User Manual

for S32K14X ICU Driver

Document Number: UM2ICUASR4.2 Rev0002R1.0.2
Rev. 1.0

Contents

Section number **Title** **Page**

Chapter 1
Revision History

Chapter 2
Introduction

Chapter 3
Driver

Section number Title Page

Section number	Title	Page

Chapter 4
Tresos Configuration Plug-in

Chapter 1
Revision History

Table 1-1. Revision History

Revision	Date	Author	Description
1.0	26/04/2019	NXP MCAL Team	Updated version for ASR 4.2.2S32K14XR1.0.2

Chapter 2
Introduction

This User Manual describes NXP Semiconductors AUTOSAR Input Capture Unit (ICU) for S32K14X .

AUTOSAR ICU driver configuration parameters and deviations from the specification are described in ICU Driver chapter of this document. AUTOSAR ICU driver requirements and APIs are described in the AUTOSAR ICU driver software specification document.

2.1 Supported Derivatives

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

Table 2-1. S32K14X Derivatives

NXP Semiconductors	s32k148_lqfp144, s32k148_lqfp176, s32k148_mapbga100, s32k146_lqfp144, s32k146_lqfp100, s32k146_lqfp64, s32k146_mapbga100, s32k144_lqfp100, s32k144_lqfp64, s32k144_mapbga100, s32k142_lqfp100, s32k142_lqfp64, s32k118_lqfp48, s32k118_lqfp64, s32k142_lqfp48, s32k144_lqfp48, s32k148_lqfp100

All of the above microcontroller devices are collectively named as S32K14X .

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 type: Bold is used for important terms, notes and warnings.

Italic font: Italic typeface is 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.

2.4 Acronyms and Definitions

Table 2-2. Acronyms and Definitions

Term	Definition
BSW	Basic Software
DEM	Diagnostic Event Manager
DET	Development Error Tracer
ECU	Electronic Control Unit
ICU	Input Capture Unit
ISR	interrupt Service Routine
OS	Operating System
RAM	Random Access Memory
ROM	Read-only Memory

Table continues on the next page...

Table 2-2. Acronyms and Definitions (continued)

Term	Definition
MCU	Microcontroller Unit
GUI	Graphical User Interface
EcuM	ECU state Manager
FTM	FlexTimer Module
PORT_CI	Port Control and Interrupts
LPTMR	Low-Power Timer
LPIT	Low Power Interrupt Timer
API	Application Programming Interface
PB Variant	Post Build Variant
PC Variant	Pre Compile Variant

2.5 Reference List

Table 2-3. Reference List

#	Title	Version
1	Specification of ICU Driver	AUTOSAR Release 4.2.2
2	S32K14X Reference Manual	Reference Manual, Rev. 9, 9/2018
3	S32K142 Mask Set Errata for Mask 0N33V (0N33V)	30/11/2017
4	S32K144 Mask Set Errata for Mask 0N57U (0N57U)	30/11/2017
5	S32K146 Mask Set Errata for Mask 0N73V (0N73V)	30/11/2017
6	S32K148 Mask Set Errata for Mask 0N20V (0N20V)	25/10/2018
7	S32K118 Mask Set Errata for Mask 0N97V (0N97V)	07/01/2019

Chapter 3
Driver

3.1 Requirements

Requirements for this driver are detailed in the AUTOSAR 4.2 Rev0002ICU Driver Software Specification document (See Table Reference List).

3.2 Driver Design Summary

The ICU Driver controls the input capture of the microcontroller. It provides the following features:

- High time / Low time measurement
- Duty Cycle measurement
- Period time measurement
- Edge detection and notification
- Edge counting (with or without hardware gating)
- Edge time stamping

For signal edge detection, the edge detector of a capture compare unit or the interrupt controller for external events are used.

For signal measuring a capture timer and at least one capture register are needed. Also, only even channels (2*n) (with n=0,1,2...) can be used for signal measurements. This is because the channel after it (2*n+1) is used internally by the ICU Driver.

The FlexTimer module of S32K14X supports period time measurement, edge detection and notification, edge counting and edge time stamping.

The PORT_CI module of S32K14X supports edge detection and notification.

The LPTMR module of S32K14X supports edge detection and notification, edge counting.

The LPIT module of S32K14X can support edge detection and notification and edge time stamping.

For details please refer to **Table 1**.

The ICU driver provides an optional API and configuration parameters for changing the base clock of the controlled hardware. A dual clock functionality is offered by switching between two configured values of the clock prescaler.

The ICU driver alos provides an optional API for reading the level of an input pin for FlexTimer channels .

For each user configured channel, a symbolic name is generated by the Tresos Studio configuration tool. The name shall be consequently used in upper applications (e.g. ICU_FTM_0_CH_1).

By default all channels offer interrupt handlers. For each channel not configured by the user in Tresos Studio configuration tool, the code for interrupt handling is removed based on a series of ifdefs.

3.3 Hardware Resources

The hardware configured by the Icu driver are FTM, LPIT, LPTMR and PORT_CI.

The FTM, LPIT, LPTMR and PORT_CI input signal to microcontroller pin mapping can be done by using chapter 2.2 "External Signal Description" from the Reference manual.

3.4 Deviation from requirement

Deviations from the AUTOSAR ICU Driver software specification are listed in table **Deviations Status Column Description**.

Table 3-1. Deviations Status Column Description

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 or out of scope for the ICU driver.

Table 3-2. ICU Deviations Table

Requirement	Status	Description	Notes
SWS_Icu_00131	N/S	The ICU driver shall describe which other modules (in which versions) are required. This description shall be done by the implementer.	Not an ICU driver requirement - this is a documentatio guideline
SWS_Icu_00116	N/I	The module shall optionally include the Dem.h file if any production error will be issued by the implementation. By this inclusion, the API's to report errors as well as the required Event Id symbols are included.This specification defines the name of the Event Id symbols, which are provided by XML to the DEM configuration tool. The DEM configuration tool assigns ECU dependent values to the Event Id symbols and publishes the symbols in Dem_IntErrId.h.	The production errors will be reported via SERR
SWS_Icu_00380	N/I	These requirements are not applicable to this specification. (SRS_BSW_00300, SRS_BSW_00301, SRS_BSW_0 ...	No requirement of type 'Replaces' was found
SWS_Icu_00149	N/S	The Icu module's environment shall check the integrity if several calls for the same ICU channel are used during runtime in different tasks or ISRs.	The SWS_Icu_00149 is a safety integrity assumption for external environment, which shall be implemented for FTE; For GTE and NTE ICU149 has a role to increase availablity because the check will be supported by ICU driver;
SWS_Icu_00150	N/I	The Icu module shall not check the integrity if several calls for the same ICU channel are used during runtime in different tasks or ISRs.	The SWS_Icu_00149 is a safety integrity assumption for external environment, which shall be implemented for FTE; For GTE and NTE ICU149 has a role to increase availablity because the check will be supported by ICU driver;
SWS_Icu_00152	N/S	The Icu module's environment shall not call Icu_DeInit during a running operation (e. g. timestamp measurement or edge counting)	
SWS_Icu_00190	N/S	Dem_EventIdType shall be imported from Dem_Types.h.	Serr.h shall include the Dem.h. All production errors are reported via Serr
SWS_Icu_00221	N/I	A re-initialization of the ICU module by executing the Icu_Init() function requires a de-initialization before by executing the Icu_DeInit() function	
SWS_Icu_00246	N/S	Icu_Irq.c shall include Icu.h for the function which shall be called in the interrupt function.and Icu_Irq.h for the declaration of interrupt functions.	Rejection reason: Icu_Irq.c is not needed, the interrupts are handled at IP level
SWS_Icu_00248	N/S	Icu_Lcfg.c shall include Icu_Cbk.h for a link time configuration if the call back function is linked to the module via the ROM structure.	Rejection reason: Icu does not support link time. Icu has only precompile and post build variant (see ICU188 and ICU189)

Table continues on the next page...

Table 3-2. ICU Deviations Table (continued)

Requirement	Status	Description	Notes
SWS_Icu_00249	N/S	Icu_PBcfg.c shall include Icu_Cbk.h for post build time configuration if the call back function is linked to the module via the ROM structure.	Rejection reason: Since the ICU is a driver module, it doesn't provide any callback functions for lower layer modules. (see chapter 8.4)
SWS_Icu_00250	N/S	Icu.c shall include Icu_Cbk.h for pre-compile time configuration	
SWS_Icu_00252	N/S	Icu_Irq.c shall include MemMap.h.	Icu_Irq.c file does not exist in sICU architecture
SWS_Icu_00253	N/S	Icu_Lcfg.c shall include [ICU.h and MemMap.h.	Rejection reason: Icu does not support link time. Icu has only precompile and post build variant (see ICU188 and ICU189)
SWS_Icu_00276	N/S	EcuM_WakeupSourceType shall be imported from EcuM_Types.h Module|Imported Type - Dem|Dem_EventIdType| Dem_EventStatusType|EcuM| EcuM_WakeupSourceType|Std_Types| Std_ReturnType| Std_VersionInfoType	
SWS_Icu_00348	N/S	Re-entrancy of operation Icu_SignalNotification_Channel is not relevant for this module (In general it is in this case not re-entrant).	Serr.h shall include the Dem.h. All production errors are reported via Serr
SWS_Icu_00349	N/S	Re-entrancy of the Icu_TimestampNotification_Channel is not relevant for this module (in general it is in this case not re-entrant).	
ICU353	N/S	Variables available for debugging shall be described in the respective Basic Software Module description.	Rejection Reason: Part6 -Table8- 1e: Avoid global variables or else justify their usage
SWS_Icu_00361	N/S	The ICU module's environment shall only use the re-entrant capability of the function Icu_CheckWakeup if the ICU module's environment takes care that there is no simultaneous usage of the same channel.	The wakeup functionality is not considered to take part in a safety functionality
SWS_Icu_00380	N/S	These requirements are not applicable to this specification.(BSW00300, BSW00301, BSW00302, BSW00304, BSW00305, BSW00306, BSW00307, BSW00308, BSW00309, BSW00310....	Not a requirement

3.5 Driver limitations

None

3.6 Driver usage and configuration tips

In this chapter, the extra features from our drivers that are not described in the AutoSAR standard are detailed

3.6.1 Icu with DMA feature

In order to speed up data transfer, the Direct Memory Access feature can be used.

