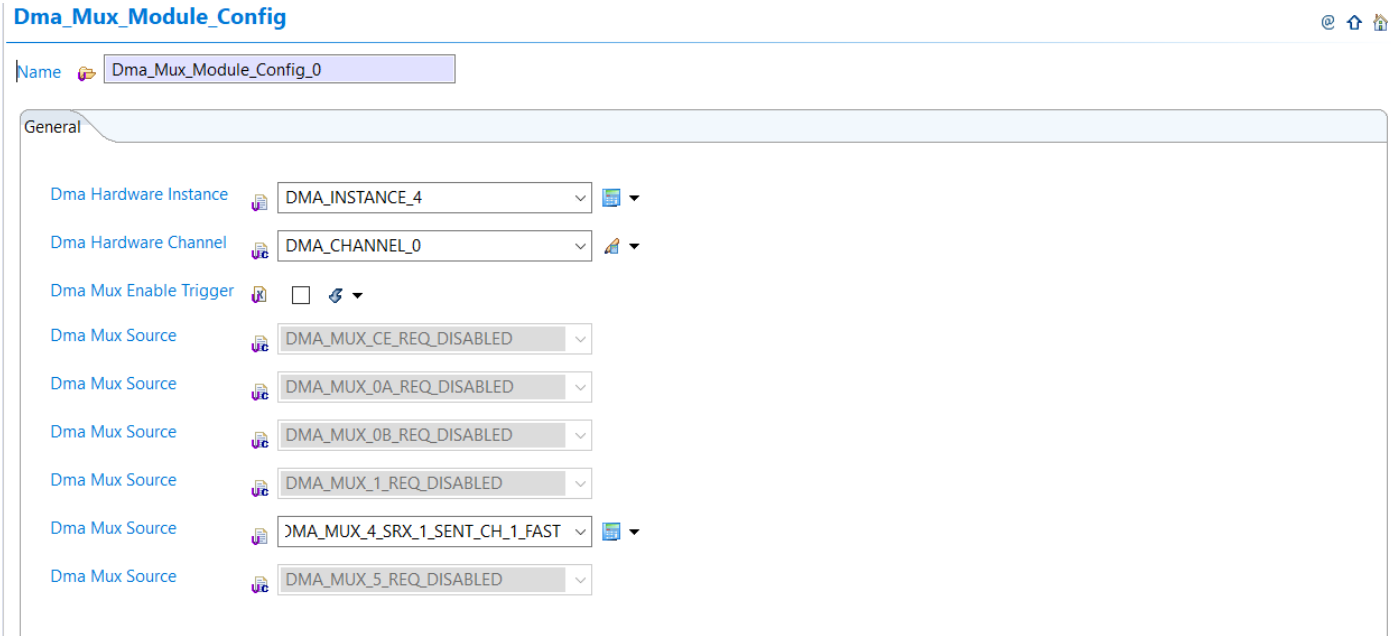
Config Variant VariantPreCompile

Rm General Configuration

Name* RmGeneral

Rm_VersionInfoApi	☑	Rm Development Error Detect	☑
Rm Enable User Mode Support	☑	RM Enable XRDC Support	☐
RM Enable Sema42 Support	☐	RM Enable Pflash Support*	☐
RM Enable Crossbar Support	☐	RM Enable Xbic Support	☐
RM Enable Virtual Wrapper feature	☐	RM Enable Can Hub Support	☐
RM Enable DMA MUX Support	☑	RM Enable MSCM Support	☐

