

# User Manual

for S32K1 I2S Driver

Document Number: UM2I2SASR4.4 Rev0000R1.0.1 Rev. 1.0

<b>1 Revision History</b>	<b>2</b>
<b>2 Introduction</b>	<b>3</b>
2.1 Supported Derivatives	3
2.2 Overview	4
2.3 About This Manual	5
2.4 Acronyms and Definitions	6
2.5 Reference List	6
<b>3 Driver</b>	<b>7</b>
3.1 Requirements	7
3.2 Driver Design Summary	7
3.3 Hardware Resources	8
3.4 Deviations from Requirements	8
3.5 Driver Limitations	8
3.6 Driver usage and configuration tips	8
3.6.1 SAI and Flexio I2S initializations	8
3.6.2 SAI and Flexio I2S data transmission	9
3.6.3 I2s over SAI DMA transfer	9
3.6.4 I2s over SAI full duplex mode	9
3.6.5 Flexio_I2s communicates with Sai	10
3.6.6 SAI auto disable clock	10
3.7 Runtime errors	10
3.8 Symbolic Names Disclaimer	10
<b>4 Tresos Configuration Plug-in</b>	<b>11</b>
4.1 Module I2s	13
4.2 Container GeneralConfiguration	13
4.3 Parameter I2sDevErrorDetect	14
4.4 Parameter I2sMulticoreSupport	14
4.5 Parameter I2sEnableUserModeSupport	15
4.6 Parameter I2sVersionInfoApi	16
4.7 Parameter I2sTimeoutMethod	16
4.8 Parameter I2sTimeoutVal	17
4.9 Parameter I2sNumLogicChn	17
4.10 Container I2sSaiSupport	17
4.11 Parameter I2sSaiEnable	18
4.12 Parameter SaiIpDevErrorDetect	18
4.13 Parameter SaiDmaFeature	19
4.14 Parameter SaiAutoDisableClockFeature	19
4.15 Parameter SaiFifoPackFeature	20

4.16 Parameter SaiBitClkSwapSupport . . . . .	20
4.17 Parameter SaiEnableDataMasking . . . . .	20
4.18 Container I2sFlexioSupport . . . . .	21
4.19 Parameter I2sFlexioI2sEnable . . . . .	21
4.20 Parameter FlexioI2sIpDevErrorDetect . . . . .	22
4.21 Parameter FlexioI2sDmaFeature . . . . .	22
4.22 Container I2sConfigSet . . . . .	23
4.23 Container I2sConfiguration . . . . .	23
4.24 Parameter I2sIpType . . . . .	23
4.25 Parameter I2sLogicalChannelId . . . . .	24
4.26 Parameter I2sCallback . . . . .	24
4.27 Reference I2sModuleRef . . . . .	25
4.28 Reference I2sChannelEcucPartitionRef . . . . .	25
4.29 Container SaiHwConfiguration . . . . .	26
4.30 Parameter SaiInstanceId . . . . .	26
4.31 Parameter SaiTxRxConfiguration . . . . .	27
4.32 Parameter SaiMode . . . . .	27
4.33 Parameter SaiMasterClkSrc . . . . .	28
4.34 Parameter SaiSyncType . . . . .	28
4.35 Parameter SaiBitClkSwap . . . . .	29
4.36 Parameter SaiBitClkNegPolar . . . . .	29
4.37 Parameter SaiSampleRate . . . . .	30
4.38 Parameter SaiDataLine0 . . . . .	30
4.39 Parameter SaiDataLine1 . . . . .	30
4.40 Parameter SaiDataLine2 . . . . .	31
4.41 Parameter SaiDataLine3 . . . . .	31
4.42 Parameter SaiFrameSize . . . . .	32
4.43 Parameter SaiSyncWidth . . . . .	32
4.44 Parameter SaiChannelMode . . . . .	33
4.45 Parameter SaiMsbFirst . . . . .	33
4.46 Parameter SaiEnableDebug . . . . .	34
4.47 Parameter SaiSyncEarly . . . . .	34
4.48 Parameter SaiSyncNegPolar . . . . .	35
4.49 Parameter SaiWord0Width . . . . .	35
4.50 Parameter SaiWordNWidth . . . . .	36
4.51 Parameter SaiFirstBitIndex . . . . .	36
4.52 Parameter SaiRunErrorReport . . . . .	36
4.53 Parameter SaiSyncErrorReport . . . . .	37
4.54 Parameter SaiFrameStartReport . . . . .	37
4.55 Parameter SaiFifoPackEnable . . . . .	38

4.56 Parameter SaiMuxMode . . . . .	38
4.57 Parameter SaiTransferType . . . . .	39
4.58 Parameter SaiElementSize . . . . .	39
4.59 Parameter SaiBufferAccessSize . . . . .	40
4.60 Parameter SaiDataLineCount . . . . .	40
4.61 Parameter SaiWordFlagIndex . . . . .	41
4.62 Parameter SaiBitClkInput . . . . .	41
4.63 Parameter SaiContOnErr . . . . .	42
4.64 Parameter SaiSyncOnDemand . . . . .	42
4.65 Reference SaiClkSrcRef . . . . .	43
4.66 Reference SaiDmaChannel0 . . . . .	43
4.67 Reference SaiDmaChannel1 . . . . .	44
4.68 Reference SaiDmaChannel2 . . . . .	44
4.69 Reference SaiDmaChannel3 . . . . .	44
4.70 Container FlexioHwConfiguration . . . . .	45
4.71 Parameter FlexioI2sInstanceId . . . . .	45
4.72 Parameter FlexioI2sBitsWidth . . . . .	46
4.73 Parameter FlexioI2sMode . . . . .	46
4.74 Parameter FlexioI2sBaudRate . . . . .	47
4.75 Parameter FlexioI2sTransferType . . . . .	47
4.76 Parameter FlexioI2sRxLine . . . . .	48
4.77 Parameter FlexioI2sWsLine . . . . .	48
4.78 Parameter FlexioI2sCallbackParam . . . . .	49
4.79 Reference FlexioI2sClkSrcRef . . . . .	49
4.80 Reference FlexioI2sTxLine . . . . .	50
4.81 Reference FlexioI2sSckLine . . . . .	50
4.82 Reference FlexioI2sDmaTxChannel . . . . .	51
4.83 Reference FlexioI2sDmaRxChannel . . . . .	51
4.84 Container CommonPublishedInformation . . . . .	51
4.85 Parameter ArReleaseMajorVersion . . . . .	53
4.86 Parameter ArReleaseMinorVersion . . . . .	53
4.87 Parameter ArReleaseRevisionVersion . . . . .	54
4.88 Parameter ModuleId . . . . .	54
4.89 Parameter SwMajorVersion . . . . .	55
4.90 Parameter SwMinorVersion . . . . .	55
4.91 Parameter SwPatchVersion . . . . .	56
4.92 Parameter VendorApiInfix . . . . .	56
4.93 Parameter VendorId . . . . .	57
4.94 Container I2sClockReferencePoint . . . . .	57
4.95 Reference I2sClockReference . . . . .	57

<b>5 Module Index</b>	<b>59</b>
5.1 Software Specification . . . . .	59
<b>6 Module Documentation</b>	<b>60</b>
6.1 I2s Driver . . . . .	60
6.1.1 Detailed Description . . . . .	60
6.1.2 Data Structure Documentation . . . . .	62
6.1.3 Macro Definition Documentation . . . . .	63
6.1.4 Types Reference . . . . .	66
6.1.5 Enum Reference . . . . .	66
6.1.6 Function Reference . . . . .	67
6.2 FLEXIO_I2S IPL . . . . .	71
6.2.1 Detailed Description . . . . .	71
6.3 I2s Driver . . . . .	72
6.3.1 Detailed Description . . . . .	72
6.3.2 Data Structure Documentation . . . . .	72
6.3.3 Enum Reference . . . . .	72
6.4 SAI IPL . . . . .	75
6.4.1 Detailed Description . . . . .	75
6.5 SAI IPL . . . . .	76
6.5.1 Detailed Description . . . . .	76
6.5.2 Macro Definition Documentation . . . . .	76



## Chapter 1

### Revision History

Revision	Date	Author	Description
1.0	24.02.2022	NXP RTD Team	Prepared for release RTD S32K1 Version 1.0.1

## Chapter 2

### Introduction

- [Supported Derivatives](#)
- [Overview](#)
- [About This Manual](#)
- [Acronyms and Definitions](#)
- [Reference List](#)

This User Manual describes the NXP Semiconductor I2s driver for S32K1XX. I2s driver configuration parameters and deviations from the specification are described in Driver chapter of this document. I2s driver requirements and APIs are described in the I2s driver software specification document.