The DMA feature can be used only in Timestamp mode and is done with the help of the Mcl driver to transfer timestamp data from a Icu input match dirrectly in the Timestamp buffer. Thus, this is an example of Peripheral to Memory data transfer and it's very usefull for avoiding the interrupt overhead. In order to have this feature, the user should check the IcuDMAChannelEnable checkbox for DMA in the Icu channel. This will be selectable only for a channel in Timestamp mode. Also, the user has to configure a Mcl channel with a Mcl Dma Transfer Completion User Notification named <IcuChannelName>_MclDmaTransferCompletionNotif for the respective Icu Channel.

for example, for Icu Channel 4, the notification should be named IcuChannel_4_MclDmaTransferCompletionNotif.

Apart from that, the user should configure the DMA instance that allows transfer from that ICU modules (in this case, FTM). The DMA channel mapping from the RM shows the sources mapping for each peripheral in the DMA channel mapping chappter. This can also be observed in the Mcl plugin in the list of DMA sources.

Other fields from the DMA's TCD do not require explicit configuration since they are specific to the ICU's FTM module.

3.6.2 Dual Clock Feature

In order to allow dinamic change of the driver working frequency, the driver has the Dual Clock Feature

The IcuEnableDualClockMode from IcuNonAUTOSAR should be enabled in order to have this feature active. Afterwards, the ICU FlexTimer Alternate Prescaler parameter allows setting a different prescaler for each module. These parameters will be changed when calling the function call Icu_SetClockMode.

Icu_SetClockMode may be called only after Icu_Init is called and when IcuEnableDualClockMode is checked. Our suggested ussage of this API is to call it when the driver is in a lower power state but still in active use.

3.6.3 Icu Lptmr Standby Wakeup Support

In case of a Standby wakeup through platform reset with Lptmr IP register content keep

The feature is a non Autosar support for wakeup through reset. This feature does not respect standard specifications and requires interrupts to be disable while calling driver Init API

Init API will not erase the interrupt flag in Lptmr and will allow execution of interrupt once the service is enabled for the platform.

3.7 Runtime Errors

The driver does not generate any DEM runtime errors.

3.8 Software specification

3.8.1 Define Reference

Constants supported by the driver are as per AUTOSAR ICU Driver software specification Version 4.2 Rev0002 .

3.8.1.1 Define ICU_VENDOR_ID

<u>**Implements**</u>**:** Icu_interface

Table 3-3. Define ICU_VENDOR_ID Description

Name	ICU_VENDOR_ID
Initializer	43

3.8.1.2 Define ICU_AR_RELEASE_MAJOR_VERSION

Implements: Icu_interface

Table 3-4. Define ICU_AR_RELEASE_MAJOR_VERSION Description

Name	ICU_AR_RELEASE_MAJOR_VERSION
Initializer	4

3.8.1.3 Define ICU_AR_RELEASE_MINOR_VERSION

Implements: Icu_interface

Table 3-5. Define ICU_AR_RELEASE_MINOR_VERSION Description

Name	ICU_AR_RELEASE_MINOR_VERSION
Initializer	2

3.8.1.4 Define ICU_AR_RELEASE_REVISION_VERSION

Implements: Icu_interface

Table 3-6. Define ICU_AR_RELEASE_REVISION_VERSION Description

Name	ICU_AR_RELEASE_REVISION_VERSION
Initializer	2

3.8.1.5 Define ICU_SW_MAJOR_VERSION

Implements: Icu_interface

Table 3-7. Define ICU_SW_MAJOR_VERSION Description

Name	ICU_SW_MAJOR_VERSION

Table continues on the next page...

Table 3-7. Define ICU_SW_MAJOR_VERSION Description (continued)

Initializer	1

3.8.1.6 Define ICU_SW_MINOR_VERSION

Implements: Icu_interface

Table 3-8. Define ICU_SW_MINOR_VERSION Description

Name	ICU_SW_MINOR_VERSION
Initializer	0

3.8.1.7 Define ICU_SW_PATCH_VERSION

Implements: Icu_interface

Table 3-9. Define ICU_SW_PATCH_VERSION Description

Name	ICU_SW_PATCH_VERSION
Initializer	2

3.8.1.8 Define ICU_CHECKWAKEUP_ID

API service ID for Icu_CheckWakeup function.

Details:

Parameters used when raising an error/exception

Table 3-10. Define ICU_CHECKWAKEUP_ID Description

Name	ICU_CHECKWAKEUP_ID
Initializer	(uint8)0x15

3.8.1.9 Define ICU_DEINIT_ID

API service ID for Icu_DeInit function.

Details:

Parameters used when raising an error/exception

Table 3-11. Define ICU_DEINIT_ID Description

Name	ICU_DEINIT_ID
Initializer	(uint8)0x01

3.8.1.10 Define ICU_DISABLEEDGECOUNT_ID

API service ID for Icu_DisableEdgeCount function.

Details:

Parameters used when raising an error/exception

Table 3-12. Define ICU_DISABLEEDGECOUNT_ID Description

Name	ICU_DISABLEEDGECOUNT_ID
Initializer	(uint8)0x0E

3.8.1.11 Define ICU_DISABLEEDGEDETECTION_ID

API service ID for Icu_DisableEdgeDetection function.

Details:

Parameters used when raising an error/exception

Table 3-13. Define ICU_DISABLEEDGEDETECTION_ID Description

Name	ICU_DISABLEEDGEDETECTION_ID
Initializer	(uint8)0x17

3.8.1.12 Define ICU_DISABLENOTIFICATION_ID

API service ID for Icu_DisableNotification function.

Details:

Parameters used when raising an error/exception

Table 3-14. Define ICU_DISABLENOTIFICATION_ID Description

Name	ICU_DISABLENOTIFICATION_ID
Initializer	(uint8)0x06

3.8.1.13 Define ICU_DISABLEWAKEUP_ID

API service ID for Icu_DisableWakeup function.

Details:

Parameters used when raising an error/exception

Table 3-15. Define ICU_DISABLEWAKEUP_ID Description

Name	ICU_DISABLEWAKEUP_ID
Initializer	(uint8)0x03

3.8.1.14 Define ICU_ENABLEEDGECOUNT_ID

API service ID for Icu_EnableEdgeCount function.

Details:

Parameters used when raising an error/exception

Table 3-16. Define ICU_ENABLEEDGECOUNT_ID Description

Name	ICU_ENABLEEDGECOUNT_ID
Initializer	(uint8)0x0D

3.8.1.15 Define ICU_ENABLEEDGEDETECTION_ID

API service ID for Icu_EnableEdgeDetection function.

Details:

Parameters used when raising an error/exception

Table 3-17. Define ICU_ENABLEEDGEDETECTION_ID Description

Name	ICU_ENABLEEDGEDETECTION_ID
Initializer	(uint8)0x16

3.8.1.16 Define ICU_ENABLENOTIFICATION_ID

API service ID for Icu_EnableNotification function.

Details:

Parameters used when raising an error/exception

Table 3-18. Define ICU_ENABLENOTIFICATION_ID Description

Name	ICU_ENABLENOTIFICATION_ID
Initializer	(uint8)0x07

3.8.1.17 Define ICU_ENABLEWAKEUP_ID

API service ID for Icu_EnableWakeup function.

Details:

Parameters used when raising an error/exception

Table 3-19. Define ICU_ENABLEWAKEUP_ID Description

Name	ICU_ENABLEWAKEUP_ID
Initializer	(uint8)0x04

3.8.1.18 Define ICU_GET_INPUT_LEVEL_ID

API service ID for Icu_GetInputLevel function.

Details:

Parameters used when raising an error/exception

Table 3-20. Define ICU_GET_INPUT_LEVEL_ID Description

Name	ICU_GET_INPUT_LEVEL_ID
Initializer	(uint8)0x7CU

3.8.1.19 Define ICU_GETDUTYCYCLEVALUES_ID

API service ID for Icu_GetDutyCycleValues function.