RmConfigSet



Dma_Mux_Module_Config

Name: Dma_Mux_Module_Config_0

General

Dma Hardware Instance: DMA_INSTANCE_4

Dma Hardware Channel: DMA_CHANNEL_0

Dma Mux Enable Trigger: □

Dma Mux Source: DMA_MUX_CE_REQ_DISABLED

Dma Mux Source: DMA_MUX_0A_REQ_DISABLED

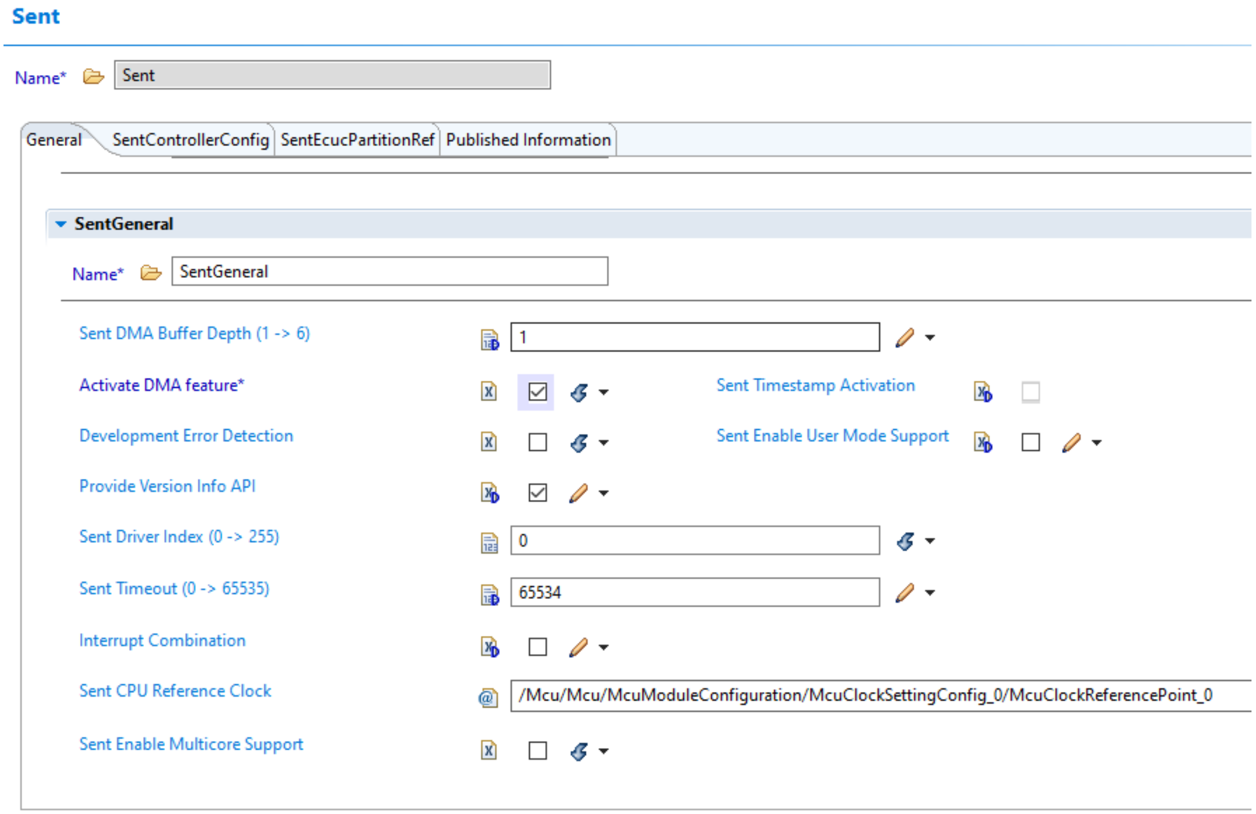
Dma Mux Source: DMA_MUX_0B_REQ_DISABLED

Dma Mux Source: DMA_MUX_1_REQ_DISABLED

Dma Mux Source:)MA_MUX_4_SRX_1_SENT_CH_1_FAST

Dma Mux Source: DMA_MUX_5_REQ_DISABLED

5. In Sent component, "General" tab, the "Activate DMA Feature" should be enabled.



Sent

Name*: Sent

SentGeneral

Name*: SentGeneral

Sent DMA Buffer Depth (1 -> 6): 1

Activate DMA feature*: ☑

Sent Timestamp Activation: □

Development Error Detection: □

Sent Enable User Mode Support: □

Provide Version Info API: ☑

Sent Driver Index (0 -> 255): 0

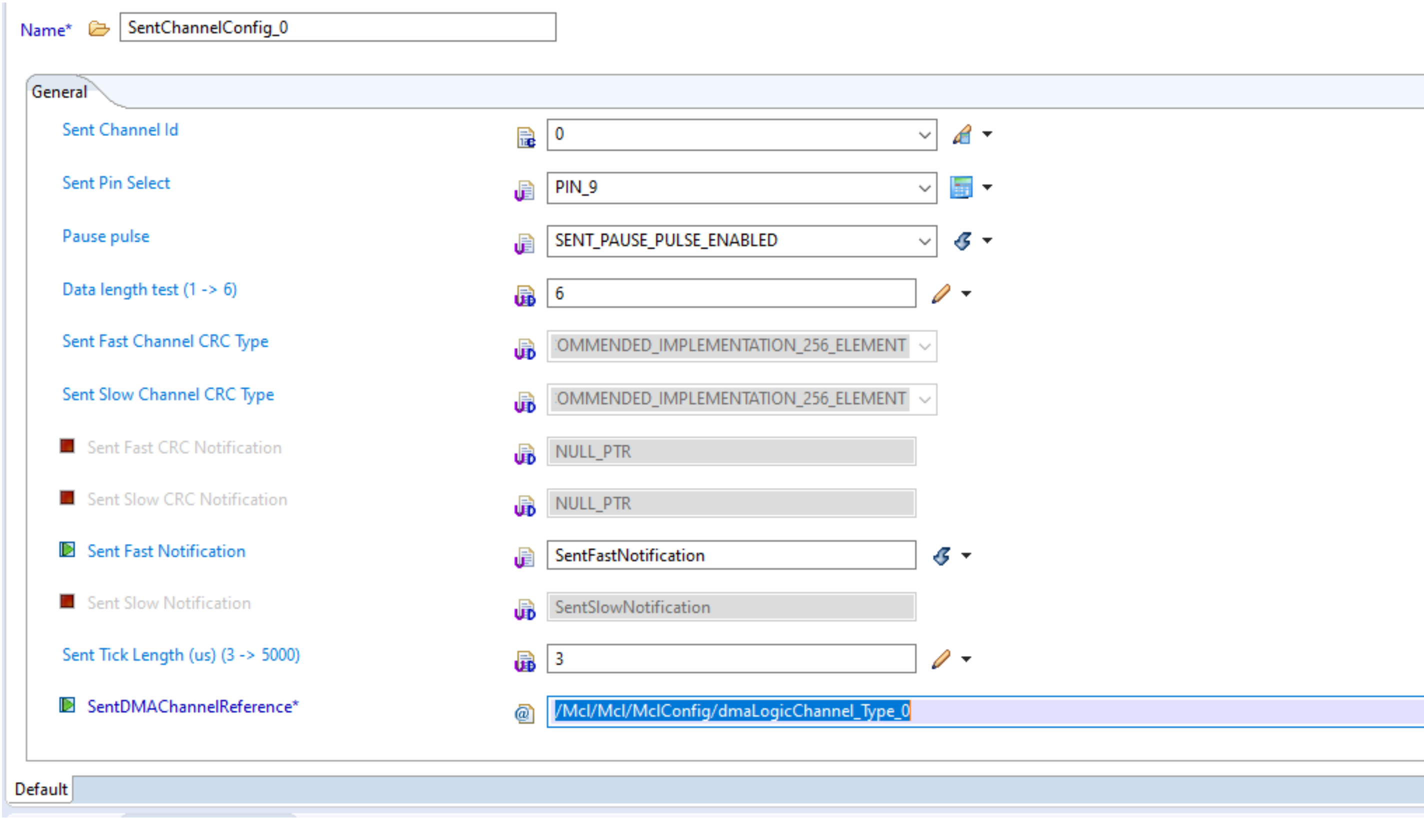
Sent Timeout (0 -> 65535): 65534

Interrupt Combination: □

Sent CPU Reference Clock: /Mcu/Mcu/McuModuleConfiguration/McuClockSettingConfig_0/McuClockReferencePoint_0

Sent Enable Multicore Support: □

6. For the corresponding Sent channel, select "SENT_USING_DMA" field and choose the corresponding Dma channels for "SentDMAChanelReference".



Name* SentChannelConfig_0

General

Sent Channel Id	0
Sent Pin Select	PIN_9
Pause pulse	SENT_PAUSE_PULSE_ENABLED
Data length test (1 -> 6)	6
Sent Fast Channel CRC Type	OMMENDED_IMPLEMENTATION_256_ELEMENT
Sent Slow Channel CRC Type	OMMENDED_IMPLEMENTATION_256_ELEMENT
🟥 Sent Fast CRC Notification	NULL_PTR
🟥 Sent Slow CRC Notification	NULL_PTR
☑ Sent Fast Notification	SentFastNotification
🟥 Sent Slow Notification	SentSlowNotification
Sent Tick Length (us) (3 -> 5000)	3
☑ SentDMAChannelReference*	/Mcl/Mcl/MclConfig/dmaLogicChannel_Type_0

3.7 Runtime errors

Driver doesn't support to generate DEM error when timeout occurs

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 Sent
  - Container SentConfigSet
    - Container SentControllerConfig
      - Parameter SentTimestampPrescaller
      - Parameter SentControllerActivation
      - Parameter EnableHwFiFo
      - Parameter SentProcessing
      - Parameter SentControllerId
      - Reference SentControllerEcucPartitionRef
      - Reference SentHwControllerRef
      - Container SentChannelConfig
      - Parameter SentChannelId
      - Parameter SentSyncAsyncSelection
      - Parameter DataLength
      - Parameter DmaBufferDepth
      - Parameter CrcStatusNibbleIncluding
      - Parameter ChannelCrcImplementationArrayType
      - Parameter SentFastChannelCRCType
      - Parameter SentSlowChannelCRCType
      - Parameter SentFastCRCErrorNotificationEnable
      - Parameter SentFastCRCErrorNotification
      - Parameter SentSlowCRCErrorNotificationEnable
      - Parameter SentSlowCRCErrorNotification
      - Parameter SentFastNotificationEnable
      - Parameter SentFastNotification
      - Parameter SentSlowNotificationEnable
      - Parameter SentSlowNotification
      - Parameter SentTickLength
      - Parameter SentTickLengthExpand
      - Reference SentFlexioChannelRef

        - Reference SentDmaChannelRef
  - Container SentGeneral
    - Parameter SentDmaActivation
    - Parameter TickLengthExpandRange
    - Parameter Support256ArrayImplementation
    - Parameter SentTimestampActivation
    - Parameter SentDevErrorDetect
    - Parameter SentEnableUserModeSupport
    - Parameter SentVersionInfoApi
    - Parameter SentIndex
    - Parameter SentTimeout
    - Parameter SentTimeoutMethod
    - Parameter SentEnableMulticoreSupport
    - Reference SentEcucPartitionRef
    - Reference SentCpuClockRef
  - Container CommonPublishedInformation
    - Parameter ArReleaseMajorVersion
    - Parameter ArReleaseMinorVersion
    - Parameter ArReleaseRevisionVersion
    - Parameter ModuleId
    - Parameter SwMajorVersion
    - Parameter SwMinorVersion
    - Parameter SwPatchVersion
    - Parameter VendorApiInfix
    - Parameter VendorId

4.1 Module Sent

This container holds the configuration of a single Sent Driver.

Included containers:

- SentConfigSet
- SentGeneral
- CommonPublishedInformation

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

4.2 Container SentConfigSet

This is the multiple configuration set container for SENT Driver.

Included subcontainers:

- SentControllerConfig

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

4.3 Container SentControllerConfig

Contains the channels of each Sent controller

Included subcontainers:

- SentChannelConfig

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

4.4 Parameter SentTimestampPrescaller

0 to 255 - Time Stamp Clock = (High Frequency Clock)/(SentTimestampPrescaller + 1). Value written in this field should be the value of SentTimestampPrescaller from above formula. Value set by user software should be such that the clock period is at least 1 ?s.