Note: The Sai driver has been renamed to I2s starting with release RTD S32K1 Version 1.0.1. The API of the driver maintains the same functionality as previous releases. However, API names have been changed, beginning with the prefix "I2s\_" instead of "Sai\_". Existing application code should be updated to be in accordance with the new API naming.

### 2.1 Supported Derivatives

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

- s32k116\_qfn32
- s32k116\_lqfp48
- s32k118\_lqfp48
- s32k118\_lqfp64
- s32k142\_lqfp48
- s32k142\_lqfp64
- s32k142\_lqfp100

- s32k142w\_lqfp48
- s32k142w\_lqfp64
- s32k144\_lqfp48
- s32k144\_lqfp64
- s32k144\_lqfp100
- s32k144\_mapbga100
- s32k144w\_lqfp48
- s32k144w\_lqfp64
- s32k146\_lqfp64
- s32k146\_lqfp100
- s32k146\_mapbga100
- s32k146\_lqfp144
- s32k148\_lqfp100
- s32k148\_mapbga100
- s32k148\_lqfp144
- s32k148\_lqfp176

All of the above microcontroller devices are collectively named as S32K1.

## 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
CAN	Controller Area Network
C/CPP	C and C++ Source Code
CDD	Complex Device Driver
CS	Chip Select
CTU	Cross Trigger Unit
DEM	Diagnostic Event Manager
DET	Development Error Tracer
DMA	Direct Memory Access
ECU	Electronic Control Unit
FIFO	First In First Out
LSB	Least Significant Bit
MCU	Micro Controller Unit
MSB	Most Significant Bit
N/A	Not Applicable
RAM	Random Access Memory
SAI	Inter-IC Sound
SIU	Systems Integration Unit
SWS	Software Specification
XML	Extensible Markup Language

## 2.5 Reference List

#	Title	Version
1	S32K1xx Series Reference Manual	Rev. 14, 09/2021
2	S32K1xx Data Sheet	Rev. 14, 08/2021
3	Errata S32K116 (0N96V)	Rev. 22/OCT/2021
4	Errata S32K118 (0N97V)	Rev. 22/OCT/2021
5	Errata S32K142 (0N33V)	Rev. 22/OCT/2021
6	Errata S32K144 (0N57U)	Rev. 22/OCT/2021
7	Errata S32K144W (0P64A)	Rev. 22/OCT/2021
8	Errata S32K146 (0N73V)	Rev. 22/OCT/2021
9	Errata S32K148 (0N20V)	Rev. 22/OCT/2021

## Chapter 3

### Driver

- [Requirements](#)
- [Driver Design Summary](#)
- [Hardware Resources](#)
- [Deviations from Requirements](#)
- [Driver Limitations](#)
- [Driver usage and configuration tips](#)
- [Runtime errors](#)
- [Symbolic Names Disclaimer](#)

### 3.1 Requirements

I2s 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 Inter-IC Sound (I2S) driver is implemented as an Autosar complex device driver and provides an interface that supports full-duplex and half-duplex serial interfaces with frame synchronization such as I2S, AC97, TDM, and codec/DSP interfaces. The driver uses the SAI and Flexio I2S peripherals in S32K1XX microcontrollers. The driver can be used to configure the SAI and Flexio I2S peripherals and initiate synchronous and asynchronous data transfers. The peripherals can be configured to operate as a receiver or transmitter, operating in slave and master mode. Hardware and software settings can be configured using an Autosar standard configuration tool. The information required for a I2s data transfer will be configured in a data structure that will be sent as parameter to the API of the driver.

### 3.3 Hardware Resources

The I2S driver uses the SAI and Flexio I2S hardware IPs. For more details and availability please check the device reference manual.

### 3.4 Deviations from Requirements

The driver deviates from the AUTOSAR CDD\_I2S Driver software specification in some places. The table identifies the AUTOSAR requirements that are not fully implemented, implemented differently, not available, not testable or out of scope for the I2S Driver.

Term	Definition
N/A	Not available
N/T	Not testable
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, implemented differently, not available, not testable or out of scope for the driver.

Requirement	Status	Description	Notes
Req_Id	N/S	These requirements are not applicable to this specification.	Not a requirement.

### 3.5 Driver Limitations

I2S driver limitations:

1. SAI FIFO packing feature is not supported for DMA transfer mode.
2. The misra violations are not fixed regarding the rule 8.13.
3. Flexio I2S is not supported DMA mode.
4. Flexio I2S can only initialize one master or one slave.

### 3.6 Driver usage and configuration tips

#### 3.6.1 SAI and Flexio I2S initializations

In order to use the I2S driver it must be first initialized in transmit/receive for SAI and master/slave for Flexio I2S for each I2S logical channel that mapping with the desired hardware unit using functions [I2s\\_Init\(\)](#). Once initialized, it cannot be initialized again until it is de-initialized using [I2s\\_DeInit\(\)](#).

### 3.6.2 SAI and Flexio I2S data transmission

In each mode (transmit/receive for SAI and master/slave for Flexio I2S) are available two types of transfers:

- Synchronous (blocking): the functions which initiate blocking transfers with configured timeout for transmission. If time expires `I2s_SyncTransmit()` will return error and the transmission will be aborted.
- Asynchronous (non-blocking): `I2s_AsyncTransmit()` will return status immediately and complete after the last data is transferred.

Uses `I2s_AbortTransmit()` to abort or stop the ongoing transfer.

Uses `I2s_GetStatus()` to get the information of the current transfer.

Uses `I2s_GetVersionInfo()` to get version information of this module.

### 3.6.3 I2s over SAI DMA transfer

- DMA module has to be initialized prior to usage in DMA mode; also, DMA channels need to be allocated by the application (the driver only takes care of configuring the DMA channels received in the configuration structure)
- There is a difference in ChannelEnable field usage between interrupt and dma mode:  
In interrupt mode, if mux line is enabled then user must turn on only one bit in ChannelEnable, which will be the data line to output data. Number of data buffers to be muxed is specified in ChannelCount field.  
In DMA mode, if mux line is enabled then user must turn on number of bits equal to number of data buffers to be muxed. The data lines corresponding to these bits will output the same as each other.  
Also in DMA mode, if a mux mode is selected, ChannelEnable must be turned on from bit 0, and immediately above (for example turning on bit 0 and bit 2 is not a correct configuration).
- When multiple SAI channel is enabled, number of DMA channels must be equal number of SAI channels. And in DMA component:  
Channel arbitration must be set to "Fixed priority".  
The first DMA channel must have the lowest DMA channel number and request source must be set properly (for example SAI0\_TX). Other DMA channels must have request field set to "No request".

### 3.6.4 I2s over SAI full duplex mode

- Configuration:
  - Initialize both Transmitter and Receiver with one is Master-Asynchronous and another is Slave-↔ Synchronous.
  - Setup different data lines between Transmitter and Receiver (BCLK and FS will be shared).
  - The SAI protocol configuration must be the same between Transmitter and Receiver.
  - Transmit data pins are configured for TDM mode (MASK\_TRISTATE - transmit data pins are tri-stated when slots are masked or channels are disabled).
- Transmission:
  - Slave request (non-blocking) should be called first (to prepare data transmission) then Master request shall start the transmission.
  - Master request can be used alone if do not need any action from Slave.

### 3.6.5 Flexio\_I2s communicates with Sai

- Flexio\_I2s is Master and Sai is Slave:
  - For Sai, A valid frame sync is also ignored (slave mode) or not generated (master mode) for the first 4 bit clocks but Flexio\_I2s does not
  - When Sai receives data from Flexio\_I2s, it ignores the falling/rising edge on WS line during the 4 first bit clocks until next falling/rising edge, so depending on Frame Sync polarity, the actual received buffer (by Sai) is less than the transferred buffer size (from Flexio\_I2s) 1 or 2 words
  - When Sai transmits data, Sai also ignores the falling/rising edge on WS line and does not generate DATA during 4 first bit clocks until next falling/rising edge Flexio\_I2s needs to generate 1 more frame clock, the actual received buffer (by Flexio\_i2s) is greater than the transferred buffer (from Sai) 2 words
- Sai is Master and Flexio\_I2s is Slave:
  - The above issue does not happen when Sai is master and Flexio\_I2s is slave, because the first 4 bit clocks without WS edge do not affect Flexio\_I2s, transmission start is just delayed with 4 bit clocks.