Details:

Parameters used when raising an error/exception

Table 3-21. Define ICU_GETDUTYCYCLEVALUES_ID Description

Name	ICU_GETDUTYCYCLEVALUES_ID
Initializer	(uint8)0x11

3.8.1.20 Define ICU_GETEDGENUMBERS_ID

API service ID for Icu_GetEdgeNumbers function.

Details:

Parameters used when raising an error/exception

Table 3-22. Define ICU_GETEDGENUMBERS_ID Description

Name	ICU_GETEDGENUMBERS_ID
Initializer	(uint8)0x0F

3.8.1.21 Define ICU_GETINPUTSTATE_ID

API service ID for Icu_GetInputState function.

Details:

Parameters used when raising an error/exception

Table 3-23. Define ICU_GETINPUTSTATE_ID Description

Name	ICU_GETINPUTSTATE_ID
Initializer	(uint8)0x08

3.8.1.22 Define ICU_GETTIMEELAPSED_ID

API service ID for Icu_GetTimeElapsed function.

Details:

Parameters used when raising an error/exception

Table 3-24. Define ICU_GETTIMEELAPSED_ID Description

Name	ICU_GETTIMEELAPSED_ID
Initializer	(uint8)0x10

3.8.1.23 Define ICU_GETTIMESTAMPINDEX_ID

API service ID for Icu_GetTimestampIndex function.

Details:

Parameters used when raising an error/exception

Table 3-25. Define ICU_GETTIMESTAMPINDEX_ID Description

Name	ICU_GETTIMESTAMPINDEX_ID
Initializer	(uint8)0x0B

3.8.1.24 Define ICU_GETVERSIONINFO_ID

API service ID for Icu_GetVersionInfo function.

Details:

Parameters used when raising an error/exception

Table 3-26. Define ICU_GETVERSIONINFO_ID Description

Name	ICU_GETVERSIONINFO_ID
Initializer	(uint8)0x12

3.8.1.25 Define ICU_INIT_ID

API service ID for Icu_Init function.

Details:

Parameters used when raising an error/exception

Table 3-27. Define ICU_INIT_ID Description

Name	ICU_INIT_ID
Initializer	(uint8)0x00

3.8.1.26 Define ICU_MODULE_ID

Implements: Icu_interface

Table 3-28. Define ICU_MODULE_ID Description

Name	ICU_MODULE_ID
Initializer	122

3.8.1.27 Define ICU_RESETEDGECOUNT_ID

API service ID for Icu_ResetEdgeCount function.

Details:

Parameters used when raising an error/exception

Table 3-29. Define ICU_RESETEDGECOUNT_ID Description

Name	ICU_RESETEDGECOUNT_ID
Initializer	(uint8)0x0C

3.8.1.28 Define ICU_SETACTIVATIONCONDITION_ID

API service ID for Icu_SetActivationCondition function.

Details:

Parameters used when raising an error/exception

Table 3-30. Define ICU_SETACTIVATIONCONDITION_ID Description

Name	ICU_SETACTIVATIONCONDITION_ID
Initializer	(uint8)0x05

3.8.1.29 Define ICU_SETMODE_ID

API service ID for Icu_SetMode function.

Details:

Parameters used when raising an error/exception

Table 3-31. Define ICU_SETMODE_ID Description

Name	ICU_SETMODE_ID
Initializer	(uint8)0x02

3.8.1.30 Define ICU_STARTSIGNALMEASUREMENT_ID

API service ID for Icu_StartSignalMeasurement function.

Details:

Parameters used when raising an error/exception

Table 3-32. Define ICU_STARTSIGNALMEASUREMENT_ID Description

Name	ICU_STARTSIGNALMEASUREMENT_ID
Initializer	(uint8)0x13

3.8.1.31 Define ICU_STARTTIMESTAMP_ID

API service ID for Icu_StartTimestamp function.

Details:

Parameters used when raising an error/exception

Table 3-33. Define ICU_STARTTIMESTAMP_ID Description

Name	ICU_STARTTIMESTAMP_ID
Initializer	(uint8)0x09

3.8.1.32 Define ICU_STOPSIGNALMEASUREMENT_ID

API service ID for Icu_StopSignalMeasurement function.

Details:

Parameters used when raising an error/exception

Table 3-34. Define ICU_STOPSIGNALMEASUREMENT_ID Description

Name	ICU_STOPSIGNALMEASUREMENT_ID
Initializer	(uint8)0x14

3.8.1.33 Define ICU_STOPTIMESTAMP_ID

API service ID for Icu_StopTimestamp function.

Details:

Parameters used when raising an error/exception

Table 3-35. Define ICU_STOPTIMESTAMP_ID Description

Name	ICU_STOPTIMESTAMP_ID
Initializer	(uint8)0x0A

3.8.1.34 Define ICU_SET_CLOCK_MODE_ID

API service ID for Icu_SetClockMode function.

Details:

Parameters used when raising an error/exception

Table 3-36. Define ICU_SET_CLOCK_MODE_ID Description

Name	ICU_SET_CLOCK_MODE_ID
Initializer	(uint8)0x7B

3.8.1.35 Define ICU_GET_CAPTURE_REGISTER_VALUE_ID

API service ID for Icu_StartSignalMeasurement function.

Details:

Parameters used when raising an error/exception

Table 3-37. Define ICU_GET_CAPTURE_REGISTER_VALUE_ID Description

Name	ICU_GET_CAPTURE_REGISTER_VALUE_ID
Initializer	(uint8)0x19U

3.8.1.36 Define ICU_E_ALREADY_INITIALIZED

API Icu_Init service called when the ICU driver and the Hardware are already initialized.

Implements: Icu_ErrorCodes_define

Table 3-38. Define ICU_E_ALREADY_INITIALIZED Description

Name	ICU_E_ALREADY_INITIALIZED
Initializer	(uint8)0x17

3.8.1.37 Define ICU_E_BUSY_OPERATION

API service Icu_SetMode is called while a running operation.

Implements: Icu_ErrorCodes_define

Table 3-39. Define ICU_E_BUSY_OPERATION Description

Name	ICU_E_BUSY_OPERATION
Initializer	(uint8)0x16

3.8.1.38 Define ICU_E_EDGECOUNTING_OVERFLOW

API Icu_GetEdgeNumbers service called when the Counter rolls over.

Implements: Icu_ErrorCodes_define

Table 3-40. Define ICU_E_EDGECOUNTING_OVERFLOW Description

Name	ICU_E_EDGECOUNTING_OVERFLOW
Initializer	(uint8)0xD0

3.8.1.39 Define ICU_E_MEASUREMENT_OVERFLOW

API Icu_GetTimeElapsed service called when the Time elapsed overflows.

Implements: Icu_ErrorCodes_define

Table 3-41. Define ICU_E_MEASUREMENT_OVERFLOW Description

Name	ICU_E_MEASUREMENT_OVERFLOW
Initializer	(uint8)0xD2

3.8.1.40 Define ICU_E_NOT_STARTED

API service Icu_StopTimestamp called on a channel which was not started or already stopped.

Implements: Icu_ErrorCodes_define

Table 3-42. Define ICU_E_NOT_STARTED Description

Name	ICU_E_NOT_STARTED
Initializer	(uint8)0x15

3.8.1.41 Define ICU_E_PARAM_ACTIVATION

API service used with an invalid or not feasible activation.

Implements: Icu_ErrorCodes_define

Table 3-43. Define ICU_E_PARAM_ACTIVATION Description

Name	ICU_E_PARAM_ACTIVATION
Initializer	(uint8)0x0C

3.8.1.42 Define ICU_E_PARAM_POINTER

API service used with an invalid application-buffer pointer.

Implements: Icu_ErrorCodes_define

Table 3-44. Define ICU_E_PARAM_POINTER Description

Name	ICU_E_PARAM_POINTER
Initializer	(uint8)0x0A

3.8.1.43 Define ICU_E_PARAM_BUFFER_SIZE

API service used with an invalid buffer size.

Implements: Icu_ErrorCodes_define

Table 3-45. Define ICU_E_PARAM_BUFFER_SIZE Description

Name	ICU_E_PARAM_BUFFER_SIZE
Initializer	(uint8)0x0E

3.8.1.44 Define ICU_E_PARAM_CHANNEL

API service used with an invalid channel identifier or channel was not configured for the functionality of the calling API.

Implements: Icu_ErrorCodes_define

Table 3-46. Define ICU_E_PARAM_CHANNEL Description

Name	ICU_E_PARAM_CHANNEL
Initializer	(uint8)0x0B

3.8.1.45 Define ICU_E_INIT_FAILED

API Icu_Init service called with wrong parameter.

Implements: Icu_ErrorCodes_define

Table 3-47. Define ICU_E_INIT_FAILED Description

Name	ICU_E_INIT_FAILED
Initializer	(uint8)0x0D

3.8.1.46 Define ICU_E_PARAM_MODE

API service Icu_SetMode used with an invalid mode.

Implements: Icu_ErrorCodes_define

Table 3-48. Define ICU_E_PARAM_MODE Description

Name	ICU_E_PARAM_MODE
Initializer	(uint8)0x0F

3.8.1.47 Define ICU_E_PARAM_NOTIFY_INTERVAL

API Icu_StartTimeStamp is called and the parameter NotifyInterval is invalid (e.g. 0, NotifyInterval < 1).