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-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: PRE-COMPILE
defaultValue	0
max	255
min	0

4.5 Parameter SentControllerActivation

Defines if a SENT controller is used in the configuration.

Deactivation of a particular SENT controller is equivalent to a SENT controller not used in the configuration.

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-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: PRE-COMPILE
defaultValue	true

4.6 Parameter EnableHwFiFo

Enable or disable hardware FIFO for this controller.

If DMA is not eanbled, this node is unused

Property	Value
type	ECUC-BOOLEAN-PARAM-DEF

Property	Value
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	false

4.7 Parameter SentProcessing

Select Interrupt or polling used for this channel.

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	POLLING
literals	['INTERRUPT', 'POLLING', 'DMA']

4.8 Parameter SentControllerId

This parameter provides the controller ID which is unique in a given SENT Driver.

The value for this parameter starts with 0 and continue without any gaps.

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

Property	Value
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	false
valueConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: PRE-COMPILE
defaultValue	1
max	255
min	0

4.9 Reference SentControllerEcucPartitionRef

EN: Maps a SENT controller to zero or multiple ECUC partitions to limit the access to this controller. The ECUC partitions referenced are a subset of the ECUC partitions where the SENT driver is mapped to.

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

4.10 Reference SentHwControllerRef

Sent Flexio Controller Reference

Reference to the Flexio Controller configure for the Request

Property	Value
type	ECUC-CHOICE-REFERENCE-DEF
origin	NXP
lowerMultiplicity	1

Property	Value
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	false
valueConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: POST-BUILD
requiresSymbolicNameValue	False
destinations	['/TS_T40D34M30I0R0/Mcl/MclConfig/FlexioCommon']

4.11 Container SentChannelConfig

This container contains the configuration (parameters) of the SENT channels.

Included subcontainers:

- None

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

4.12 Parameter SentChannelId

The Id of this field will be part of the notification message.

Property	Value
type	ECUC-INTEGER-PARAM-DEF
origin	AUTOSAR_ECUC
symbolicNameValue	true
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	false

Property	Value
valueConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: POST-BUILD
defaultValue	0
max	255
min	0

4.13 Parameter SentSyncAsyncSelection

Note

EN: pause pulse enabled has the effect of a in sync receive. Pause pulse disable means Async receive

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

4.14 Parameter DataLength

1 to 6 Data Nibbles supported in this Channel

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

Property	Value
	VARIANT-POST-BUILD: POST-BUILD
defaultValue	6
max	6
min	1

4.15 Parameter DmaBufferDepth

number of frames were received before processing data

Property	Value
type	ECUC-INTEGER-PARAM-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	true
valueConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: POST-BUILD
defaultValue	1
max	5
min	1

4.16 Parameter CrcStatusNibbleIncluding

Enable Including Status Nibble in Crc calculation (Fast Msg only).

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-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: POST-BUILD
defaultValue	false

4.17 Parameter ChannelCrcImplementationArrayType

Sent Channel CRC Implementation Array Type

IMPLEMENTATION_256_ELEMENTS_ARRAY: Implementation using 256 Elements Array

IMPLEMENTATION_16_ELEMENTS_ARRAY: Implementation using 16 Elements Array

Property	Value
type	ECUC-ENUMERATION-PARAM-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	true
valueConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: POST-BUILD
defaultValue	IMPLEMENTATION_16_ELEMENTS_ARRAY
literals	['IMPLEMENTATION_16_ELEMENTS_ARRAY', 'IMPLEMENTATION_↩ 256_ELEMENTS_ARRAY']

4.18 Parameter SentFastChannelCRCType

Sent Fast Channel CRC Type

SENT_RECOMMENDED_IMPLEMENTATION: Recommend Implementation.

SENT_LEGACY_IMPLEMENTATION: Legacy Implementation.

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

4.19 Parameter SentSlowChannelCRCType

Sent Slow Channel CRC Type (Serial short type only)

SENT_RECOMMENDED_IMPLEMENTATION: Recommend Implementation.

SENT_LEGACY_IMPLEMENTATION: Legacy Implementation.

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

4.20 Parameter SentFastCRCErrorNotificationEnable

Enable Fast Error Notification.

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-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: POST-BUILD
defaultValue	false

4.21 Parameter SentFastCRCErrorNotification

This parameter defines the existence and the name of a callout function that is called after an error occured. No call should occur if this is not enabled.

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

4.22 Parameter SentSlowCRCErrorNotificationEnable

Enable Slow Error Notification.

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-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: POST-BUILD
defaultValue	false

4.23 Parameter SentSlowCRCErrorNotification

This parameter defines the existence and the name of a callout function that is called after an error occured. No call should occur if this is not enabled.

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

4.24 Parameter SentFastNotificationEnable

Sent Fast Notification Enable.

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-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: POST-BUILD
defaultValue	false

4.25 Parameter SentFastNotification

User callback function

Property	Value
type	ECUC-FUNCTION-NAME-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1

Property	Value
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	true
valueConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: POST-BUILD
defaultValue	SentFastNotification

4.26 Parameter SentSlowNotificationEnable

Sent Slow Notification Enable.

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-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: POST-BUILD
defaultValue	false

4.27 Parameter SentSlowNotification

User callback function

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

4.28 Parameter SentTickLength

Specifies the tick length us.

This parameter is used in order to decode the slave signal

NoteImplementation specific Parameter.

Property	Value
type	ECUC-INTEGER-PARAM-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	true
valueConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: POST-BUILD
defaultValue	3
max	5000
min	3

4.29 Parameter SentTickLengthExpand

Specifies the tick length ns.

This parameter is used in order to decode the slave signal

NoteImplementation specific Parameter.

Property	Value
type	ECUC-INTEGER-PARAM-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	true
valueConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: POST-BUILD
defaultValue	3000
max	5000000
min	1000

4.30 Reference SentFlexioChannelRef

Sent Flexio Channel Reference

Reference to the Flexio Channel configure for the Request

Property	Value
type	ECUC-CHOICE-REFERENCE-DEF
origin	NXP
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	false
valueConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: POST-BUILD
requiresSymbolicNameValue	False
destinations	['/TS_T40D34M30I0R0/Mcl/MclConfig/FlexioCommon/FlexioMclLogicChannels']

4.31 Reference SentDmaChannelRef

Sent DMA Channel Reference

Reference to the DMA Channel configure for the Request

Property	Value
type	ECUC-CHOICE-REFERENCE-DEF
origin	NXP
lowerMultiplicity	0
upperMultiplicity	1
postBuildVariantMultiplicity	false
multiplicityConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: POST-BUILD
postBuildVariantValue	false
valueConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: POST-BUILD
requiresSymbolicNameValue	False
destinations	['/TS_T40D34M30I0R0/Mcl/MclConfig/dmaLogicChannel_Type']

4.32 Container SentGeneral

This container holds the parameters related each SENT Driver Unit.

Included subcontainers:

- None

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

4.33 Parameter SentDmaActivation

Activates DMA feature

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-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: PRE-COMPILE
defaultValue	false

4.34 Parameter TickLengthExpandRange

Defines if a SENT controller is used in expand mode ().

Property	Value
type	ECUC-BOOLEAN-PARAM-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A

Property	Value
multiplicityConfigClasses	N/A
postBuildVariantValue	false
valueConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: PRE-COMPILE
defaultValue	false

4.35 Parameter Support256ArrayImplementation

Defines if supporting Crc calculation using 256 Elements Array.

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-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: PRE-COMPILE
defaultValue	false

4.36 Parameter SentTimestampActivation

Defines if Timestamp feature is active on current controller.

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-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: PRE-COMPILE
defaultValue	false

4.37 Parameter SentDevErrorDetect

Switches the Development Error Detection and Notification: ON or OFF.

When this option is OFF code size is reduced, but no error detection is available.