### 3.6.6 SAI auto disable clock

If there is no more request (transmit/receive) added in a long time after, "autoDisableClock" can be set in the last request to disable clock generation until the next request.

## 3.7 Runtime errors

The driver generates the following DET runtime errors at runtime.

Function	Error Code	Condition triggering the error
<a href="#">I2s_SyncTransmit()</a>	I2S_E_TIMEOUT	Timed out. Ongoing transmission timed out.

## 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 [I2s](#)
  - Container [GeneralConfiguration](#)
    - \* Parameter [I2sDevErrorDetect](#)
    - \* Parameter [I2sMulticoreSupport](#)
    - \* Parameter [I2sEnableUserModeSupport](#)
    - \* Parameter [I2sVersionInfoApi](#)
    - \* Parameter [I2sTimeoutMethod](#)
    - \* Parameter [I2sTimeoutVal](#)
    - \* Parameter [I2sNumLogicChn](#)
    - \* Container [I2sSaiSupport](#)
      - Parameter [I2sSaiEnable](#)
      - Parameter [SaiIpDevErrorDetect](#)
      - Parameter [SaiDmaFeature](#)
      - Parameter [SaiAutoDisableClockFeature](#)
      - Parameter [SaiFifoPackFeature](#)
      - Parameter [SaiBitClkSwapSupport](#)
      - Parameter [SaiEnableDataMasking](#)
    - \* Container [I2sFlexioSupport](#)
      - Parameter [I2sFlexioI2sEnable](#)
      - Parameter [FlexioI2sIpDevErrorDetect](#)
      - Parameter [FlexioI2sDmaFeature](#)
  - Container [I2sConfigSet](#)
    - \* Container [I2sConfiguration](#)
      - Parameter [I2sIpType](#)
      - Parameter [I2sLogicalChannelId](#)
      - Parameter [I2sCallback](#)
      - Reference [I2sModuleRef](#)
      - Reference [I2sChannelEcucPartitionRef](#)
    - \* Container [SaiHwConfiguration](#)
      - Parameter [SaiInstanceId](#)

- Parameter [SaiTxRxConfiguration](#)
- Parameter [SaiMode](#)
- Parameter [SaiMasterClkSrc](#)
- Parameter [SaiSyncType](#)
- Parameter [SaiBitClkSwap](#)
- Parameter [SaiBitClkNegPolar](#)
- Parameter [SaiSampleRate](#)
- Parameter [SaiDataLine0](#)
- Parameter [SaiDataLine1](#)
- Parameter [SaiDataLine2](#)
- Parameter [SaiDataLine3](#)
- Parameter [SaiFrameSize](#)
- Parameter [SaiSyncWidth](#)
- Parameter [SaiChannelMode](#)
- Parameter [SaiMsbFirst](#)
- Parameter [SaiEnableDebug](#)
- Parameter [SaiSyncEarly](#)
- Parameter [SaiSyncNegPolar](#)
- Parameter [SaiWord0Width](#)
- Parameter [SaiWordNWidth](#)
- Parameter [SaiFirstBitIndex](#)
- Parameter [SaiRunErrorReport](#)
- Parameter [SaiSyncErrorReport](#)
- Parameter [SaiFrameStartReport](#)
- Parameter [SaiFifoPackEnable](#)
- Parameter [SaiMuxMode](#)
- Parameter [SaiTransferType](#)
- Parameter [SaiElementSize](#)
- Parameter [SaiBufferAccessSize](#)
- Parameter [SaiDataLineCount](#)
- Parameter [SaiWordFlagIndex](#)
- Parameter [SaiBitClkInput](#)
- Parameter [SaiContOnErr](#)
- Parameter [SaiSyncOnDemand](#)
- Reference [SaiClkSrcRef](#)
- Reference [SaiDmaChannel0](#)
- Reference [SaiDmaChannel1](#)
- Reference [SaiDmaChannel2](#)
- Reference [SaiDmaChannel3](#)
- \* Container [FlexioHwConfiguration](#)
  - Parameter [FlexioI2sInstanceId](#)
  - Parameter [FlexioI2sBitsWidth](#)
  - Parameter [FlexioI2sMode](#)
  - Parameter [FlexioI2sBaudRate](#)
  - Parameter [FlexioI2sTransferType](#)
  - Parameter [FlexioI2sRxLine](#)
  - Parameter [FlexioI2sWsLine](#)



- Parameter [FlexioI2sCallbackParam](#)
- Reference [FlexioI2sClkSrcRef](#)
- Reference [FlexioI2sTxLine](#)
- Reference [FlexioI2sSckLine](#)
- Reference [FlexioI2sDmaTxChannel](#)
- Reference [FlexioI2sDmaRxChannel](#)
- Container [CommonPublishedInformation](#)
  - \* Parameter [ArReleaseMajorVersion](#)
  - \* Parameter [ArReleaseMinorVersion](#)
  - \* Parameter [ArReleaseRevisionVersion](#)
  - \* Parameter [ModuleId](#)
  - \* Parameter [SwMajorVersion](#)
  - \* Parameter [SwMinorVersion](#)
  - \* Parameter [SwPatchVersion](#)
  - \* Parameter [VendorApiInfix](#)
  - \* Parameter [VendorId](#)
- Container [I2sClockReferencePoint](#)
  - \* Reference [I2sClockReference](#)

## 4.1 Module I2s

Configuration of the I2s module.

Included containers:

- [GeneralConfiguration](#)
- [I2sConfigSet](#)
- [CommonPublishedInformation](#)
- [I2sClockReferencePoint](#)

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

## 4.2 Container GeneralConfiguration

This container contains the global configuration parameters of the Non-Autosar I2s driver.

Included subcontainers:

- [I2sSaiSupport](#)
- [I2sFlexioSupport](#)

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

### 4.3 Parameter I2sDevErrorDetect

Switches the Development Error Detection and Notification ON or OFF.

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	true

### 4.4 Parameter I2sMulticoreSupport

This parameter globally enables the possibility to support multicore. If I2sMulticoreSupport is disabled, then for all the variants no partition shall be defined. If I2sMulticoreSupport is enabled, at least one EcucPartition needs to be defined (in all variants).

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

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	false

## 4.5 Parameter I2sEnableUserModeSupport

### I2s Enable User Mode Support

When this parameter is enabled, the I2s module will adapt to run from User Mode, with the following measure : configuring REG\_PROT for I2s IPs so that the registers under protection can be accessed from user mode by setting UAA bit in REG\_PROT\_GCR to 1 for more information and availability on this platform, please see chapter "User Mode Support" in IM.

True: I2s module will adapt to run from User Mode.

False:I2s module will not adapt to run from User Mode.

If this parameter is not ediatable, that means I2s driver can run in User Mode.

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

## 4.6 Parameter I2sVersionInfoApi

Adds / removes the service I2s\_GetVersionInfo() from the code.

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

## 4.7 Parameter I2sTimeoutMethod

Configures the timeout method for I2s.

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

Note: If OSIF\_COUNTER\_SYSTEM or OSIF\_COUNTER\_CUSTOM are selected make sure the corresponding timer is enabled in OsIf General configuration.

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: POST-BUILD
defaultValue	OSIF_COUNTER_DUMMY
literals	['OSIF_COUNTER_DUMMY', 'OSIF_COUNTER_SYSTEM', 'OSIF_COUNTER_CUSTOM']

## 4.8 Parameter I2sTimeoutVal

This is a timeout (microseconds) value which is used to wait for each synchronization transfer

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: PRE-COMPILE VARIANT-PRE-COMPILE: PRE-COMPILE
defaultValue	1000
max	4294967295
min	1

## 4.9 Parameter I2sNumLogicChn

Defines number of logical channel ID shall be used.

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: PRE-COMPILE VARIANT-PRE-COMPILE: PRE-COMPILE
defaultValue	4
max	256
min	1

## 4.10 Container I2sSaiSupport

Vendor specific: Container for the I2s Sai related configuration parameters.

Included subcontainers:

- None

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

### 4.11 Parameter I2sSaiEnable

Vendor specific: Enable/Disable Sai support.

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

### 4.12 Parameter SaiIpDevErrorDetect

Switches the Development Error Detection and Notification ON or OFF.

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

### 4.13 Parameter SaiDmaFeature

Enable DMA feature for transferring and Notification ON or OFF.