Implements: Icu_ErrorCodes_define

Table 3-49. Define ICU_E_PARAM_NOTIFY_INTERVAL Description

Name	ICU_E_PARAM_NOTIFY_INTERVAL
Initializer	(uint8)0x18

3.8.1.48 Define ICU_E_PARAM_VINFO

API Icu_GetVersionInfo is called and the parameter versioninfo is is invalid (e.g. NULL).

Implements: Icu_ErrorCodes_define

Table 3-50. Define ICU_E_PARAM_VINFO Description

Name	ICU_E_PARAM_VINFO
Initializer	(uint8)0x19

3.8.1.49 Define ICU_E_TIMESTAMP_OVERFLOW

API Icu_GetTimestampIndex service called when the Time stamp count overflows.

Implements: Icu_ErrorCodes_define

Table 3-51. Define ICU_E_TIMESTAMP_OVERFLOW Description

Name	ICU_E_TIMESTAMP_OVERFLOW
Initializer	(uint8)0xD1

3.8.1.50 Define ICU_E_UNINIT

API service used without module initialization.

Implements: Icu_ErrorCodes_define

Table 3-52. Define ICU_E_UNINIT Description

Name	ICU_E_UNINIT
Initializer	(uint8)0x14

3.8.1.51 Define ICU_E_PARAM_CLOCK_MODE

API Icu_SetClockMode service called with wrong parameter.

Implements: Icu_ErrorCodes_define

Table 3-53. Define ICU_E_PARAM_CLOCK_MODE Description

Name	ICU_E_PARAM_CLOCK_MODE
Initializer	(uint8)0x7A

3.8.1.52 Define ICU_E_INVALID_MODE

API Icu_EnableWakeup and Icu_DisableWakeup services called in ICU_SLEEP_MODE.

Implements: Icu_ErrorCodes_define

Table 3-54. Define ICU_E_INVALID_MODE Description

Name	ICU_E_INVALID_MODE
Initializer	(uint8)0xD3

3.8.1.53 Define ICU_E_FORBIDEN_MODE

API Check mode before set up feature for channel.

Implements: Icu_ErrorCodes_define

Table 3-55. Define ICU_E_FORBIDEN_MODE Description

Name	ICU_E_FORBIDEN_MODE
Initializer	(uint8)0xD4

3.8.1.54 Define ICU_GET_VERSION_INFO_API

Configuration of Optional API's.

Details:

Adds or removes the service`Icu_GetVersionInfo()`from the code. STD_ON:`Icu_GetVersionInfo()`can be used. STD_OFF:`Icu_GetVersionInfo()`can not be used.

Table 3-56. Define ICU_GET_VERSION_INFO_API Description

Name	ICU_GET_VERSION_INFO_API
Initializer	(STD_ON)

3.8.1.55 Define ICU_DE_INIT_API

Adds or removes the service`Icu_DeInit()`from the code. STD_ON:`Icu_DeInit()`can be used. STD_OFF:`Icu_DeInit()`can not be used.

Implements: ICU_DE_INIT_API_define

Table 3-57. Define ICU_DE_INIT_API Description

Name	ICU_DE_INIT_API
Initializer	(STD_ON)

3.8.1.56 Define ICU_SET_MODE_API

Adds or removes the service`Icu_SetMode()`from the code. STD_ON:`Icu_SetMode()`can be used. STD_OFF:`Icu_SetMode()`can not be used.

Implements: ICU_SET_MODE_API_define

Table 3-58. Define ICU_SET_MODE_API Description

Name	ICU_SET_MODE_API
Initializer	(STD_ON)

3.8.1.57 Define ICU_DISABLE_WAKEUP_API

Adds or removes the service`Icu_DisableWakeup()`from the code. STD_ON:`Icu_DisableWakeup()`can be used. STD_OFF:`Icu_DisableWakeup()`can not be used.

Implements: ICU_DISABLE_WAKEUP_API_define

Table 3-59. Define ICU_DISABLE_WAKEUP_API Description

Name	ICU_DISABLE_WAKEUP_API
Initializer	(STD_ON)

3.8.1.58 Define ICU_ENABLE_WAKEUP_API

Adds or removes the service`Icu_EnableWakeup()`from the code. STD_ON:`Icu_EnableWakeup()`can be used. STD_OFF:`Icu_EnableWakeup()`can not be used.

Implements: ICU_ENABLE_WAKEUP_API_define

Table 3-60. Define ICU_ENABLE_WAKEUP_API Description

Name	ICU_ENABLE_WAKEUP_API
Initializer	(STD_ON)

3.8.1.59 Define ICU_TIMESTAMP_API

Adds or removes all services related to the timestamping functionality as listed below from the code:`Icu_StartTimestamp()`,`Icu_StopTimestamp()`,`Icu_GetTimestampIndex()`. STD_ON: The services listed above can be used. STD_OFF: The services listed above can not be used.

Implements: ICU_TIMESTAMP_API_define

Table 3-61. Define ICU_TIMESTAMP_API Description

Name	ICU_TIMESTAMP_API
Initializer	(STD_ON)

3.8.1.60 Define ICU_EDGE_COUNT_API

Adds or removes all services related to the edge counting functionality as listed below, from the code:`Icu_ResetEdgeCount()`,`Icu_EnableEdgeCount()`,`Icu_DisableEdgeCount()`,`Icu_GetEdgeNumbers()`. STD_ON: The services listed above can be used. STD_OFF: The services listed above can not be used. .

Table 3-62. Define ICU_EDGE_COUNT_API Description

Name	ICU_EDGE_COUNT_API
Initializer	(STD_ON)

3.8.1.61 Define ICU_GET_TIME_ELAPSED_API

Adds or removes the service`Icu_GetTimeElapsed()`from the code. STD_ON:`Icu_GetTimeElapsed()`can be used. STD_OFF:`Icu_GetTimeElapsed()`can not be used.

Implements: ICU_GET_TIME_ELAPSED_API_define

Table 3-63. Define ICU_GET_TIME_ELAPSED_API Description

Name	ICU_GET_TIME_ELAPSED_API
Initializer	(STD_ON)

3.8.1.62 Define ICU_GET_DUTY_CYCLE_VALUES_API

Adds or removes the service`Icu_GetDutyCycleValues()`from the code. STD_ON:`Icu_GetDutyCycleValues()`can be used. STD_OFF:`Icu_GetDutyCycleValues()`can not be used.

Implements: ICU_GET_DUTY_CYCLE_VALUES_API_define

Table 3-64. Define ICU_GET_DUTY_CYCLE_VALUES_API Description

Name	ICU_GET_DUTY_CYCLE_VALUES_API
Initializer	(STD_ON)

3.8.1.63 Define ICU_GET_INPUT_STATE_API

Adds or removes the service`Icu_GetInputState()`from the code. STD_ON:`Icu_GetInputState()`can be used. STD_OFF:`Icu_GetInputState()`can not be used.

Implements: ICU_GET_INPUT_STATE_API_define

Table 3-65. Define ICU_GET_INPUT_STATE_API Description

Name	ICU_GET_INPUT_STATE_API
Initializer	(STD_ON)

3.8.1.64 Define ICU_SIGNAL_MEASUREMENT_API

Adds or removes the services`Icu_StartSignalMeasurement()` and `Icu_StopSignalMeasurement()`from the code. STD_ON:`Icu_StartSignalMeasurement()` and `Icu_StopSignalMeasurement()`can be used. STD_OFF:`Icu_StartSignalMeasurement()` and `Icu_StopSignalMeasurement()`can not be used.

Implements: ICU_SIGNAL_MEASUREMENT_API_define

Table 3-66. Define ICU_SIGNAL_MEASUREMENT_API Description

Name	ICU_SIGNAL_MEASUREMENT_API
Initializer	(STD_ON)

3.8.1.65 Define ICU_WAKEUP_FUNCTIONALITY_API

Adds or removes the service`Icu_CheckWakeup()`from the code. STD_ON:`Icu_CheckWakeup()`can be used. STD_OFF:`Icu_CheckWakeup()`can not be used.

Implements: ICU_WAKEUP_FUNCTIONALITY_API_define

Table 3-67. Define ICU_WAKEUP_FUNCTIONALITY_API Description

Name	ICU_WAKEUP_FUNCTIONALITY_API
Initializer	(STD_ON)

3.8.1.66 Define ICU_EDGE_DETECT_API

Adds or removes the services`Icu_EnableEdgeDetection()`and`Icu_DisableEdgeDetection()`from the code. STD_ON:`Icu_EnableEdgeDetection()`and`Icu_DisableEdgeDetection()`can be used. STD_OFF:`Icu_EnableEdgeDetection()`and`Icu_DisableEdgeDetection()`can not be used.

Implements: ICU_EDGE_DETECT_API_define

Table 3-68. Define ICU_EDGE_DETECT_API Description

Name	ICU_EDGE_DETECT_API
Initializer	(STD_ON)

3.8.1.67 Define ICU_OVERFLOW_NOTIFICATION_API

Implementation specific parameter Adds or Removes the code related to overflow notification STD_ON: Overflow notification function will be called if overflow occurs STD_OFF: Overflow notification function will not be called though overflow occurs.