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-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: PRE-COMPILE
defaultValue	false

4.38 Parameter SentEnableUserModeSupport

When this parameter is enabled, the SENT module will adapt to run from User Mode, with the following measures:

(if applicable) a) configuring REG_PROT for the SENT Controllers so that the registers under protection can be accessed from user mode by setting UAA bit in REG_PROT_GCR to 1

(if applicable) b) using 'call trusted function' stubs for all internal function calls that access registers requiring supervisor mode.

(if applicable) c) other module specific measures for more information, please see chapter 5.7 User Mode Support in IM

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

4.39 Parameter SentVersionInfoApi

Switches the Sent_GetVersionInfo() API: ON or OFF.

When this option is ON driver supports API for getting Version information for the Driver.

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-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: PRE-COMPILE
defaultValue	true

4.40 Parameter SentIndex

Specifies the InstanceId of this module instance.

If only one instance is present it shall have the Id 0.

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-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: PRE-COMPILE
defaultValue	0
max	255
min	0

4.41 Parameter SentTimeout

Specifies the timeout for Sent module.

If the timeout is exceeded, the while loop will be exited

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-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: PRE-COMPILE
defaultValue	100000
max	4294967295
min	0

4.42 Parameter SentTimeoutMethod

Configures the timeout method.

Based on this selection a certain timeout method from OsIf will be used in the driver.

Note: If SystemTimer or CustomTimer are selected make sure the corresponding timer is enabled in OsIf General configuration.

Note: 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	OSIF_COUNTER_DUMMY
literals	['OSIF_COUNTER_DUMMY', 'OSIF_COUNTER_SYSTEM', 'OSIF_COU↩NTER_CUSTOM']

4.43 Parameter SentEnableMulticoreSupport

Vendor specific:

Enable Multicore support for Sent driver

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

4.44 Reference SentEcucPartitionRef

Maps the SENT driver to zero or multiple ECUC partitions to make the driver API available in the according partition. Depending on the addressed timer resource the interfaces operate as follows:

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

4.45 Reference SentCpuClockRef

Reference to the CPU clock configuration, which is set in the MCU driver configuration.

MCU plugin need to be added and then give the reference to it.

Property	Value
type	ECUC-REFERENCE-DEF
origin	NXP
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	false
valueConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: PRE-COMPILE
requiresSymbolicNameValue	False
destination	/AUTOSAR/EcucDefs/Mcu/McuModuleConfiguration/McuClockSetting↩ Config/McuClockReferencePoint

4.46 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

4.47 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

Property	Value
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

4.48 Parameter ArReleaseMinorVersion

Minor 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	7
max	7
min	7

4.49 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

Property	Value
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

4.50 Parameter ModuleId

Module ID of this module from Module List.

Note: Implementation Specific Parameter

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	125
max	125
min	125

4.51 Parameter SwMajorVersion

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

Note: Implementation Specific Parameter

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

Property	Value
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

4.52 Parameter SwMinorVersion

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

Note: Implementation Specific Parameter

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

4.53 Parameter SwPatchVersion

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

Note: Implementation Specific Parameter

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

4.54 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>.

This parameter is mandatory for all modules with upper multiplicity >

1. It shall not be used for modules with upper multiplicity =1.

Note: Implementation Specific Parameter

Property	Value
type	ECUC-STRING-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	

4.55 Parameter VendorId

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

Note: Implementation Specific Parameter

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

This chapter describes the Tresos configuration plug-in for the Sent Driver. The most of the parameters are described below.

Chapter 5

Module Index

5.1 Software Specification

Here is a list of all modules:

Chapter 6

Module Documentation

6.1 SENT_DRIVER

6.1.1 Detailed Description

Files

- file Sent_Types.h

  AUTOSAR Sent - Sent generic types file.

Data Structures

- struct Flexio_Sent_Ip_TimerConfig
- struct Flexio_Sent_Ip_TimerControl
- struct Sent_Ipw_CtrlConfigType

  The structure contains the hardware channel configuration type. More...
- struct Sent_ControllerConfigType

  Structure that contains Sent Hw configuration. More...
- struct Sent_ConfigType

  This type contains initialization data. More...

Macros

- #define SENT_E_INVALID_CTRL

  API service used with an invalid or inactive channel parameter.
- #define SENT_SID_DEINIT

  Service ID (APIs) for Det reporting.
- #define SENT_SID_INIT

  Service ID (APIs) for Det reporting.
- #define SENT_SID_GETFASTCHANNELMSGDATA

Service ID (APIs) for Det reporting.

- #define SENT_SID_GETSERIALCHANNELMSGDATA

  Service ID (APIs) for Det reporting.
- #define SENT_SID_GETFASTMSGDATA

  Service ID (APIs) for Det reporting.
- #define SENT_MAX_NIBBLE_DATA

  Maximum number of nibbles configurable by user according user manual.
- #define SENT_TIMER(x)
- #define SENT_SHORT_MSGID_MASK
- #define SENT_ENHANCED8_MSGID74_MASK
- #define SENT_ENHANCED4_MSGID_MASK
- #define SENT_ENHANCED_BIT_CONF_MASK
- #define SENT_GET_BIT2(x, y)
- #define SENT_GET_BIT3(x, y)
- #define SENT_BITXOR(x, y)
- #define SET_GET_BIT24(x, y)

Enum Reference

- enum Sent_DriverStatusType

  Sent_DriverStatusType.

Function Reference

- void Sent_Init (const Sent_ConfigType ∗Config)

  Initialize the Sent driver.
- void Sent_DeInit (void)

  DeInitializes the Sent module.
- Std_ReturnType Sent_GetFastChannelMsgData (uint8 ControllerId, uint8 ChannelId)

  This function processing polling of Fast notification.
- Std_ReturnType Sent_GetFastMsgData (uint8 ControllerId)

  This function processing polling of Fast notification.
- Std_ReturnType Sent_GetSerialChannelMsgData (uint8 ControllerId, uint8 ChannelId)

  This function processing polling of Serial notification.
- Std_ReturnType Sent_GetSerialMsgData (uint8 ControllerId)

  This function processing polling of Serial notification.
- Std_ReturnType Sent_StartChannelReceiving (uint8 ControllerId, uint8 ChannelId)

  This function starts to receive data from SENT bus.
- Std_ReturnType Sent_StopChannelReceiving (uint8 ControllerId, uint8 ChannelId)

  This function stops to receive data from SENT bus.
- Flexio_Sent_Ip_StatusType Flexio_Sent_Ip_Init (uint8 Instance, const Flexio_CtrlConfigType ∗Config)

  Initialize the FLEXIO_Sent driver.
- void Flexio_Sent_Ip_DeInit (uint8 Instance)

  De-Initialize the FLEXIO_Sent driver.
- Flexio_Sent_Ip_StatusType Flexio_Sent_Ip_GetFastChannelMsgData (const uint8 Instance, const uint8 ChannelId)

Get Fast message data for a channel.

- Flexio_Sent_Ip_StatusType Flexio_Sent_Ip_GetFastMsgData (const uint8 Instance)

  Get Fast message data for all channel.
- Flexio_Sent_Ip_StatusType Flexio_Sent_Ip_GetSerialChannelMsgData (const uint8 Instance, const uint8 ChannelId)

  Get Serial message data for a channel.
- Flexio_Sent_Ip_StatusType Flexio_Sent_Ip_GetSerialMsgData (const uint8 Instance)

  Get Serial message data for all channel.
- Flexio_Sent_Ip_StatusType Flexio_Sent_Ip_StartChannelReceiving (const uint8 Instance, const uint8 ChannelId)

  Start receiving SENT data from bus on specific channel.
- Flexio_Sent_Ip_StatusType Flexio_Sent_Ip_StopChannelReceiving (const uint8 Instance, const uint8 ChannelId)

  Stop receiving SENT data from bus on specific channel.
- void Sent_Ipw_ControllerInit (uint8 ControllerId, const Sent_Ipw_CtrlConfigType ∗IpwController)

  Initialize the Sent driver.
- void Sent_Ipw_ControllerDeInit (uint8 ControllerId)

  DeInitializes the Sent module.
- Std_ReturnType Sent_Ipw_GetFastChannelMsgData (uint8 ControllerId, uint8 ChannelId)

  This function processing polling of Fast notification.
- Std_ReturnType Sent_Ipw_GetFastMsgData (uint8 ControllerId)

  This function gets Fast message data for all channel.
- Std_ReturnType Sent_Ipw_GetSerialChannelMsgData (uint8 ControllerId, uint8 ChannelId)

  This function processing polling of Serial notification.
- Std_ReturnType Sent_Ipw_GetSerialMsgData (uint8 ControllerId)

  This function gets Serial message data for all channel.
- Std_ReturnType Sent_Ipw_StartChannelReceiving (uint8 ControllerId, uint8 ChannelId)

  This function starts to receive data from SENT bus.
- Std_ReturnType Sent_Ipw_StopChannelReceiving (uint8 ControllerId, uint8 ChannelId)

  This function stops to receive data from SENT bus.