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

### 4.14 Parameter SaiAutoDisableClockFeature

Sai Auto Disable Clock Feature. When enabled, allows the driver to disable clock generation after transmission completed or aborted. The clock generation will be started again at next transmission.

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

## 4.15 Parameter SaiFifoPackFeature

Enable FIFO packing feature. When this parameter is enabled, the Sai FIFO Packing Enable can be configured.

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

## 4.16 Parameter SaiBitClkSwapSupport

Sai Enable Bit Clock Swap Support

When this parameter is enabled, the Sai Bit Clk Swap can be configured.

If this parameter is not ediatable, that means this feature is not available on this device.

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

## 4.17 Parameter SaiEnableDataMasking

Enable to allow masking of sent or received data. Data will be masked according to the SaiChannelMode selection.



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

## 4.18 Container I2sFlexioSupport

Vendor specific: Container for the I2s Flexio related configuration parameters.

Included subcontainers:

- None

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

## 4.19 Parameter I2sFlexioI2sEnable

Vendor specific: Enable/Disable Flexio support.

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	true

## 4.20 Parameter FlexioI2sIpDevErrorDetect

Switches the Development Error Detection and Notification ON or OFF.

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	true

## 4.21 Parameter FlexioI2sDmaFeature

Enable DMA feature for transferring and Notification ON or OFF.

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

## 4.22 Container I2sConfigSet

This container contains the configuration parameters and sub containers of the I2s module.

Included subcontainers:

- [I2sConfiguration](#)
- [SaiHwConfiguration](#)
- [FlexioHwConfiguration](#)

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

## 4.23 Container I2sConfiguration

This container contains the configuration (parameters) of the I2s Controller(s).

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

## 4.24 Parameter I2sIpType

Selects Hardware IP.

Property	Value
type	ECUC-ENUMERATION-PARAM-DEF

Property	Value
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	SAI
literals	['SAI', 'FLEXIO_I2S']

## 4.25 Parameter I2sLogicalChannelId

Identifies the I2s Logical channel ID corresponding with I2s configuration.

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: PRE-COMPILE VARIANT-PRE-COMPILE: PRE-COMPILE
defaultValue	1
max	255
min	0

## 4.26 Parameter I2sCallback

I2s callback. This function will be called for all I2s events.

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

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	NULL_PTR

## 4.27 Reference I2sModuleRef

Maps an I2s channel to zero or one ECUC partition to limit the access to this channel group. The ECUC partitions referenced are a subset of the ECUC partitions where the I2s driver is mapped to.

Property	Value
type	ECUC-CHOICE-REFERENCE-DEF
origin	NXP
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	true
valueConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: POST-BUILD
requiresSymbolicNameValue	False
destinations	['/TS_T40D2M10I1R0/I2s/I2sConfigSet/SaiHwConfiguration', '/TS_T40D2M10I1R0/I2s/I2sConfigSet/FlexioHwConfiguration']

## 4.28 Reference I2sChannelEcucPartitionRef

Maps a I2s channel to zero or one ECUC partition to limit the access to this channel group. The ECUC partitions referenced are a subset of the ECUC partitions where the I2s driver is mapped to.

Property	Value
type	ECUC-REFERENCE-DEF
origin	AUTOSAR_ECUC
lowerMultiplicity	0
upperMultiplicity	Infinite
postBuildVariantMultiplicity	true

Property	Value
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/EcuDefs/EcuC/EcuPartitionCollection/EcuPartition

## 4.29 Container SaiHwConfiguration

Configuration of a Sai module available on the platform.

Included subcontainers:

- None

Property	Value
type	ECUC-PARAM-CONF-CONTAINER-DEF
lowerMultiplicity	0
upperMultiplicity	4
postBuildVariantMultiplicity	false
multiplicityConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: PRE-COMPILE

## 4.30 Parameter SaiInstanceId

Identifies the Sai Instance.

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

Property	Value
defaultValue	0
max	1
min	0

### 4.31 Parameter SaiTxRxConfiguration

Selects Tx or Rx configuration.

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	Transmitter
literals	['Transmitter', 'Receiver']

### 4.32 Parameter SaiMode

Master or Slave.

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	Master
literals	['Master', 'Slave']

### 4.33 Parameter SaiMasterClkSrc

Select master clock source.

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	BUS_CLK
literals	['BUS_CLK', 'EXTERNAL_CLK', 'SOSC_DIV1_CLK', 'EXTERNAL_OTHER_CLK']

### 4.34 Parameter SaiSyncType

SAI run in sync or async mode.

ASYNCR: Independent clock.

SYNC\_WITH\_OTHER: Bit clock and frame sync signal is taken from transmitter/receiver.

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	ASYNCR
literals	['ASYNCR', 'SYNC_WITH_OTHER']



### 4.35 Parameter SaiBitClkSwap

This field swaps the bit clock used by the transmitter. When the transmitter is configured in asynchronous mode and this bit is set, the transmitter is clocked by the receiver bit clock (RX\_BCLK). This allows the transmitter and receiver to share the same bit clock, but the transmitter continues to use the transmit frame sync (TX\_SYNC).

When the transmitter is configured in synchronous mode, the transmitter BCS field and receiver BCS field must be set to the same value. When both are set, the transmitter and receiver are both clocked by the transmitter bit clock (TX\_BCLK) but use the receiver frame sync (RX\_SYNC).

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

### 4.36 Parameter SaiBitClkNegPolar

True if bit clock is negative polar (active low with drive outputs on falling edge and sample inputs on rising edge), false otherwise.

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

## 4.37 Parameter SaiSampleRate

Sample rate for the Sai module.

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-POST-BUILD: POST-BUILD
	VARIANT-PRE-COMPILE: PRE-COMPILE
defaultValue	16000
max	4294967295
min	0

## 4.38 Parameter SaiDataLine0

Enable or disable Sai Data Line 0.

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

## 4.39 Parameter SaiDataLine1

Enable or disable Sai Data Line 1.

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

## 4.40 Parameter SaiDataLine2

Enable or disable Sai Data Line 2.

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

## 4.41 Parameter SaiDataLine3

Enable or disable Sai Data Line 3.

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

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

## 4.42 Parameter SaiFrameSize

Frame size in number of words. When in mux line mode, the number of unmasked word in a frame must be divisible by number of channels.

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-POST-BUILD: POST-BUILD
	VARIANT-PRE-COMPILE: PRE-COMPILE
defaultValue	1
max	16
min	1

## 4.43 Parameter SaiSyncWidth

Active sync width in number of bit clocks (must not longer than first word width).

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

Property	Value
postBuildVariantValue	true
valueConfigClasses	VARIANT-POST-BUILD: POST-BUILD
	VARIANT-PRE-COMPILE: PRE-COMPILE
defaultValue	1
max	32
min	1

## 4.44 Parameter SaiChannelMode

Configures if transmit data pins are configured for TDM mode or Output mode.

MASK\_TRISTATE - TDM mode, transmit data pins are tri-stated when slots are masked or channels are disabled.

MASK\_ZERO - Output mode, transmit data pins are never tri-stated and will output zero when slots are masked or channels are disabled.

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	MASK_TRISTATE
literals	['MASK_TRISTATE', 'MASK_ZERO']

## 4.45 Parameter SaiMsbFirst

True if data is MSB first, false if LSB first.

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

Property	Value
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	true
valueConfigClasses	VARIANT-POST-BUILD: POST-BUILD
	VARIANT-PRE-COMPILE: PRE-COMPILE
default Value	MSB
literals	['MSB', 'LSB']

## 4.46 Parameter SaiEnableDebug

Enables/disables transmitter or receiver operation in Debug mode.

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

## 4.47 Parameter SaiSyncEarly

True if frame sync is one bit clock early.

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

Property	Value
	VARIANT-PRE-COMPILE: PRE-COMPILE
defaultValue	false

## 4.48 Parameter SaiSyncNegPolar

True if frame sync is negative polarity (active low), false otherwise.

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

## 4.49 Parameter SaiWord0Width

Configures the number of bits in the first word in each frame.

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-POST-BUILD: POST-BUILD
	VARIANT-PRE-COMPILE: PRE-COMPILE
defaultValue	16
max	32
min	1

## 4.50 Parameter SaiWordNWidth

Configures the number of bits in each word except the first word in the frame.