Table 3-69. Define ICU_OVERFLOW_NOTIFICATION_API Description

Name	ICU_OVERFLOW_NOTIFICATION_API
Initializer	(STD_OFF)

3.8.1.68 Define ICU_GET_INPUT_LEVEL_API

Adds or removes the service`Icu_GetInputLevel()`from the code. STD_ON:`Icu_GetInputLevel()`can be used. STD_OFF:`Icu_GetInputLevel()`can not be used.

Table 3-70. Define ICU_GET_INPUT_LEVEL_API Description

Name	ICU_GET_INPUT_LEVEL_API
Initializer	(STD_OFF)

3.8.1.69 Define ICU_DUAL_CLOCK_MODE_API

Adds or removes the service`Icu_SetClockMode()`from the code. STD_ON:`Icu_SetClockMode()`can be used. STD_OFF:`Icu_SetClockMode()`can not be used.

Implements: ICU_DUAL_CLOCK_MODE_API_define

Table 3-71. Define ICU_DUAL_CLOCK_MODE_API Description

Name	ICU_DUAL_CLOCK_MODE_API

Table continues on the next page...

Table 3-71. Define ICU_DUAL_CLOCK_MODE_API Description (continued)

Initializer	(STD_OFF)

3.8.1.70 Define ICU_TIMESTAMP_USES_DMA

Implements: ICU_TIMESTAMP_USES_DMA_define

Table 3-72. Define ICU_TIMESTAMP_USES_DMA Description

Name	ICU_TIMESTAMP_USES_DMA
Initializer	(STD_OFF)

3.8.1.71 Define ICU_CAPTURERGISTER_API

Adds or removes the service`Icu_GetCaptureRegisterValue()`from the code.
STD_ON:`Icu_GetCaptureRegisterValue()`can be used.
STD_OFF:`Icu_GetCaptureRegisterValue()`can not be used.

Implements: ICU_CAPTURERGISTER_API_define

Table 3-73. Define ICU_CAPTURERGISTER_API Description

Name	ICU_CAPTURERGISTER_API
Initializer	(STD_OFF)

3.8.1.72 Define ICU_REPORT_WAKEUP_SOURCE

Switch for enabling Wakeup source reporting. STD_ON: Report Wakeup source.
STD_OFF: Do not report Wakeup source.

Table 3-74. Define ICU_REPORT_WAKEUP_SOURCE Description

Name	ICU_REPORT_WAKEUP_SOURCE
Initializer	(STD_ON)

3.8.1.73 Define ICU_DEV_ERROR_DETECT

Switches the Development Error Detection and Notification on or off. STD_ON: Enabled. STD_OFF: Disabled.

Implements: ICU_DEV_ERROR_DETECT_define

Table 3-75. Define ICU_DEV_ERROR_DETECT Description

Name	ICU_DEV_ERROR_DETECT
Initializer	(STD_ON)

3.8.1.74 Define ICU_ENABLE_USER_MODE_SUPPORT

Support for User mode.If this parameter has been configured to 'STD_ON', the ICU driver code can be executed from both supervisor and user mode.

Implements: ICU_ENABLE_USER_MODE_SUPPORT_define

Table 3-76. Define ICU_ENABLE_USER_MODE_SUPPORT Description

Name	ICU_ENABLE_USER_MODE_SUPPORT
Initializer	(STD_OFF)

3.8.2 Enum Reference

Enumeration of all constants supported by the driver are as per AUTOSAR ICU Driver software specification Version 4.2 Rev0002 .

3.8.2.1 Enumeration Icu_ActivationType

Definition of the type of activation of an ICU channel.

Implements: Icu_ActivationType_enumeration

Table 3-77. Enumeration Icu_ActivationType Values

Name	Initializer	Description
ICU_FALLING_EDGE	0x0U	An appropriate action shall be executed when a falling edge occurs on the ICU input signal.
ICU_RISING_EDGE	0x1U	An appropriate action shall be executed when a rising edge occurs on the ICU input signal.
ICU_BOTH_EDGES	0x2U	An appropriate action shall be executed when a falling or rising edge occurs on the ICU input signal.

3.8.2.2 Enumeration Icu_InputStateType

Input state of an ICU channel.

Implements: Icu_InputState_enumeration

Table 3-78. Enumeration Icu_InputStateType Values

Name	Initializer	Description
ICU_ACTIVE	0U	An activation edge has been detected
ICU_IDLE	1U	No activation edge has been detected since the last call of Icu_GetInputState() or Icu_Init().

3.8.2.3 Enumeration Icu_MeasurementModeType

Definition of the measurement mode type.

Implements: Icu_MeasurementModeType_enumeration

Table 3-79. Enumeration Icu_MeasurementModeType Values

Name	Initializer	Description
ICU_MODE_SIGNAL_EDGE_DETECT	0x01U	Mode for detecting edges
ICU_MODE_SIGNAL_MEASUREMENT	0x02U	Mode for measuring different times between various configurable edges
ICU_MODE_TIMESTAMP	0x04U	Mode for capturing timer values on configurable edges
ICU_MODE_EDGE_COUNTER	0x08U	Mode for counting edges on configurable edges.

3.8.2.4 Enumeration Icu_ModeType

Allow enabling or disabling of all interrupts which are not required for the ECU wakeup.

Implements: Icu_ModeType_enumeration

Table 3-80. Enumeration Icu_ModeType Values

Name	Initializer	Description
ICU_MODE_NORMAL	0U	Normal operation, all used interrupts are enabled according to the notification requests.
ICU_MODE_SLEEP	1U	Reduced power operation. In sleep mode only those notifications are available which are configured as wakeup capable.

3.8.2.5 Enumeration Icu_SignalMeasurementPropertyType

Definition of the measurement property type.

Implements: Icu_SignalMeasurementPropertyType_enumeration

Table 3-81. Enumeration Icu_SignalMeasurementPropertyType Values

Name	Initializer	Description
ICU_LOW_TIME	0x01U	The channel is configured for reading the elapsed Signal Low Time
ICU_HIGH_TIME	0x02U	The channel is configured for reading the elapsed Signal High Time
ICU_PERIOD_TIME	0x04U	The channel is configured for reading the elapsed Signal Period Time

Table continues on the next page...

Table 3-81. Enumeration Icu_SignalMeasurementPropertyType Values (continued)

Name	Initializer	Description
ICU_DUTY_CYCLE	0x08U	The channel is configured to read values which are needed for calculating the duty cycle (coherent Active and Period Time).

3.8.2.6 Enumeration Icu_TimestampBufferType

Definition of the timestamp measurement property type.

Implements: Icu_TimestampBufferType_enumeration

Table 3-82. Enumeration Icu_TimestampBufferType Values

Name	Initializer	Description
ICU_LINEAR_BUFFER	0U	The buffer will just be filled once
ICU_CIRCULAR_BUFFER	1U	After reaching the end of the buffer, the driver restarts at the beginning of the buffer.

3.8.2.7 Enumeration Icu_LevelType

Definition of the actual status of PIN property type.

Implements: Icu_LevelType_enumeration

Table 3-83. Enumeration Icu_LevelType Values

Name	Initializer	Description
ICU_LEVEL_LOW	0x0U	Default Input PIN Status
ICU_LEVEL_HIGH	0x1U	High level Input PIN Status.

3.8.2.8 Enumeration Icu_SelectPrescalerType

Definition of prescaler property type.

Implements: Icu_SelectPrescalerType_enumeration

Table 3-84. Enumeration Icu_SelectPrescalerType Values

Name	Initializer	Description
ICU_NORMAL_CLOCK_MODE	0x0U	Default channel prescaler.
ICU_ALTERNATE_CLOCK_MODE	0x1U	Alternate channel prescaler mode.

3.8.3 Function Reference

Functions of all functions supported by the driver are as per AUTOSAR ICU Driver software specification Version 4.2 Rev0002 .

3.8.3.1 Function Icu_Init

This function initializes the driver.

Details:

This service is a non reentrant function used for driver initialization. The Initialization function shall initialize all relevant registers of the configured hardware with the values of the structure referenced by the parameter ConfigPtr. If the hardware allows for only one usage of the register, the driver module implementing that functionality is responsible for initializing the register. The initialization function of this module shall always have a pointer as a parameter, even though for Variant PC no configuration set shall be given. Instead a NULL pointer shall be passed to the initialization function. The Icu module environment shall not call Icu_Init during a running operation (e. g. timestamp measurement or edge counting).

Return: void.

Implements: Icu_Init_Activity

Violates: Violates MISRA 2004 Required Rule 8.10 could be made static

Prototype: `void Icu_Init(const Icu_ConfigType *ConfigPtr);`

Table 3-85. Icu_Init Arguments

Type	Name	Direction	Description
const`Icu_ConfigType`*	ConfigPtr	**input**	Pointer to a selected configuration structure.

3.8.3.2 Function Icu_DeInit

This function de-initializes the ICU module.