6.1.2 Data Structure Documentation

6.1.2.1 struct Flexio_Sent_Ip_TimerConfig

FlexIO timer config register This is a structure used by all FlexIO drivers as timer config value. It is needed for parameter of set timer config register value.

Definition at line 113 of file Flexio_Sent_Ip_HwAccess.h.

6.1.2.2 struct Flexio_Sent_Ip_TimerControl

FlexIO timer control register This is a structure used by all FlexIO drivers as timer control value. It is needed for parameter of set timer control register value.

Definition at line 129 of file Flexio_Sent_Ip_HwAccess.h.

6.1.2.3 struct Sent_Ipw_CtrlConfigType

The structure contains the hardware channel configuration type.

Definition at line 114 of file Sent_Ipw_Types.h.

6.1.2.4 struct Sent_ControllerConfigType

Structure that contains Sent Hw configuration.

It contains the information specific to one Sent Hw unit

Definition at line 136 of file Sent_Types.h.

Data Fields

Type	Name	Description
const uint8	CtrHwOffset	HWoffet of configured controller. < HWoffet of configured controller
const uint8	CtrHwID	Summary of all the channel in a controller.
const uint8	ChnlConfigured	Structure containing the hardware specific configuration for the channel.
const Sent_Ipw_CtrlConfigType ∗	CtrlConfig	

6.1.2.5 struct Sent_ConfigType

This type contains initialization data.

This contains initialization data for the Sent driver. It shall contain:

- The number of Sent modules to be configured
- Dem error reporting configuration
- Sent dependent properties for used HW units

Definition at line 160 of file Sent_Types.h.

Data Fields

Type	Name	Description
const uint32	CoreIDConfigured	CodeID for configured controller. < Summary of all the configured controller.
const uint8	CtrConfigured	Pointer to Sent hardware unit configuration.
const Sent_ControllerConfigType ∗const ∗const	ControllerConfig	

6.1.3 Macro Definition Documentation

6.1.3.1 SENT_E_INVALID_CTRL

```
#define SENT_E_INVALID_CTRL
```

API service used with an invalid or inactive channel parameter.

The Sent Driver module shall report the development error "SENT_E_INVALID_CTRL (0x07)", when API Service used with an invalid or inactive channel parameter.

Definition at line 174 of file CDD_Sent.h.

6.1.3.2 SENT_SID_DEINIT

```
#define SENT_SID_DEINIT
```

Service ID (APIs) for Det reporting.

Service ID (APIs) for Det reporting

Definition at line 180 of file CDD_Sent.h.

6.1.3.3 SENT_SID_INIT

```
#define SENT_SID_INIT
```

Service ID (APIs) for Det reporting.

Service ID (APIs) for Det reporting

Definition at line 186 of file CDD_Sent.h.

6.1.3.4 SENT_SID_GETFASTCHANNELMSGDATA

```
#define SENT_SID_GETFASTCHANNELMSGDATA
```

Service ID (APIs) for Det reporting.

Service ID (APIs) for Det reporting

Definition at line 192 of file CDD_Sent.h.

6.1.3.5 SENT_SID_GETSERIALCHANNELMSGDATA

```
#define SENT_SID_GETSERIALCHANNELMSGDATA
```

Service ID (APIs) for Det reporting.

Service ID (APIs) for Det reporting

Definition at line 198 of file CDD_Sent.h.

6.1.3.6 SENT_SID_GETFASTMSGDATA

```
#define SENT_SID_GETFASTMSGDATA
```

Service ID (APIs) for Det reporting.

Service ID (APIs) for Det reporting

Definition at line 204 of file CDD_Sent.h.

6.1.3.7 SENT_MAX_NIBBLE_DATA

```
#define SENT_MAX_NIBBLE_DATA
```

Maximum number of nibbles configurable by user according user manual.

Definition at line 213 of file CDD_Sent.h.

6.1.3.8 SENT_TIMER

```
#define SENT_TIMER(
              x )
```

Timer of FLEXIO IP used for the simulation of the SENT protocol

Definition at line 135 of file Flexio_Sent_Ip.h.

6.1.3.9 SENT_SHORT_MSGID_MASK

```
#define SENT_SHORT_MSGID_MASK
```

SENT SHORT DATA

Definition at line 140 of file Flexio_Sent_Ip.h.

6.1.3.10 SENT_ENHANCED8_MSGID74_MASK

```
#define SENT_ENHANCED8_MSGID74_MASK
```

SENT ENHANCED 12 BIT DATA + 8 BIT ID

Definition at line 155 of file Flexio_Sent_Ip.h.

6.1.3.11 SENT_ENHANCED4_MSGID_MASK

```
#define SENT_ENHANCED4_MSGID_MASK
```

SENT ENHANCED 16 BIT DATA + 4 BIT ID

Definition at line 170 of file Flexio_Sent_Ip.h.

6.1.3.12 SENT_ENHANCED_BIT_CONF_MASK

```
#define SENT_ENHANCED_BIT_CONF_MASK
```

Get configuration bit for enhanced data

Definition at line 192 of file Flexio_Sent_Ip.h.

6.1.3.13 SENT_GET_BIT2

```
#define SENT_GET_BIT2(
              x,
              y )
```

Get all #BIT2 from status nibble

Definition at line 199 of file Flexio_Sent_Ip.h.

6.1.3.14 SENT_GET_BIT3

```
#define SENT_GET_BIT3(
             x,
             y )
```

Get all #BIT3 from status nibble

Definition at line 203 of file Flexio_Sent_Ip.h.

6.1.3.15 SENT_BITXOR

```
#define SENT_BITXOR(
             x,
             y )
```

bit XOR for CRC calculation

Definition at line 207 of file Flexio_Sent_Ip.h.

6.1.3.16 SET_GET_BIT24

```
#define SET_GET_BIT24(
             x,
             y )
```

Get bit from enhanced serial message to create 24 bit sequence for CRC checking x: BIT serial data y: bit order

Definition at line 213 of file Flexio_Sent_Ip.h.

6.1.4 Enum Reference

6.1.4.1 Sent_DriverStatusType

```
enum Sent_DriverStatusType
```

Sent_DriverStatusType.

Sent Driver status used for checking and preventing double driver intialization. SENT_UNINIT = The state S↩ENT_UNINIT means that the Sent module has not been initialized yet and cannot be used. SENT_INIT = The SENT_INIT state indicates that the Sent driver has been initialized, making each available channel ready for service.

Enumerator

SENT_UNINIT	Driver not initialized.
SENT_INIT	Driver ready.

Definition at line 122 of file Sent_Types.h.

6.1.5 Function Reference

6.1.5.1 Sent_Init()

```
void Sent_Init (
            const Sent_ConfigType * Config )
```

Initialize the Sent driver.

This function performs software initialization of Sent driver:

- Maps logical channels to hardware channels
- Initializes all channels
- Sets driver state machine to SENT_INIT.

Parameters

Config	Pointer to the Sent configuration structure. The function reads configuration data from this structure and initializes the driver accordingly. The application may free this structure after the function returns

Returns

Error or success Status returned by API

Note

Service ID: 0x02.