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-POST-BUILD: POST-BUILD
	VARIANT-PRE-COMPILE: PRE-COMPILE
defaultValue	16
max	32
min	8

## 4.51 Parameter SaiFirstBitIndex

Index from LSB of first bit to be transmitted/received, valid range from 0 to 31.

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-POST-BUILD: POST-BUILD
	VARIANT-PRE-COMPILE: PRE-COMPILE
defaultValue	1
max	31
min	0

## 4.52 Parameter SaiRunErrorReport

Execute the callback function on underrun/overflow event.



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
default Value	false

### 4.53 Parameter SaiSyncErrorReport

Execute the callback function when an error on external frame sync has occurred.

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
default Value	false

### 4.54 Parameter SaiFrameStartReport

Execute the callback function on frame start event event.

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

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

## 4.55 Parameter SaiFifoPackEnable

Sai FIFO Packing Enable

Enable FIFO packing base on element size:

1-byte: 8-bit FIFO packing is enabled.

2-byte: 16-bit FIFO packing is enabled.

4-byte: FIFO packing is disabled.

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

## 4.56 Parameter SaiMuxMode

Mux type.

MUX\_DISABLED: Each data line is a channel, uses a separate memory block.

MUX\_LINE: Only one data line (data line 0 for DMA mode) is used. Words on data line is alternated between channels, each channel data is a separate memory block.

MUX\_MEM: Words in memory block is alternated between channels, each channel data is on a separate data line.

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	MUX_DISABLED
literals	['MUX_DISABLED', 'MUX_LINE', 'MUX_MEM']

## 4.57 Parameter SaiTransferType

Select the Sai transfer type, using Interrupt or DMA.

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	INTERRUPT
literals	['INTERRUPT', 'DMA']

## 4.58 Parameter SaiElementSize

Size in bytes of each read/write from/to buffer and should not be less than Sai Word Width.

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

Property	Value
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	true
valueConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: POST-BUILD
defaultValue	2
max	4
min	1

## 4.59 Parameter SaiBufferAccessSize

Number of read/write operations done on FIFO during one interrupt/dma operation. Maximum value is given by FIFO depth.

When FIFO packing is disabled, this value is equal with number of words each time read from/write to buffer.

If FIFO packing is enabled, the driver will actually read from/write to buffer the number multiplied by number of words packed into 1 FIFO read/write operation(example: N, with 1byte words and packing =>  $N \cdot (32/8)$  elements read from buffer per operation).

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-POST-BUILD: POST-BUILD
	VARIANT-PRE-COMPILE: PRE-COMPILE
defaultValue	4
max	8
min	1

## 4.60 Parameter SaiDataLineCount

Number of Sai Data Lines to enable, only used when both line mux mode and interrupt mode is selected.

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	2
max	4
min	2

## 4.61 Parameter SaiWordFlagIndex

Configure which word sets the start of word flag, should less than frame size.

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-POST-BUILD: POST-BUILD
	VARIANT-PRE-COMPILE: PRE-COMPILE
defaultValue	0
max	15
min	0

## 4.62 Parameter SaiBitClkInput

Use slave mode timing in datasheet for master mode.

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

Property	Value
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	false
valueConfigClasses	VARIANT-POST-BUILD: POST-BUILD
	VARIANT-PRE-COMPILE: PRE-COMPILE
default Value	false

### 4.63 Parameter SaiContOnErr

True continue transferring the same work that caused FIFO error, False start the next frame.

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
default Value	true

### 4.64 Parameter SaiSyncOnDemand

True frame sync is generated only when fifo is not empty (transmit) or not full (receive), False otherwise.

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

Property	Value
postBuildVariantValue	false
valueConfigClasses	VARIANT-POST-BUILD: PRE-COMPILE
	VARIANT-PRE-COMPILE: PRE-COMPILE
defaultValue	true

## 4.65 Reference SaiClkSrcRef

Reference to the I2sClockReferencePoint from which the clock is derived.

Property	Value
type	ECUC-REFERENCE-DEF
origin	NXP
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	true
valueConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: POST-BUILD
requiresSymbolicNameValue	False
destination	/TS_T40D2M10I1R0/I2s/I2sClockReferencePoint

## 4.66 Reference SaiDmaChannel0

Specifies channel 0 DMA channel used.

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	['/AUTOSAR/EcucDefs/Mcl/MclConfig/dmaLogicChannel_Type']

## 4.67 Reference SaiDmaChannel1

Specifies channel 1 DMA channel used.

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	['/AUTOSAR/EcucDefs/Mcl/MclConfig/dmaLogicChannel_Type']

## 4.68 Reference SaiDmaChannel2

Specifies channel 2 DMA channel used.

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	['/AUTOSAR/EcucDefs/Mcl/MclConfig/dmaLogicChannel_Type']

## 4.69 Reference SaiDmaChannel3

Specifies channel 3 DMA channel used.



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	['/AUTOSAR/EcucDefs/Mcl/MclConfig/dmaLogicChannel_Type']

## 4.70 Container FlexioHwConfiguration

Configuration of a Flexio module available on the platform.

Included subcontainers:

- None

Property	Value
type	ECUC-PARAM-CONF-CONTAINER-DEF
lowerMultiplicity	0
upperMultiplicity	Infinite
postBuildVariantMultiplicity	false
multiplicityConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: PRE-COMPILE

## 4.71 Parameter FlexioI2sInstanceId

Identifies the Flexio Channel.

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

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

## 4.72 Parameter FlexioI2sBitsWidth

Number of bits in a word - multiple of 8.

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-POST-BUILD: POST-BUILD
	VARIANT-PRE-COMPILE: PRE-COMPILE
defaultValue	8
max	32
min	1

## 4.73 Parameter FlexioI2sMode

Master or Slave.

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

Property	Value
postBuildVariantValue	true
valueConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: POST-BUILD
defaultValue	Master
literals	['Master', 'Slave']

## 4.74 Parameter FlexioI2sBaudRate

The actual baud rate in Hz that can be set by the driver. May be substantially different than the requested baud rate if the frequency of the Flexio input clock is too high or too low.

For best results it is recommended to use an input clock with a frequency 200-500 times greater than the desired baud rate

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-POST-BUILD: POST-BUILD
	VARIANT-PRE-COMPILE: PRE-COMPILE
defaultValue	20000
max	4294967295
min	1

## 4.75 Parameter FlexioI2sTransferType

Select the Flexio I2s transfer type, using Interrupt or Polling or DMA.

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

Property	Value
multiplicityConfigClasses	N/A
postBuildVariantValue	true
valueConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: POST-BUILD
defaultValue	INTERRUPT
literals	['INTERRUPT', 'POLLING', 'DMA']

## 4.76 Parameter FlexioI2sRxLine

FlexioI2sRxLine is enabled indirectly via the label node 'Enable Flexio Add pin' of Mcl component referenced from FlexioI2sTxLine.

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

## 4.77 Parameter FlexioI2sWsLine

FlexioI2sWsLine is enabled indirectly via the label node 'Enable Flexio Add pin' of Flexio\_Mcl\_Ip component referenced from FlexioI2sSckLine.

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

Property	Value
	VARIANT-PRE-COMPILE: PRE-COMPILE
default Value	true

## 4.78 Parameter FlexioI2sCallbackParam

FlexioI2s callback Param. This param will be called for all FlexioI2s events.

Property	Value
type	ECUC-FUNCTION-NAME-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
default Value	NULL_PTR

## 4.79 Reference FlexioI2sClkSrcRef

Reference to the I2sClockReferencePoint from which the clock is derived.

Property	Value
type	ECUC-REFERENCE-DEF
origin	NXP
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	true
valueConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: POST-BUILD
requiresSymbolicNameValue	False
destination	/TS_T40D2M10I1R0/I2s/I2sClockReferencePoint

## 4.80 Reference FlexioI2sTxLine

Reference to the FLEXIO logic channel, which is set in the Mcl driver configuration.

In Master mode, Channel N is referred in TX line and Channel N+1 is referred in SCK line.

In Slave mode, Channel N is referred in TX line and Channel N+1/N+2 are referred in SCK line.

Note: Implementation Specific Parameter.

Property	Value
type	ECUC-REFERENCE-DEF
origin	NXP
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	true
valueConfigClasses	VARIANT-POST-BUILD: POST-BUILD
	VARIANT-PRE-COMPILE: PRE-COMPILE
requiresSymbolicNameValue	False
destination	/TS_T40D2M10I1R0/Mcl/MclConfig/FlexioCommon/FlexioMclLogicChannels

## 4.81 Reference FlexioI2sSckLine

Reference to the FLEXIO logic channel, which is set in the Mcl driver configuration.