Details:

This service is a Non reentrant function used for ICU De-Initialization After the call of this service, the state of the peripherals used by configuration shall be the same as after power on reset. Values of registers which are not writable are excluded. This service shall disable all used interrupts and notifications. The Icu module environment shall not call Icu_DeInit during a running operation (e. g. timestamp measurement or edge counting)

Return: void.

Pre: Icu_Init must be called before.

Implements: Icu_DeInit_Activity

Violates: Violates MISRA 2004 Required Rule 8.10 could be made static

Prototype: `void Icu_DeInit(void);`

3.8.3.3 Function Icu_SetMode

This function sets the ICU mode.

Details:

This service is a non reentrant function used for ICU mode selection. This service shall set the operation mode to the given mode parameter. This service can be called during running operations. If so, an ongoing operation that generates interrupts on a wakeup capable channel like e.g. time stamping or edge counting might lead to the ICU module not being able to properly enter sleep mode. This is then a system or ECU configuration issue not a problem of this specification.

Return: void.

Pre: Icu_Init must be called before.

Implements: Icu_SetMode_Activity

Violates: Violates MISRA 2004 Required Rule 8.10 could be made static

Prototype: `void Icu_SetMode(Icu_ModeType Mode);`

Table 3-86. Icu_SetMode Arguments

Type	Name	Direction	Description
`Icu_ModeType`	Mode	**input**	Specifies the operation mode.

3.8.3.4 Function Icu_GetVersionInfo

This service returns the version information of this module.

Details:

This service is Non reentrant and returns the version information of this module. The version information includes:

- Module Id
- Vendor Id
- Vendor specific version numbers If source code for caller and callee of this function is available this function should be realized as a macro. The macro should be defined in the modules header file.

Return: void.

Implements: Icu_GetVersionInfo_Activity

Violates: Violates MISRA 2004 Required Rule 8.10 could be made static

Prototype: `void Icu_GetVersionInfo(Std_VersionInfoType *versioninfo);`

Table 3-87. Icu_GetVersionInfo Arguments

Type	Name	Direction	Description
Std_VersionInfoType *	versioninfo	**output**	Pointer to location to store version info.

3.8.3.5 Function Icu_CheckWakeup

Checks if a wakeup capable ICU channel is the source for a wakeup event.

Details:

The function calls the ECU state manager service EcuM_SetWakeupEvent in case of a valid ICU channel wakeup event.

Return: void.

Pre: Icu_Init must be called before. The channel must be configured as wakeup capable.

Implements: Icu_CheckWakeup_Activity

Violates: Violates MISRA 2004 Required Rule 8.10 could be made static

Prototype: `void Icu_CheckWakeup(EcuM_WakeupSourceType WakeupSource);`

Table 3-88. Icu_CheckWakeup Arguments

Type	Name	Direction	Description
EcuM_WakeupSourceType	WakeupSource	**input**	Information on wakeup source to be checked.

3.8.3.6 Function Icu_DisableWakeup

This function disables the wakeup capability of a single ICU channel.

Details:

This service is reentrant function and shall disable the wakeup capability of a single ICU channel.This service is only feasible for ICU channels configured statically as wakeup capable true. The function Icu_DisableWakeup shall be pre compile time configurable On,Off by the configuration parameter IcuDisableWakeupApi.

Return: void.

Pre: Icu_Init must be called before.

Implements: Icu_DisableWakeup_Activity

Violates: Violates MISRA 2004 Required Rule 8.10 could be made static

Prototype: `void Icu_DisableWakeup(Icu_ChannelType Channel);`

Table 3-89. Icu_DisableWakeup Arguments

Type	Name	Direction	Description
`Icu_ChannelType`	Channel	**input**	Numeric identifier of the ICU channel.

3.8.3.7 Function Icu_EnableWakeup

This function (re-)enables the wakeup capability of the given ICU channel.

Details:

The function is reentrant and re-enable the wake-up capability of a single ICU channel.

Return: void.

Pre: Icu_Init must be called before. The channel must be configured as wakeup capable.

Implements: Icu_EnableWakeup_Activity

Violates: Violates MISRA 2004 Required Rule 8.10 could be made static

Prototype: `void Icu_EnableWakeup(Icu_ChannelType Channel);`

Table 3-90. Icu_EnableWakeup Arguments

Type	Name	Direction	Description
`Icu_ChannelType`	Channel	**input**	Numeric identifier of the ICU channel.

3.8.3.8 Function Icu_DisableEdgeCount

This function disables the counting of edges of the given channel.

Details:

This function is reentrant and disables the counting of edges of the given channel.

Return: void.

Pre: Icu_Init must be called before.The given channel must be configured in Measurement Mode Edge Counter.

Implements: Icu_DisableEdgeCount_Activity

Violates: Violates MISRA 2004 Required Rule 8.10 could be made static

Prototype: `void Icu_DisableEdgeCount(Icu_ChannelType Channel);`

Table 3-91. Icu_DisableEdgeCount Arguments

Type	Name	Direction	Description
`Icu_ChannelType`	Channel	**input**	Numeric identifier of the ICU channel.

3.8.3.9 Function Icu_EnableEdgeCount

This function enables the counting of edges of the given channel.

Details:

This service is reentrant and shall enable the counting of edges of the given channel. Note: This service doesnot do the real counting itself. This is done by the hardware (capture unit). Only the configured edges shall be counted (rising edge, falling edge or both edges).

Configuration of the edge is done in Icu_Init or Icu_SetActivationCondition. The configured edge can be changed during runtime using Icu_SetActivationCondition. Interrupts are not required for edge counting.If interrupts are enabled, the interrupt service routine will set the overflow flag if more than 0xFFFFFF edges are measured.

Return: void.

Pre: Icu_Init must be called before. The given channel must be configured in Measurement Mode Edge Counter.

Implements: Icu_EnableEdgeCount_Activity

Violates: Violates MISRA 2004 Required Rule 8.10 could be made static

Prototype: `void Icu_EnableEdgeCount(Icu_ChannelType Channel);`

Table 3-92. Icu_EnableEdgeCount Arguments

Type	Name	Direction	Description
`Icu_ChannelType`	Channel	**input**	Numeric identifier of the ICU channel.

3.8.3.10 Function Icu_ResetEdgeCount

This function resets the value of the counted edges to zero.

Details:

This function is reentrant and resets the value of the counted edges to zero.

Return: void.

Pre: Icu_Init must be called before.

Implements: Icu_ResetEdgeCount_Activity

Violates: Violates MISRA 2004 Required Rule 8.10 could be made static

Prototype: `void Icu_ResetEdgeCount(Icu_ChannelType Channel);`

Table 3-93. Icu_ResetEdgeCount Arguments

Type	Name	Direction	Description
`Icu_ChannelType`	Channel	**input**	Numeric identifier of the ICU channel.

3.8.3.11 Function Icu_GetEdgeNumbers

This function reads the number of counted edges.

Details:

This function is reentrant reads the number of counted edges after the last call of`Icu_ResetEdgeCount()`.

Return: Icu_EdgeNumberType - Number of the counted edges.

Pre: Icu_Init must be called before. The given channel must be configured in Measurement Mode Edge Counter.

Implements: Icu_GetEdgeNumbers_Activity

Violates: Violates MISRA 2004 Required Rule 8.10 could be made static

Prototype: `Icu_EdgeNumberType Icu_GetEdgeNumbers(Icu_ChannelType Channel);`

Table 3-94. Icu_GetEdgeNumbers Arguments

Type	Name	Direction	Description
`Icu_ChannelType`	Channel	**input**	Numeric identifier of the ICU channel.

3.8.3.12 Function Icu_DisableEdgeDetection

This function disables the detection of edges of the given channel.

Details:

This function is reentrant and disables the detection of edges of the given channel.

Return: void.

Pre: Icu_Init must be called before. The given channel must be configured in Measurement Mode Edge Detection.

Implements: Icu_DisableEdgeDetection_Activity

Violates: Violates MISRA 2004 Required Rule 8.10 could be made static

Prototype: `void Icu_DisableEdgeDetection(Icu_ChannelType Channel);`

Table 3-95. Icu_DisableEdgeDetection Arguments

Type	Name	Direction	Description
`Icu_ChannelType`	Channel	**input**	Numeric identifier of the ICU channel.

3.8.3.13 Function Icu_EnableEdgeDetection

This function enables or re-enables the detection of edges of the given channel.

Details:

This function is reentrant enables or re-enables the detection of edges of the given channel.

Return: void.

Pre: Icu_Init must be called before. The given channel must be configured in Measurement Mode Edge Counter.

Implements: Icu_EnableEdgeDetection_Activity

Violates: Violates MISRA 2004 Required Rule 8.10 could be made static

Prototype: `void Icu_EnableEdgeDetection(Icu_ChannelType Channel);`

Table 3-96. Icu_EnableEdgeDetection Arguments

Type	Name	Direction	Description
`Icu_ChannelType`	Channel	**input**	Numeric identifier of the ICU channel.

3.8.3.14 Function Icu_DisableNotification

This function disables the notification of a channel.

Details:

This function is reentrant and disables the notification of a channel.

Return: void.

Pre: Icu_Init must be called before.