Synchronous, non re-entrant function.

6.1.5.2 Sent_DeInit()

```
void Sent_DeInit (
            void )
```

DeInitializes the Sent module.

This function performs software de initialization of Sent modules to reset values. The service influences only the peripherals, which are allocated by static configuration and the runtime configuration set passed by the previous call of Sent_Init() The driver needs to be initialized before calling Sent_DeInit(). Otherwise, the function Sent_DeInit shall raise the development error SENT_UNINIT and leave the desired de initialization functionality without any action.

Parameters

in	*void*	

Returns

void

Note

Service ID: 0x01.

Synchronous, non re-entrant function.

6.1.5.3 Sent_GetFastChannelMsgData()

```
Std_ReturnType Sent_GetFastChannelMsgData (
            uint8 ControllerId,
            uint8 ChannelId )
```

This function processing polling of Fast notification.

This function gets Fast message data for a channel.

Parameters

in	*Controller↵ Id*	The Id of controller
in	*ChannelId*	The Id of channel

Returns

E_OK: Channel request has been accepted. E_NOT_OK: Channel request has not been accepted.

Note

Service ID: 0x03.

Synchronous, non reentrant function.

6.1.5.4 Sent_GetFastMsgData()

```
Std_ReturnType Sent_GetFastMsgData (
            uint8 ControllerId )
```

This function processing polling of Fast notification.

This function gets Fast message data for all channel.

Parameters

in	*Controller↩ Id*	The Id of controller

Returns

E_OK: Channel request has been accepted. E_NOT_OK: Channel request has not been accepted.

Note

Service ID: 0x05.

Synchronous, non reentrant function.

6.1.5.5 Sent_GetSerialChannelMsgData()

```
Std_ReturnType Sent_GetSerialChannelMsgData (
            uint8 ControllerId,
            uint8 ChannelId )
```

This function processing polling of Serial notification.

This function gets Serial message data for a channel.

Parameters

in	*Controller↩ Id*	The Id of controller
in	*ChannelId*	The Id of channel

Returns

E_OK: Channel request has been accepted. E_NOT_OK: Channel request has not been accepted.

Note

Service ID: 0x04.

Synchronous, non reentrant function.

6.1.5.6 Sent_GetSerialMsgData()

```
Std_ReturnType Sent_GetSerialMsgData (
            uint8 ControllerId )
```

This function processing polling of Serial notification.

This function gets Serial message data for all channel.

Parameters

in	*ControllerId*	The Id of controller

Returns

E_OK: Channel request has been accepted. E_NOT_OK: Channel request has not been accepted.

Note

Service ID: 0x06.

Synchronous, non reentrant function.

6.1.5.7 Sent_StartChannelReceiving()

```
Std_ReturnType Sent_StartChannelReceiving (
            uint8 ControllerId,
            uint8 ChannelId )
```

This function starts to receive data from SENT bus.

This function starts to receive data from SENT bus.

Parameters

in	*ControllerId*	The Id of controller
in	*ChannelId*	The Id of channel

Returns

E_OK: Channel request has been accepted. E_NOT_OK: Channel request has not been accepted.

Note

Service ID: 0x07.

Synchronous, non reentrant function.

6.1.5.8 Sent_StopChannelReceiving()

```
Std_ReturnType Sent_StopChannelReceiving (
            uint8 ControllerId,
            uint8 ChannelId )
```

This function stops to receive data from SENT bus.

This function stops to receive data from SENT bus.

Parameters

in	*ControllerId*	The Id of controller
in	*ChannelId*	The Id of channel

Returns

E_OK: Channel request has been accepted. E_NOT_OK: Channel request has not been accepted.

Note

Service ID: 0x08.

Synchronous, non reentrant function.

6.1.5.9 Flexio_Sent_Ip_Init()

```
Flexio_Sent_Ip_StatusType Flexio_Sent_Ip_Init (
            uint8 Instance,
            const Flexio_CtrlConfigType * Config )
```

Initialize the FLEXIO_Sent driver.

This function initializes the FLEXIO_Sent driver.

Parameters

instance	FLEXIO peripheral instance number
config	Pointer to the FLEXIO_Sent controller configuration structure. The function reads configuration data from this structure and initializes the driver accordingly. The application may free this structure after the function returns.

Returns

Error or success status returned by API

6.1.5.10 Flexio_Sent_Ip_DeInit()

```
void Flexio_Sent_Ip_DeInit (
            uint8 Instance )
```

De-Initialize the FLEXIO_Sent driver.

This function de-initializes the FLEXIO_Sent driver.

Parameters

instance	FLEXIO peripheral instance number

Returns

Error or success status returned by API

6.1.5.11 Flexio_Sent_Ip_GetFastChannelMsgData()

```
Flexio_Sent_Ip_StatusType Flexio_Sent_Ip_GetFastChannelMsgData (
            const uint8 Instance,
            const uint8 ChannelId )
```

Get Fast message data for a channel.

This function gets Fast message data for a channel.

Parameters

instance	FLEXIO peripheral instance number
channelID	ID of FLEXIO TIMER channel

Returns

Error or success status returned by API

6.1.5.12 Flexio_Sent_Ip_GetFastMsgData()

```
Flexio_Sent_Ip_StatusType Flexio_Sent_Ip_GetFastMsgData (
             const uint8 Instance )
```

Get Fast message data for all channel.

This function gets Fast message data for all channel.

Parameters

instance	FLEXIO peripheral instance number

Returns

Error or success status returned by API

6.1.5.13 Flexio_Sent_Ip_GetSerialChannelMsgData()

```
Flexio_Sent_Ip_StatusType Flexio_Sent_Ip_GetSerialChannelMsgData (
             const uint8 Instance,
             const uint8 ChannelId )
```

Get Serial message data for a channel.

This function gets Serial message data for a channel.

Parameters

instance	FLEXIO peripheral instance number
channelID	ID of FLEXIO TIMER channel

Returns

Error or success status returned by API

6.1.5.14 Flexio_Sent_Ip_GetSerialMsgData()

```
Flexio_Sent_Ip_StatusType Flexio_Sent_Ip_GetSerialMsgData (
             const uint8 Instance )
```

Get Serial message data for all channel.

This function gets Serial message data for all channel.

Parameters

instance	FLEXIO peripheral instance number

Returns

Error or success status returned by API

6.1.5.15 Flexio_Sent_Ip_StartChannelReceiving()

```
Flexio_Sent_Ip_StatusType Flexio_Sent_Ip_StartChannelReceiving (
             const uint8 Instance,
             const uint8 ChannelId )
```

Start receiving SENT data from bus on specific channel.

Start receiving SENT data from bus on specific channel.

Parameters

instance	FLEXIO peripheral instance number
channelID	ID of FLEXIO TIMER channel

Returns

Error or success status returned by API

6.1.5.16 Flexio_Sent_Ip_StopChannelReceiving()

```
Flexio_Sent_Ip_StatusType Flexio_Sent_Ip_StopChannelReceiving (
             const uint8 Instance,
             const uint8 ChannelId )
```

Stop receiving SENT data from bus on specific channel.

Start receiving SENT data from bus on specific channel.

Parameters

instance	FLEXIO peripheral instance number
channelID	ID of FLEXIO TIMER channel

Returns

Error or success status returned by API

6.1.5.17 Sent_Ipw_ControllerInit()

```
void Sent_Ipw_ControllerInit (
            uint8 ControllerId,
            const Sent_Ipw_CtrlConfigType * IpwController )
```

Initialize the Sent driver.

6.1.5.18 Sent_Ipw_ControllerDeInit()

```
void Sent_Ipw_ControllerDeInit (
            uint8 ControllerId )
```

DeInitializes the Sent module.

6.1.5.19 Sent_Ipw_GetFastChannelMsgData()

```
Std_ReturnType Sent_Ipw_GetFastChannelMsgData (
            uint8 ControllerId,
            uint8 ChannelId )
```

This function processing polling of Fast notification.