In Master mode, Channel N is referred in TX line and Channel N+1 is referred in SCK line.

In Slave mode, Channel N is referred in TX line and Channel N+1/N+2 are referred in SCK line.

Note: Implementation Specific Parameter.

Property	Value
type	ECUC-REFERENCE-DEF
origin	NXP
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	true
valueConfigClasses	VARIANT-POST-BUILD: POST-BUILD
	VARIANT-PRE-COMPILE: PRE-COMPILE
requiresSymbolicNameValue	False
destination	/TS_T40D2M10I1R0/Mcl/MclConfig/FlexioCommon/FlexioMclLogicChannels

## 4.82 Reference FlexioI2sDmaTxChannel

Specifies Flexio I2s Tx Dma Channel used.

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	['/AUTOSAR/EcucDefs/Mcl/MclConfig/dmaLogicChannel_Type']

## 4.83 Reference FlexioI2sDmaRxChannel

Specifies Flexio I2s Rx Dma Channel used.

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	['/AUTOSAR/EcucDefs/Mcl/MclConfig/dmaLogicChannel_Type']

## 4.84 Container CommonPublishedInformation

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

Included subcontainers:

## Tresos Configuration Plug-in

- None



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

## 4.85 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

## 4.86 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

Property	Value
	VARIANT-PRE-COMPILE: PUBLISHED-INFORMATION
defaultValue	4
max	4
min	4

## 4.87 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

## 4.88 Parameter ModuleId

Module ID of this module from Module 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	255

Property	Value
max	255
min	255

## 4.89 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	1
max	1
min	1

## 4.90 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
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

## 4.91 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	1
max	1
min	1

## 4.92 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	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	false
valueConfigClasses	VARIANT-POST-BUILD: PUBLISHED-INFORMATION
	VARIANT-PRE-COMPILE: PUBLISHED-INFORMATION
defaultValue	<b>S32K1 I2S Driver</b>

## 4.93 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

## 4.94 Container I2sClockReferencePoint

This container contains a parameter, which represents a reference to a container of the type McuClockReferencePoint (defined in module MCU).

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

## 4.95 Reference I2sClockReference

Reference to a container of the type McuClockReferencePoint, to select an input clock.

## Tresos Configuration Plug-in

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/EcuDefs/Mcu/McuModuleConfiguration/McuClockSetting↵ Config/McuClockReferencePoint

This chapter describes the Tresos configuration plug-in for the I2S Driver. The most of the parameters are described below.



# Chapter 5

## Module Index

### 5.1 Software Specification

Here is a list of all modules:

I2s Driver . . . . .	60
FLEXIO_I2S IPL . . . . .	71
I2s Driver . . . . .	72
SAI IPL . . . . .	75
SAI IPL . . . . .	76

## Chapter 6

### Module Documentation

#### 6.1 I2s Driver

##### 6.1.1 Detailed Description

##### Data Structures

- struct [I2s\\_StatusStructType](#)  
*Definition of the type of transfer status. [More...](#)*
- struct [I2s\\_ConfigType](#)  
*This type contains initialization data. [More...](#)*
- struct [I2s\\_RequestType](#)  
*Definition for Request Buffer. This is the structure which is passed to `I2s_SyncTransmit` or `I2s_AsyncTransmit` function. This holds the necessary information required for the communication of I2S Hw with the Slave device. [More...](#)*

##### Macros

- `#define I2S_E_UNINIT`  
*API service used without I2s module initialization.*
- `#define I2S_E_BUSY`  
*I2s module is busy with a running operation.*
- `#define I2S_E_TIMEOUT`  
*I2s module is timeout with a running operation.*
- `#define I2S_E_INIT_FAILED`  
*I2s module is not properly initialized.*
- `#define I2S_E_PARAM_POINTER`  
*API service is called using an invalid pointer (e.g. the pointer should not be NULL).*
- `#define I2S_E_PARAM_CHANNEL`  
*API service is called using an invalid logical channel.*
- `#define I2S_E_PARAM_CONFIG`  
*API service is called using an invalid core.*



- `#define I2S_INIT_ID`  
*API service ID for I2s\_Init function.*
- `#define I2S_DEINIT_ID`  
*API service ID for I2s\_DeInit function.*
- `#define I2S_SYNCTRANSMIT_ID`  
*API service ID for I2s\_SyncTransmit function.*
- `#define I2S_ASYNCTRANSMIT_ID`  
*API service ID for I2s\_AsyncTransmit function.*
- `#define I2S_GETSTATUS_ID`  
*API service ID for I2s\_GetStatus function.*
- `#define I2S_ABORTTRANSMIT_ID`  
*API service ID for I2s\_AbortTransmit function.*
- `#define I2S_GETVERSIONINFO_ID`  
*API service ID for I2s\_GetVersionInfo function.*

## Types Reference

- `typedef uint8 I2s_HwUnitCount`  
*This gives the number of I2S configuration.*
- `typedef uint8 I2s_DataType`  
*This gives the transfer data type.*
- `typedef uint16 I2s_WordMaskType`  
*I2s Word Mask for transmitted data.*
- `typedef uint8 I2s_LogicalChannel`  
*This gives the logical channel.*

## Enum Reference

- `enum I2s_DataDirectionType`  
*Definition of the type of transfer direction.*

## Function Reference

- `void I2s_Init (const I2s_ConfigType *const pConfigPtr)`  
*Initializes the I2S module.*
- `void I2s_DeInit (void)`  
*DeInitializes the I2S module.*
- `Std_ReturnType I2s_SyncTransmit (I2s_LogicalChannel LogicChn, const I2s_RequestType *const pRequest)`  
*Sends or receives an I2S message blocking.*
- `Std_ReturnType I2s_AsyncTransmit (I2s_LogicalChannel LogicChn, const I2s_RequestType *const pRequest)`  
*Starts an asynchronous transmission on the I2S bus.*
- `Std_ReturnType I2s_GetStatus (I2s_LogicalChannel LogicChn, I2s_StatusStructType *const pStatus)`  
*Gets the status of an I2S instance.*
- `void I2s_AbortTransmit (I2s_LogicalChannel LogicChn)`  
*Aborts the I2s ongoing transmission.*

## 6.1.2 Data Structure Documentation

### 6.1.2.1 struct I2s\_\_StatusStructType

Definition of the type of transfer status.

Definition at line 147 of file CDD\_I2s\_Types.h.

### 6.1.2.2 struct I2s\_\_ConfigType

This type contains initialization data.

This contains initialization data for the I2S driver. It shall contain:

- The number of I2S modules to be configured
- I2S dependent properties for used HW units

Definition at line 168 of file CDD\_I2s\_Types.h.

Data Fields

Type	Name	Description
const I2s__HwUnitCount	cfgCount	Number of I2S configurations. < Pointer to I2s hardware unit configuration array
const I2s__HwConfigType *const *	pHwConfig	

### 6.1.2.3 struct I2s\_\_RequestType

Definition for Request Buffer. This is the structure which is passed to I2s\_\_SyncTransmit or I2s\_\_AsyncTransmit function. This holds the necessary information required for the communication of I2S Hw with the Slave device.

Definition at line 182 of file CDD\_I2s\_Types.h.

Data Fields

Type	Name	Description
uint32	bufferSize	Buffer Size : The number of words for reading or writing of each channel. The size of each buffer word element is "I2s Buffer Element Size" selected in configurator. < Direction of the data. Can be either Send or Receive.
I2s__DataDirectionType	dataDirection	
I2s__DataType **	pDataBuffer	Buffer to Store or to transmit Serial data. <

### 6.1.3 Macro Definition Documentation

#### 6.1.3.1 I2S\_E\_UNINIT

```
#define I2S_E_UNINIT
```

API service used without I2s module initialization.

Development errors. The following errors shall be detectable by the I2S module depending on its configuration (development / production mode).

All error codes

Definition at line 136 of file CDD\_I2s.h.

#### 6.1.3.2 I2S\_E\_BUSY

```
#define I2S_E_BUSY
```

I2s module is busy with a running operation.

Definition at line 141 of file CDD\_I2s.h.

#### 6.1.3.3 I2S\_E\_TIMEOUT

```
#define I2S_E_TIMEOUT
```

I2s module is timeout with a running operation.

Definition at line 146 of file CDD\_I2s.h.