Implements: Icu_DisableNotification_Activity

Violates: Violates MISRA 2004 Required Rule 8.10 could be made static

Prototype: `void Icu_DisableNotification(Icu_ChannelType Channel);`

Table 3-97. Icu_DisableNotification Arguments

Type	Name	Direction	Description
`Icu_ChannelType`	Channel	**input**	Numeric identifier of the ICU channel.

3.8.3.15 Function Icu_EnableNotification

This function enables the notification on the given channel.

Details:

This function is reentrant and enables the notification on the given channel. The notification will be reported only when the channel measurement property is enabled or started

Return: void.

Pre: Icu_Init must be called before.

Implements: Icu_EnableNotification_Activity

Violates: Violates MISRA 2004 Required Rule 8.10 could be made static

Prototype: `void Icu_EnableNotification(Icu_ChannelType Channel);`

Table 3-98. Icu_EnableNotification Arguments

Type	Name	Direction	Description
`Icu_ChannelType`	Channel	**input**	Numeric identifier of the ICU channel.

3.8.3.16 Function Icu_SetActivationCondition

This function sets the activation-edge for the given channel.

Details:

This service is reentrant and shall set the activation-edge according to Activation parameter for the given channel. This service shall support channels which are configured for the following Icu_MeasurementMode:

- ICU_MODE_SIGNAL_EDGE_DETECT
- ICU_MODE_TIMESTAMP
- ICU_MODE_EDGE_COUNTER

Return: void.

Pre: Icu_Init must be called before. The channel must be properly configured (ICU_MODE_SIGNAL_EDGE_DETECT, ICU_MODE_TIMESTAMP, ICU_MODE_EDGE_COUNTER).

Implements: Icu_SetActivationCondition_Activity

Violates: Violates MISRA 2004 Required Rule 8.10 could be made static

Prototype: `void Icu_SetActivationCondition(Icu_ChannelType Channel, Icu_ActivationType Activation);`

Table 3-99. Icu_SetActivationCondition Arguments

Type	Name	Direction	Description
`Icu_ChannelType`	Channel	**input**	Numeric identifier of the ICU channel.
`Icu_ActivationType`	Activation	**input**	Type of activation.

3.8.3.17 Function Icu_GetInputLevel

This function returns the actual status of PIN.

Details:

This function returns the actual status of PIN. Icu_GetInputLevel() function only supports channels using FTM.

Return: Icu_LevelType.

Pre: Icu_Init must be called before.

Implements: Icu_GetInputLevel_Activity

Prototype: `Icu_LevelType Icu_GetInputLevel(Icu_ChannelType Channel);`

Table 3-100. Icu_GetInputLevel Arguments

Type	Name	Direction	Description
`Icu_ChannelType`	Channel	**input**	Numeric identifier of the ICU channel.

3.8.3.18 Function Icu_GetInputState

This function returns the status of the ICU input.

Details:

This service is reentrant shall return the status of the ICU input. Only channels which are configured for the following Icu_MeasurementMode shall be supported:

- ICU_MODE_SIGNAL_EDGE_DETECT,
- ICU_MODE_SIGNAL_MEASUREMENT.

Return: Icu_InputStateType.

Pre: Icu_Init must be called before.

Implements: Icu_GetInputState_Activity

Violates: Violates MISRA 2004 Required Rule 8.10 could be made static

Prototype: `Icu_InputStateType Icu_GetInputState(Icu_ChannelType Channel);`

Table 3-101. Icu_GetInputState Arguments

Type	Name	Direction	Description
`Icu_ChannelType`	Channel	**input**	Numeric identifier of the ICU channel.

Table 3-102. Icu_GetInputState Return Values

Name	Description
ICU_ACTIVE	An activation edge has been detected.
ICU_IDLE	No activation edge has been detected since the last call of Icu_GetInputState() or Icu_Init().

3.8.3.19 Function Icu_StartSignalMeasurement

This function starts the measurement of signals.

Details:

This service is reentrant and starts the measurement of signals beginning with the configured default start edge which occurs first after the call of this service. This service shall only be available in Measurement Mode ICU_MODE_SIGNAL_MEASUREMENT. This service shall reset the state for the given channel to ICU_IDLE.

Return: void.

Pre: Icu_Init must be called before. The given channel must be configured in Measurement Mode Signal Measurement.

Implements: Icu_StartSignalMeasurement_Activity

Violates: Violates MISRA 2004 Required Rule 8.10 could be made static

Prototype: `void Icu_StartSignalMeasurement(Icu_ChannelType Channel);`

Table 3-103. Icu_StartSignalMeasurement Arguments

Type	Name	Direction	Description
`Icu_ChannelType`	Channel	**input**	Numeric identifier of the ICU channel.

3.8.3.20 Function Icu_StopSignalMeasurement

This function stops the measurement of signals of the given channel.

Details:

This function is reentrant and stops the measurement of signals of the given channel.

Return: void.

Pre: Icu_Init must be called before. The given channel must be configured in Measurement Mode Signal Measurement.

Implements: Icu_StopSignalMeasurement_Activity

Violates: Violates MISRA 2004 Required Rule 8.10 could be made static

Prototype: `void Icu_StopSignalMeasurement(Icu_ChannelType Channel);`

Table 3-104. Icu_StopSignalMeasurement Arguments

Type	Name	Direction	Description
`Icu_ChannelType`	Channel	**input**	Numeric identifier of the ICU channel.

3.8.3.21 Function Icu_GetDutyCycleValues

This function reads the coherent active time and period time for the given ICU Channel.

Details:

The function is reentrant and reads the coherent active time and period time for the given ICU Channel, if it is configured in Measurement Mode Signal Measurement, Duty Cycle Values.

Return: void.

@ implements Icu_GetDutyCycleValues_Activity

Pre: Icu_Init must be called before. The given channel must be configured in Measurement Mode Signal Measurement, Duty Cycle Values.

Violates: Violates MISRA 2004 Required Rule 8.10 could be made static

Prototype: `void Icu_GetDutyCycleValues(Icu_ChannelType Channel, Icu_DutyCycleType *DutyCycleValues);`

Table 3-105. Icu_GetDutyCycleValues Arguments

Type	Name	Direction	Description
`Icu_ChannelType`	Channel	**input**	Numeric identifier of the ICU channel.
`Icu_DutyCycleType*`	DutyCycleValues	**output**	Pointer to a buffer where the results (high time and period time) shall be placed.

3.8.3.22 Function Icu_GetTimeElapsed

This function reads the elapsed Signal Low, High or Period Time for the given channel.

Details:

This service is reentrant and reads the elapsed Signal Low Time for the given channel that is configured in Measurement Mode Signal Measurement, Signal Low Time. The elapsed time is measured between a falling edge and the consecutive rising edge of the channel. This service reads the elapsed Signal High Time for the given channel that is configured in Measurement Mode Signal Measurement,Signal High Time.The elapsed time is measured between a rising edge and the consecutive falling edge of the channel. This service reads the elapsed Signal Period Time for the given channel that is configured in Measurement Mode Signal Measurement, Signal Period Time.The elapsed time is measured between consecutive rising (or falling) edges of the channel. The period start edge is

configurable.

Return: Icu_ValueType - the elapsed Signal Low Time for the given channel that is configured in Measurement Mode Signal Measurement, Signal Low Time.

Pre: Icu_Init must be called before. The given channel must be configured in Measurement Mode Signal Measurement.

Implements: Icu_GetTimeElapsed_Activity

Violates: Violates MISRA 2004 Required Rule 8.10 could be made static

Prototype: `Icu_ValueType Icu_GetTimeElapsed(Icu_ChannelType Channel);`

Table 3-106. Icu_GetTimeElapsed Arguments

Type	Name	Direction	Description
`Icu_ChannelType`	Channel	**input**	Numeric identifier of the ICU channel.

3.8.3.23 Function Icu_GetTimestampIndex

This function reads the timestamp index of the given channel.

Details:

This function reentrant and reads the timestamp index of the given channel, which is next to be written.

Return: Icu_IndexType - Timestamp index of the given channel.

Pre: Icu_Init must be called before. Icu_StartTimestamp must be called before.

Implements: Icu_GetTimestampIndex_Activity

Violates: Violates MISRA 2004 Required Rule 8.10 could be made static

Prototype: `Icu_IndexType Icu_GetTimestampIndex(Icu_ChannelType Channel);`

Table 3-107. Icu_GetTimestampIndex Arguments

Type	Name	Direction	Description
`Icu_ChannelType`	Channel	**input**	Numeric identifier of the ICU channel.

3.8.3.24 Function Icu_StartTimestamp

This function starts the capturing of timer values on the edges.

Details:

This function is reentrant and starts the capturing of timer values on the edges activated by the service`Icu_SetActivationCondition()`to an external buffer.

Return: void.

Pre: Icu_Init must be called before.