6.1.5.20 Sent_Ipw_GetFastMsgData()

```
Std_ReturnType Sent_Ipw_GetFastMsgData (
            uint8 ControllerId )
```

This function gets Fast message data for all channel.

6.1.5.21 Sent_Ipw_GetSerialChannelMsgData()

```
Std_ReturnType Sent_Ipw_GetSerialChannelMsgData (
            uint8 ControllerId,
            uint8 ChannelId )
```

This function processing polling of Serial notification.

6.1.5.22 Sent_Ipw_GetSerialMsgData()

```
Std_ReturnType Sent_Ipw_GetSerialMsgData (
            uint8 ControllerId )
```

This function gets Serial message data for all channel.

6.1.5.23 Sent_Ipw_StartChannelReceiving()

```
Std_ReturnType Sent_Ipw_StartChannelReceiving (
            uint8 ControllerId,
            uint8 ChannelId )
```

This function starts to receive data from SENT bus.

6.1.5.24 Sent_Ipw_StopChannelReceiving()

```
Std_ReturnType Sent_Ipw_StopChannelReceiving (
            uint8 ControllerId,
            uint8 ChannelId )
```

This function stops to receive data from SENT bus.

6.2 FLEXIO_SENT_DRIVER

6.2.1 Detailed Description

Data Structures

- struct Flexio_Sent_SerialMsgType

  Flexio_Sent_SerialMsgType. More...
- struct Flexio_Sent_FastMsgType

  Flexio_Sent_FastMsgType. More...
- struct Flexio_ChnlConfig

  Flexio Channel configuration structure. More...
- struct Flexio_DataProcessType

  Flexio data processing structure. More...
- struct Flexio_Sent_Ip_StateType

  Receiver internal context structure. More...
- struct Flexio_Sent_Ip_UserConfigType

  Sent configuration structure. More...
- struct Flexio_CtrlConfigType

  Sent configuration structure. More...

Types Reference

- typedef void(∗ Sent_CallbackType) (Flexio_Sent_Ip_ErrorStatusType ErrorEvent)

  Callback for all peripherals which supports SENT features.
- typedef void(∗ Sent_FastNotificationType) (Flexio_Sent_Ip_StatusType Event, Flexio_Sent_FastMsgType ∗SentFastMsg)

  Callback for all peripherals which supports SENT features.
- typedef void(∗ Sent_SlowNotificationType) (Flexio_Sent_Ip_StatusType Event, Flexio_Sent_SerialMsgType ∗SentSerialMsg)

  Callback for all peripherals which supports SENT features.

Enum Reference

- enum Flexio_Sent_Ip_SyncAsyncType

  Sent_SyncAsyncType.
- enum Flexio_Sent_Ip_DriverType

  Flexio_Sent_Ip_DriverType.
- enum Sent_StatusFastProcessType

  Sent_StatusFastProcessType.
- enum Sent_StatusCRCImplementType

  Sent_StatusCRCImplementType.
- enum Sent_StatusSerialProcessType

  Sent_StatusSerialProcessType.
- enum Flexio_Sent_IP_SlowSerialType

  Flexio_Sent_IP_SlowSerialType.

6.2.2 Data Structure Documentation

6.2.2.1 struct Flexio_Sent_SerialMsgType

Flexio_Sent_SerialMsgType.

Used for received Serial data. It contains the data received, after being processed

Definition at line 219 of file Flexio_Sent_Ip_Types.h.

Data Fields

Type	Name	Description
Flexio_Sent_IP_SlowSerialType	MsgType	Type of serial message (SHORT OR ENHANCED 4BIT/8BIT)
uint8	ChannelId	The channel number on which the data was received
uint8	Instance	The Instance number on which the data was received
uint8	MessageId	Serial Message ID
uint16	MessageData	Data contained in the Serial Message
uint8	SerialCrc	Serial Message CRC

6.2.2.2 struct Flexio_Sent_FastMsgType

Flexio_Sent_FastMsgType.

Used for received Fast data. It contains the data received, after being processed

Definition at line 236 of file Flexio_Sent_Ip_Types.h.

Data Fields

Type	Name	Description
uint8	Instance	The Instance number on which the data was received
uint8	ChannelId	The channel number on which the data was received
uint8	Length	Lenght of the fast message
uint8	DataNibble[((uint8) 0x06U)]	Content of each nibble in the fast message
uint8	StatusCommunication	The Status Communication Nibble
uint8	FastCrc	Fast Message CRC

6.2.2.3 struct Flexio_ChnlConfig

Flexio Channel configuration structure.

Used for configuring each channel of the Flexio Controller

Definition at line 266 of file Flexio_Sent_Ip_Types.h.

Data Fields

Type	Name	Description
uint8	Timer	Which FLEXIO Timer is used for SENT channel
uint8	Pin	Which FLEXIO Pin is input for SENT channel
uint8	NibbleCnt	Data Nibbles supported in this Channel
uint32	TickLengthUs	Tick length is Us
Sent_StatusCRCImplementType	FastCrc	CRC implementation type
Sent_StatusCRCImplementType	SlowCrc	CRC implementation type
boolean	StatusNibbleCrcCalIncluding	Status Nibble Crc Calculate Including
Flexio_Sent_Ip_SyncAsyncType	PausePulse	pause pulse configuration

6.2.2.4 struct Flexio_DataProcessType

Flexio data processing structure.

- n8NibbleVal[DataIndx = 0]=Status & Communication n8NibbleVal[DataIndx = 1]=1st nibble, ... n8Nibble↩ Val[DataIndx = NIBBLE]=NIBBLEth nibble n8NibbleVal[DataIndx = NIBBLE+1]=CRC

Definition at line 292 of file Flexio_Sent_Ip_Types.h.

6.2.2.5 struct Flexio_Sent_Ip_StateType

Receiver internal context structure.

This structure is used by the driver for its internal logic. It must be provided by the application through the Flexio_Sent_Ip_Init() function, then it cannot be freed until the driver is de-initialized using Flexio_Sent_Ip_DeInit(). The application should make no assumptions about the content of this structure.

Definition at line 313 of file Flexio_Sent_Ip_Types.h.

Data Fields

Type	Name	Description
uint8	Instance	
uint8	ResourceAllocation	Count the number of Flexio Timer which is used for SENT
Flexio_Sent_Ip_StatusType	Status	Current Status of the driver
Flexio_Sent_Ip_DriverType	DriverType	The way to handle Fast/Serial message
Sent_CallbackType	SentFastErrorNotif[FEATURE_FLEXIO_MAX_TIMER_COUNT]	Fast CRC callback function
Sent_CallbackType	SentSerialErrorNotif[FEATURE_FLEXIO_MAX_TIMER_COUNT]	Slow CRC callback function
Sent_FastNotificationType	SentFastNotif[FEATURE_FLEXIO_MAX_TIMER_COUNT]	Fast callback function
Sent_SlowNotificationType	SentSerialNotif[FEATURE_FLEXIO_MAX_TIMER_COUNT]	Serial message callback function
Flexio_ChnlConfig	ChnlCfg[FEATURE_FLEXIO_MAX_TIMER_COUNT]	Index of first used internal resource Instance (Timer)

6.2.2.6 struct Flexio_Sent_Ip_UserConfigType

Sent configuration structure.

This structure is used to provide configuration parameters for the Flexio based SENT at initialization time.

Definition at line 334 of file Flexio_Sent_Ip_Types.h.