#### 6.1.3.4 I2S\_E\_INIT\_FAILED

```
#define I2S_E_INIT_FAILED
```

I2s module is not properly initialized.

Definition at line 151 of file CDD\_I2s.h.

### 6.1.3.5 I2S\_E\_PARAM\_POINTER

```
#define I2S_E_PARAM_POINTER
```

API service is called using an invalid pointer (e.g. the pointer should not be NULL).

Definition at line 156 of file CDD\_I2s.h.

### 6.1.3.6 I2S\_E\_PARAM\_CHANNEL

```
#define I2S_E_PARAM_CHANNEL
```

API service is called using an invalid logical channel.

Definition at line 161 of file CDD\_I2s.h.

### 6.1.3.7 I2S\_E\_PARAM\_CONFIG

```
#define I2S_E_PARAM_CONFIG
```

API service is called using an invalid core.

Definition at line 166 of file CDD\_I2s.h.

### 6.1.3.8 I2S\_INIT\_ID

```
#define I2S_INIT_ID
```

API service ID for I2s\_Init function.

All AUTOSAR API's service IDs

Definition at line 175 of file CDD\_I2s.h.

### 6.1.3.9 I2S\_DEINIT\_ID

```
#define I2S_DEINIT_ID
```

API service ID for I2s\_DeInit function.

Definition at line 180 of file CDD\_I2s.h.

#### 6.1.3.10 I2S\_SYNCTRANSMIT\_ID

```
#define I2S_SYNCTRANSMIT_ID
```

API service ID for I2s\_SyncTransmit function.

Definition at line 185 of file CDD\_I2s.h.

#### 6.1.3.11 I2S\_ASYNCTRANSMIT\_ID

```
#define I2S_ASYNCTRANSMIT_ID
```

API service ID for I2s\_AsyncTransmit function.

Definition at line 190 of file CDD\_I2s.h.

#### 6.1.3.12 I2S\_GETSTATUS\_ID

```
#define I2S_GETSTATUS_ID
```

API service ID for I2s\_GetStatus function.

Definition at line 195 of file CDD\_I2s.h.

#### 6.1.3.13 I2S\_ABORTTRANSMIT\_ID

```
#define I2S_ABORTTRANSMIT_ID
```

API service ID for I2s\_AbortTransmit function.

Definition at line 200 of file CDD\_I2s.h.

#### 6.1.3.14 I2S\_GETVERSIONINFO\_ID

```
#define I2S_GETVERSIONINFO_ID
```

API service ID for I2s\_GetVersionInfo function.

Definition at line 205 of file CDD\_I2s.h.

### 6.1.4 Types Reference

#### 6.1.4.1 I2s\_\_HwUnitCount

```
typedef uint8 I2s__HwUnitCount
```

This gives the number of I2S configuration.

Definition at line 127 of file CDD\_I2s\_Types.h.

#### 6.1.4.2 I2s\_\_DataType

```
typedef uint8 I2s__DataType
```

This gives the transfer data type.

Definition at line 132 of file CDD\_I2s\_Types.h.

#### 6.1.4.3 I2s\_\_WordMaskType

```
typedef uint16 I2s__WordMaskType
```

I2s Word Mask for transmitted data.

Definition at line 137 of file CDD\_I2s\_Types.h.

#### 6.1.4.4 I2s\_\_LogicalChannel

```
typedef uint8 I2s__LogicalChannel
```

This gives the logical channel.

Definition at line 142 of file CDD\_I2s\_Types.h.

### 6.1.5 Enum Reference

#### 6.1.5.1 I2s\_\_DataDirectionType

```
enum I2s__DataDirectionType
```

Definition of the type of transfer direction.

Enumerator

I2S_SEND_DATA	Used to send data.
I2S_RECEIVE_DATA	Used to receive data.

Definition at line 156 of file CDD\_I2s\_Types.h.

## 6.1.6 Function Reference

### 6.1.6.1 I2s\_Init()

```
void I2s_Init (
    const I2s_ConfigType *const pConfigPtr )
```

Initializes the I2S module.

This function performs software initialization of I2S driver.

Parameters

in	<i>pConfig</i>	Pointer to I2S driver configuration set.
----	----------------	--

Returns

void

Note

Service ID: 0x00.

Synchronous, non re-entrant function.

### 6.1.6.2 I2s\_DeInit()

```
void I2s_DeInit (
    void )
```

DeInitializes the I2S module.

This function performs software de initialization of I2S 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 [I2s\\_Init\(\)](#). The driver needs to be initialized before calling [I2s\\_DeInit\(\)](#). Otherwise, the function [I2s\\_DeInit](#) shall raise the development error I2S\_E\_UNINIT and leave the desired de initialization functionality without any action.

Module Documentation

Parameters

in	<i>void</i>	
----	-------------	--

Returns

void

Note

- Service ID: 0x01.
- Synchronous, non re-entrant function.

6.1.6.3 I2s\_SyncTransmit()

```
Std_ReturnType I2s_SyncTransmit (
    I2s_LogicalChannel LogicChn,
    const I2s_RequestType *const pRequest )
```

Sends or receives an I2S message blocking.

Sends the slave address and based on the direction of the message it sends or receives data by using a blocking mechanism.

Parameters

in	<i>LogicChn</i>	I2S logical channel to be addressed.
in	<i>pRequest</i>	Pointer to data information to be used

Returns

Std\_ReturnType.

Return values

<i>E_NOT_OK</i>	If the I2S Instance is not valid or I2S driver is not initialized or pRequest is NULL or I2S Instance is in busy state.
<i>E_OK</i>	Otherwise.

Note

- Service ID: 0x02.
- Synchronous, non reentrant function.



#### 6.1.6.4 I2s\_AsyncTransmit()

```
Std_ReturnType I2s_AsyncTransmit (
    I2s_LogicalChannel LogicChn,
    const I2s_RequestType *const pRequest )
```

Starts an asynchronous transmission on the I2S bus.

Sends the slave address and enables the interrupts that will send or receive data depending on the direction of the message.

Parameters

in	<i>LogicChn</i>	I2S logical channel to be addressed.
in	<i>pRequest</i>	Pointer to data information to be used

Returns

Std\_ReturnType.

Return values

<i>E_NOT_OK</i>	If the I2S Instance is not valid or I2S driver is not initialized or pRequest is NULL or I2S Instance is in busy state.
<i>E_OK</i>	Otherwise.

Note

Service ID: 0x03.

Synchronous, non reentrant function.

#### 6.1.6.5 I2s\_GetStatus()

```
Std_ReturnType I2s_GetStatus (
    I2s_LogicalChannel LogicChn,
    I2s_StatusStructType *const pStatus )
```

Gets the status of an I2S instance.

Gets the status of an I2S instance and checks for errors.

## Module Documentation

### Parameters

in	<i>LogicChn</i>	I2S logical channel to be addressed.
out	<i>pStatus</i>	Pointer for storing the current transfer status

### Returns

Std\_ReturnType.

### Return values

<i>E_NOT_OK</i>	If the I2S Instance is not valid or I2S driver is not initialized or pStatus is NULL.
<i>E_OK</i>	Otherwise.

### Note

Service ID: 0x04.

Synchronous, non re-entrant function.

#### 6.1.6.6 I2s\_AbortTransmit()

```
void I2s_AbortTransmit (
    I2s_LogicalChannel LogicChn )
```

Aborts the I2s ongoing transmission.

Aborts the I2s ongoing transmission.

### Parameters

in	<i>LogicChn</i>	I2S logical channel to be addressed.
----	-----------------	--------------------------------------

### Returns

void.

### Note

Service ID: 0x05.

Synchronous, non reentrant function.

## 6.2 FLEXIO\_I2S IPL

### 6.2.1 Detailed Description

## 6.3 I2s Driver

### 6.3.1 Detailed Description

#### Data Structures

- struct [I2s\\_Ipw\\_ConfigType](#)  
*This structure contains config structures for all IPs available. [More...](#)*
- struct [I2s\\_HwConfigType](#)  
*This structure holds the HWUnit configuration parameters. [More...](#)*
- struct [I2s\\_LogicalConfigState](#)  
*This structure holds the Logical configuration parameters. [More...](#)*

#### Enum Reference

- enum [I2s\\_StatusType](#)  
*Definition for different state and errors of Operation Status.*
- enum [I2s\\_TransmitReceiveType](#)  
*Definition of the transmitter/receiver of a SAI hw unit.*
- enum [I2s\\_ModeType](#)  
*Definition of the master/slave mode.*
- enum [I2s\\_SupportedIPsType](#)  
*This enum contains all IPs which can integrate SAI functionalities.*

### 6.3.2 Data Structure Documentation

#### 6.3.2.1 struct I2s\_Ipw\_ConfigType

This structure contains config structures for all IPs available.

Definition at line 194 of file `I2s_Ipw_Types.h`.

#### 6.3.2.2 struct I2s\_HwConfigType

This structure holds the HWUnit configuration parameters.

Definition at line 209 of file `I2s_Ipw_Types.h`.

#### 6.3.2.3 struct I2s\_LogicalConfigState

This structure holds the Logical configuration parameters.

Definition at line 224 of file `I2s_Ipw_Types.h`.

### 6.3.3 Enum Reference

#### 6.3.3.1 I2s\_StatusType

enum [I2s\\_StatusType](#)

Definition for different state and errors of Operation Status.

Enumerator

I2S_STATUS_UNINITIALIZED	Function unsupported
I2S_STATUS_COMPLETED	Function completed successfully
I2S_STATUS_ERROR	Function didn't complete successfully
I2S_STATUS_TIMEOUT	Function timed out
I2S_STATUS_BUSY	Function busy
I2S_STATUS_ABORTED	Function aborted

Definition at line 150 of file I2s\_Ipw\_Types.h.

### 6.3.3.2 I2s\_TransmitReceiveType

```
enum I2s_TransmitReceiveType
```

Definition of the transmitter/receiver of a SAI hw unit.

Enumerator

I2S_TRANSMITTER	Transmitter.
I2S_RECEIVER	Receiver.

Definition at line 163 of file I2s\_Ipw\_Types.h.

### 6.3.3.3 I2s\_ModeType

```
enum I2s_ModeType
```

Definition of the master/slave mode.

Enumerator

I2S_MASTER	Master.
I2S_SLAVE	Slave.

Definition at line 172 of file I2s\_Ipw\_Types.h.

### 6.3.3.4 I2s\_SupportedIPsType

enum `I2s_SupportedIPsType`

This enum contains all IPs which can integrate SAI functionalities.

Definition at line 181 of file `I2s_Ipw_Types.h`.

## 6.4 SAI IPL

### 6.4.1 Detailed Description

## 6.5 SAI IPL

### 6.5.1 Detailed Description

#### Macros

- `#define SAI_IP_FPACK_8_BIT`
- `#define SAI_IP_FPACK_16_BIT`
- `#define FEATURE_SAI_IP_HAS_CHMOD`  
*Channel mode tristate or output zero.*
- `#define FEATURE_SAI_IP_MSEL_BUS_CLK`  
*Master clock from Bus clock.*
- `#define FEATURE_SAI_IP_MSEL_SOSC_DIV1_CLK`  
*Master clock from SOSC\_DIV1 clock.*
- `#define FEATURE_SAI_IP_MSEL_MCLK_PIN`  
*Master clock from MCLK pin.*
- `#define FEATURE_SAI_IP_MSEL_OTHER_MCLK_PIN`  
*Master clock from MCLK pin from other SAI instance.*
- `#define FEATURE_SAI_IP_HAS_PARAM`  
*SAI has PARAM register.*
- `#define SAI0_IP_CHANNEL_COUNT`  
*SAI channel count.*
- `#define SAI_IP_NULL_PTRS`  
*SAI NULL PTRs.*

### 6.5.2 Macro Definition Documentation

#### 6.5.2.1 SAI\_IP\_FPACK\_8\_BIT

```
#define SAI_IP_FPACK_8_BIT
```

8-bit FIFO packing is enabled

Definition at line 81 of file Sai\_Ip\_FeatureDefines.h.

#### 6.5.2.2 SAI\_IP\_FPACK\_16\_BIT

```
#define SAI_IP_FPACK_16_BIT
```

16-bit FIFO packing is enabled

Definition at line 82 of file Sai\_Ip\_FeatureDefines.h.



### 6.5.2.3 FEATURE\_SAI\_IP\_HAS\_CHMOD

```
#define FEATURE_SAI_IP_HAS_CHMOD
```

Channel mode tristate or output zero.

Definition at line 84 of file Sai\_Ip\_FeatureDefines.h.

### 6.5.2.4 FEATURE\_SAI\_IP\_MSEL\_BUS\_CLK

```
#define FEATURE_SAI_IP_MSEL_BUS_CLK
```

Master clock from Bus clock.

Definition at line 86 of file Sai\_Ip\_FeatureDefines.h.

### 6.5.2.5 FEATURE\_SAI\_IP\_MSEL\_SOSC\_DIV1\_CLK

```
#define FEATURE_SAI_IP_MSEL_SOSC_DIV1_CLK
```

Master clock from SOSC\_DIV1 clock.

Definition at line 88 of file Sai\_Ip\_FeatureDefines.h.

### 6.5.2.6 FEATURE\_SAI\_IP\_MSEL\_MCLK\_PIN

```
#define FEATURE_SAI_IP_MSEL_MCLK_PIN
```

Master clock from MCLK pin.

Definition at line 90 of file Sai\_Ip\_FeatureDefines.h.

### 6.5.2.7 FEATURE\_SAI\_IP\_MSEL\_OTHER\_MCLK\_PIN

```
#define FEATURE_SAI_IP_MSEL_OTHER_MCLK_PIN
```

Master clock from MCLK pin from other SAI instance.

Definition at line 92 of file Sai\_Ip\_FeatureDefines.h.

### 6.5.2.8 FEATURE\_SAI\_IP\_HAS\_PARAM

```
#define FEATURE_SAI_IP_HAS_PARAM
```

SAI has PARAM register.

Definition at line 94 of file Sai\_Ip\_FeatureDefines.h.

### 6.5.2.9 SAI0\_IP\_CHANNEL\_COUNT

```
#define SAI0_IP_CHANNEL_COUNT
```

SAI channel count.

Definition at line 96 of file Sai\_Ip\_FeatureDefines.h.

### 6.5.2.10 SAI\_IP\_NULL\_PTRS

```
#define SAI_IP_NULL_PTRS
```

SAI NULL PTRs.

Definition at line 102 of file Sai\_Ip\_FeatureDefines.h.

**How to Reach Us:**

**Home Page:**

[nxp.com](http://nxp.com)

**Web Support:**

[nxp.com/support](http://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](http://nxp.com/SalesTermsandConditions).

NXP, the NXP logo, NXP SECURE CONNECTIONS FOR A SMARTER WORLD, COOLFLUX, EMBRACE, GREENCHIP, HITAG, I2C BUS, ICODE, JCOP, LIFE VIBES, MIFARE, MIFARE CLASSIC, MIFARE DESFire, MIFARE PLUS, MIFARE FLEX, MANTIS, MIFARE ULTRALIGHT, MIFARE4MOBILE, MIGLO, NTAG, ROADLINK, SMARTLX, SMARTMX, STARPLUG, TOPFET, TRENCHMOS, UCODE, Freescale, the Freescale logo, Altivec, C-5, CodeTEST, CodeWarrior, ColdFire, ColdFire+, C-Ware, the Energy Efficient Solutions logo, Kinetis, Layerscape, MagniV, mobileGT, PEG, PowerQUICC, Processor Expert, QorIQ, QorIQ Qonverge, Ready Play, SafeAssure, the SafeAssure logo, StarCore, Symphony, VortiQa, Vybrid, Airfast, BeeKit, BeeStack, CoreNet, Flexis, MXC, Platform in a Package, QUICC Engine, SMARTMOS, Tower, TurboLink, and UMEMS are trademarks of NXP B.V. All other product or service names are the property of their respective owners. ARM, AMBA, ARM Powered, Artisan, Cortex, Jazelle, Keil, SecurCore, Thumb, TrustZone, and Vision are registered trademarks of ARM Limited (or its subsidiaries) in the EU and/or elsewhere. ARM7, ARM9, ARM11, big.LITTLE, CoreLink, CoreSight, DesignStart, Mali, mbed, NEON, POP, Sensinode, Socrates, ULINK and Versatile are trademarks of ARM Limited (or its subsidiaries) in the EU and/or elsewhere. All rights reserved. Oracle and Java are registered trademarks of Oracle and/or its affiliates. The Power Architecture and Power.org word marks and the Power and Power.org logos and related marks are trademarks and service marks licensed by Power.org.

© 2022 NXP B.V.