Data Fields

Type	Name	Description
const uint8	ControllerId	Id of Controller contain channel
const uint8	ControllerHwOffset	HWOffet of Controller contain channel
const uint8	ChannelId	Id of channel
const uint8	ChannelHwOffset	HWOffet of channel
const uint8	SentPin	Pin is used for trigger
const Flexio_Sent_Ip_SyncAsyncType	PausePulse	pause pulse configuration
const uint8	NumberOfNibbles	Configure number of nibbles
const uint32	TickTime	Configure Tick time in Us
const boolean	StatusNibbleCrcCalIncluding	Status Nibble Crc Calculate Including
const Sent_StatusCRCImplementType	SentFastCrcImpelement	CRC implementation for Fast message
const Sent_StatusCRCImplementType	SentSlowCrcImpelement	CRC implementation for Serial message
const Sent_CallbackType	SentFastErrorNotif	Fast callback function
const Sent_CallbackType	SentSerialErrorNotif	Serial message callback function
const Sent_FastNotificationType	SentFastNotif	Fast callback function
const Sent_SlowNotificationType	SentSerialNotif	Serial message callback function

6.2.2.7 struct Flexio_CtrlConfigType

Sent configuration structure.

This structure is used to provide configuration parameters for a controller.

Definition at line 363 of file Flexio_Sent_Ip_Types.h.

Data Fields

Type	Name	Description
const uint8	CtrHwOffset	HWoffet of configured controller
const uint8	CtrHwID	HWID of configured controller
const uint8	ChnlConfigured	Summary of configured channels in a controller.
const Flexio_Sent_Ip_DriverType	DriverType	
const Flexio_Sent_Ip_UserConfigType ∗const ∗const	ChnlConfig	
Flexio_Sent_Ip_StateType ∗	CtrlState	

6.2.3 Types Reference

6.2.3.1 Sent_CallbackType

```
typedef void(* Sent_CallbackType) (Flexio_Sent_Ip_ErrorStatusType ErrorEvent)
```

Callback for all peripherals which supports SENT features.

Definition at line 249 of file Flexio_Sent_Ip_Types.h.

6.2.3.2 Sent_FastNotificationType

```
typedef void(* Sent_FastNotificationType) (Flexio_Sent_Ip_StatusType Event, Flexio_Sent_FastMsgType *Sent←
FastMsg)
```

Callback for all peripherals which supports SENT features.

Definition at line 254 of file Flexio_Sent_Ip_Types.h.

6.2.3.3 Sent_SlowNotificationType

```
typedef void(* Sent_SlowNotificationType) (Flexio_Sent_Ip_StatusType Event, Flexio_Sent_SerialMsgType
*SentSerialMsg)
```

Callback for all peripherals which supports SENT features.

Definition at line 259 of file Flexio_Sent_Ip_Types.h.

6.2.4 Enum Reference

6.2.4.1 Flexio_Sent_Ip_SyncAsyncType

```
enum Flexio_Sent_Ip_SyncAsyncType
```

Sent_SyncAsyncType.

Used for value received by TRESOS interface configuration. Describe the channel type.

Definition at line 119 of file Flexio_Sent_Ip_Types.h.

6.2.4.2 Flexio_Sent_Ip_DriverType

```
enum Flexio_Sent_Ip_DriverType
```

Flexio_Sent_Ip_DriverType.

This to present the way to handle Fast/Serial message

Enumerator

FLEXIO_DRIVER_TYPE_INTERRUPTS	Driver uses interrupts for data transfers
FLEXIO_DRIVER_TYPE_POLLING	Driver is based on polling
FLEXIO_DRIVER_TYPE_DMA	Driver uses DMA for data transfers

Definition at line 129 of file Flexio_Sent_Ip_Types.h.

6.2.4.3 Sent_StatusFastProcessType

```
enum Sent_StatusFastProcessType
```

Sent_StatusFastProcessType.

To check data processing when convert timer data to SENT's nibble value

Enumerator

STATUS_SENT_FAST_IDLE	Status before receiving sync/calibration pulse
STATUS_SENT_FAST_SYNC_CALIB	sync/calibration pulse is checked done
STATUS_SENT_FAST_CHECKED	Frame receiving is done
STATUS_SENT_FAST_RE_SYNC_CALIB	CRC checking is done
STATUS_SENT_FAST_PAUSE_PULSE	CRC checking is done in case using pause pulse

Definition at line 163 of file Flexio_Sent_Ip_Types.h.

6.2.4.4 Sent_StatusCRCImplementType

```
enum Sent_StatusCRCImplementType
```

Sent_StatusCRCImplementType.

To chose the way to Fast/Slow message implement CRC

Enumerator

SENT_RECOMMENDED_IMPLEMENTATION_16_ELEMENT	CRC Recommended Implementation: Implementation using 16 Element Array
SENT_LEGACY_IMPLEMENTATION_16_ELEMENT	CRC Legacy Implementation: Implementation using 16 Element Array

Definition at line 178 of file Flexio_Sent_Ip_Types.h.

6.2.4.5 Sent_StatusSerialProcessType

```
enum Sent_StatusSerialProcessType
```

Sent_StatusSerialProcessType.

To check data processing when covert timer data to SENT's nibble value

Enumerator

STATUS_SENT_SERIAL_IDLE	Status of serial message before processing
STATUS_SENT_SERIAL_SHORT	Serial message is SHORT SERIAL type
STATUS_SENT_SERIAL_ENHANCED	Serial message is ENHANCED SERIAL type

Definition at line 193 of file Flexio_Sent_Ip_Types.h.

6.2.4.6 Flexio_Sent_IP_SlowSerialType

```
enum Flexio_Sent_IP_SlowSerialType
```

Flexio_Sent_IP_SlowSerialType.

Enumerator

SENT_SHORT_SERIAL	Serial message is SHORT SERIAL type
SENT_ENHANCED_SERIAL_4_ID	Serial message is ENHANCED SERIAL type
SENT_ENHANCED_SERIAL_8_ID	Serial message is ENHANCED SERIAL type

Definition at line 206 of file Flexio_Sent_Ip_Types.h.

Chapter 7

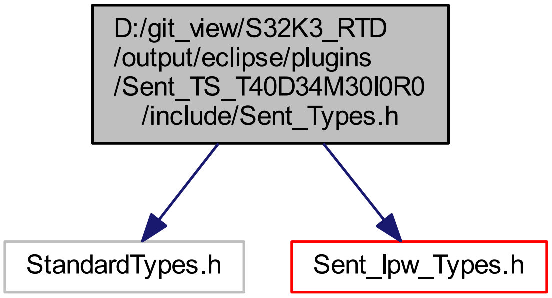
File Documentation

7.1 D:/git_view/S32K3_RTD/output/eclipse/plugins/Sent_TS_↩ T40D34M30I0R0/include/Sent_Types.h File Reference

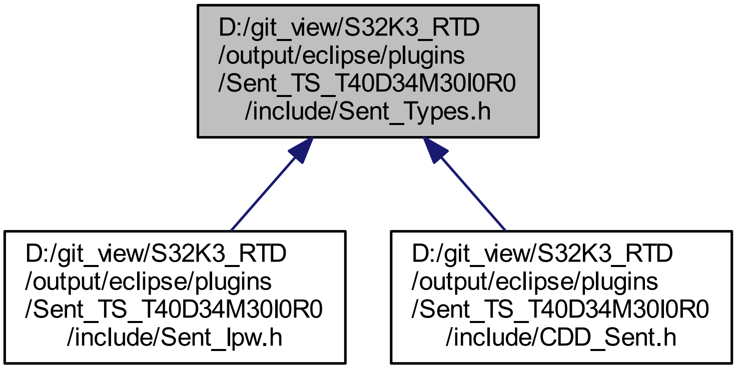
AUTOSAR Sent - Sent generic types file.

```
#include "StandardTypes.h"
#include "Sent_Ipw_Types.h"
```

Include dependency graph for Sent_Types.h:



This graph shows which files directly or indirectly include this file:



Data Structures

- struct Sent_ControllerConfigType

  Structure that contains Sent Hw configuration. More...
- struct Sent_ConfigType

  This type contains initialization data. More...

Enumerations

- enum Sent_DriverStatusType

  Sent_DriverStatusType.

7.1.1 Detailed Description

AUTOSAR Sent - Sent generic types file.

Version

3.0.0

Contains generic Sent types and structures.

How to Reach Us:

Home Page:
nxp.com

Web Support:
nxp.com/support