Implements: Icu_StartTimestamp_Activity

Violates: Violates MISRA 2004 Required Rule 8.10 could be made static

Prototype: `void Icu_StartTimestamp(Icu_ChannelType Channel, Icu_ValueType *BufferPtr, uint16 BufferSize, uint16 NotifyInterval);`

Table 3-108. Icu_StartTimestamp Arguments

Type	Name	Direction	Description
`Icu_ChannelType`	Channel	**input**	Numeric identifier of the ICU channel.
`Icu_ValueType*`	BufferPtr	**input**	Pointer to the buffer-array where the timestamp values shall be placed.
uint16	BufferSize	**input**	Size of the external buffer (number of entries).
uint16	NotifyInterval	**input**	Notification interval (number of events).

3.8.3.25 Function Icu_StopTimestamp

This function stops the timestamp measurement of the given channel.

Details:

This function is reentrant and stops the timestamp measurement of the given channel.

Return: void.

Pre: Icu_Init must be called before.

Implements: Icu_StopTimestamp_Activity

Violates: Violates MISRA 2004 Required Rule 8.10 could be made static

Prototype: `void Icu_StopTimestamp(Icu_ChannelType Channel);`

Table 3-109. Icu_StopTimestamp Arguments

Type	Name	Direction	Description
`Icu_ChannelType`	Channel	**input**	Numeric identifier of the ICU channel.

3.8.3.26 Function Icu_SetClockMode

This function sets the clock mode for the driver

Details:

This function is reentrant and it changes the prescaler for the driver, changing the power consumption of the driver

Return: void.

Pre: Icu_Init must be called before.

Implements: Icu_SetClockMode_Activity

Violates: Violates MISRA 2004 Required Rule 8.10 could be made static

Prototype: `void Icu_SetClockMode(Icu_SelectPrescalerType Prescaler);`

Table 3-110. Icu_SetClockMode Arguments

Type	Name	Direction	Description
`Icu_SelectPrescalerType`	Prescaler	**input**	Prescaler type: Normal or Alternate

3.8.3.27 Function Icu_GetCaptureRegisterValue

This service returns the value of Capture register.

Details:

This service returns the value of Capture register.

Return: Icu_ValueType.

Pre: Icu_Init must be called before.

Implements: Icu_GetCaptureRegisterValue_Activity

Violates: Violates MISRA 2004 Required Rule 8.10 could be made static

Prototype: `Icu_ValueType Icu_GetCaptureRegisterValue_Activity(Icu_ChannelType Channel);`

Table 3-111. Icu_GetCaptureRegisterValue_Activity Arguments

Type	Name	Direction	Description
`Icu_ChannelType`	Channel	**input**	Numeric identifier of the ICU channel.

3.8.4 Structs Reference

Data structures supported by the driver are as per AUTOSAR ICU Driver software specification Version 4.2 Rev0002 .

3.8.4.1 Structure Icu_ChannelConfigType

Structure that contains ICU channel configuration.

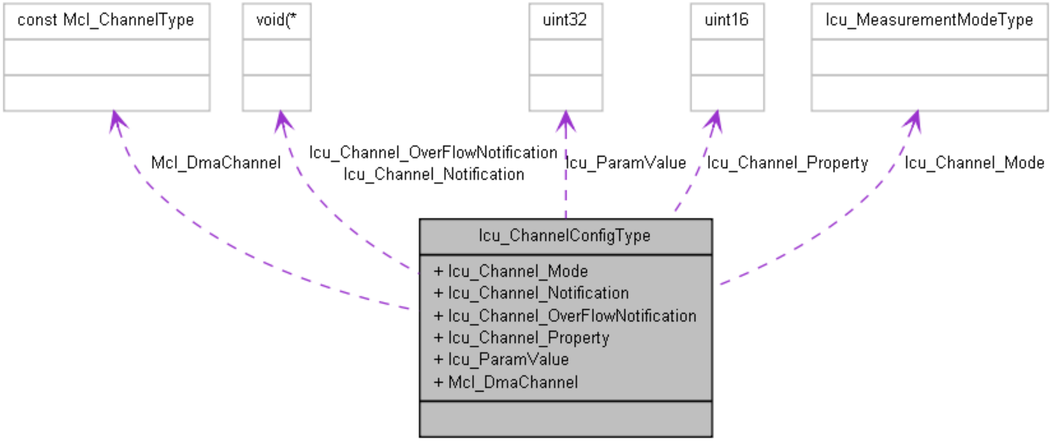


Figure 3-1. Struct Icu_ChannelConfigType

Details:

It contains the information like Icu Channel Mode, Channel Notification function, overflow Notification function.

Implements: Icu_ChannelConfigType_struct

Declaration:

```
typedef struct
                    {
                          Icu_MeasurementModeType Icu_Channel_Mode,
                          Icu_NotifyType Icu_Channel_Notification,
                          Icu_NotifyType Icu_Channel_OverFlowNotification,
                          Icu_MeasurementSubModeType Icu_Channel_Property,
                          Icu_ParamType Icu_ParamValue,
                          Icu_WakeupValueType Icu_Channel_WakeupValue,
                          const Mcl_ChannelType Mcl_DmaChannel
                    } Icu_ChannelConfigType;
```

Table 3-112. Structure Icu_ChannelConfigType member description

Member	Description
Icu_Channel_Mode	The measurement mode for a channel (Timestamp, Signal Measurement, Signal Edge Detect, Edge Counter)
Icu_Channel_Notification	The configured notification for IcuChannel
Icu_Channel_OverFlowNotification	The overflow notification for a specific channel
Icu_Channel_Property	The measurement property for a specific channel
Icu_ParamValue	Specific configuration for a channel such as edge polarity, prescalers

Table continues on the next page...

Table 3-112. Structure Icu_ChannelConfigType member description (continued)

Member	Description
Mcl_DmaChannel	Reference to the MCL channel
Icu_Channel_WakeupValue	Reference to the EcuM wakeup source

3.8.4.2 Structure Icu_ConfigType

This type contains initialization data.

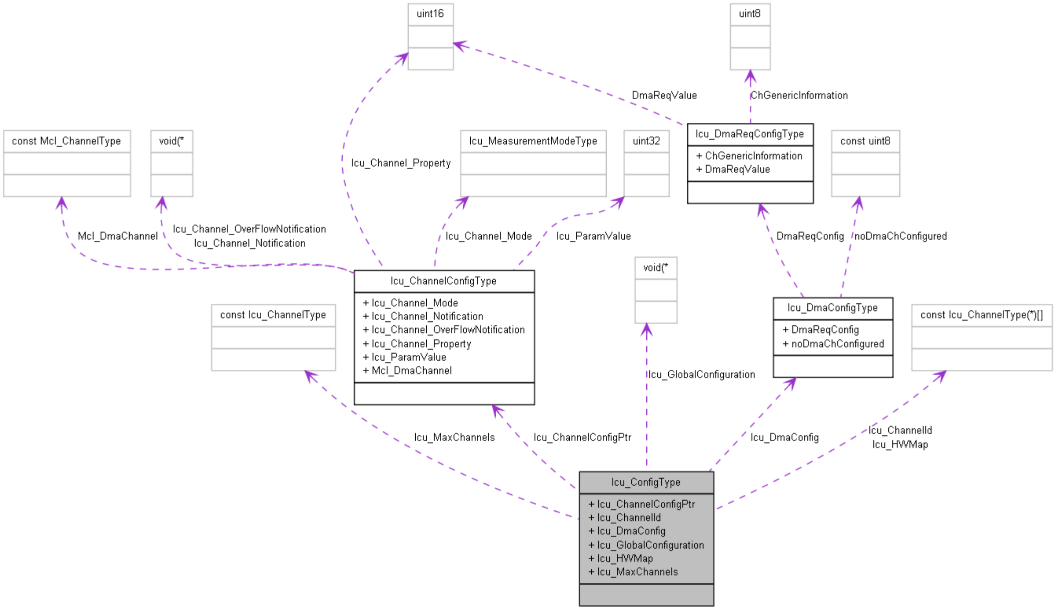


Figure 3-2. Struct Icu_ConfigType

Details:

The notification functions shall be configurable as function pointers within the initialization data structure (`Icu_ConfigType`). This type of the external data structure shall contain the initialization data for the ICU driver. It shall contain:

- Wakeup Module Info (in case the wakeup-capability is true)
- ICU dependent properties for used HW units
- Clock source with optional prescaler (if provided by HW)

Implements: Icu_ConfigType_struct

Declaration:

```
typedef struct
                    {
        CONST(Icu_ChannelConfigType, ICU_CONST)(*Icu_ChannelConfigPtr)[];
        CONST(Icu_IpConfigType, ICU_CONST)IpConfig;
        CONST(Icu_ChannelType, ICU_CONST)(* Icu_HWMap)[];
        CONST(Icu_ChannelType, ICU_CONST)nNumChannels;
                    } Icu_ConfigType;
```

Table 3-113. Structure Icu_ConfigType member description

Member	Description
Icu_ChannelConfigPtr	Pointer to the list of Icu configured channels.
IpConfig	Combined IP specific configuration structure.
Icu_HWMap	Index table to translate HW channels to logical used to process interrupts for notifications.
nNumChannels	The number of configured channels.

3.8.4.3 Structure Icu_IpConfigType

Structure that Combined IP specific configuration structure value and active time value.

Implements: Icu_IpConfigType_struct

Declaration:

```
typedef struct
                    {
                      Icu_Ftm_IpConfigType pFtmIpConfig,
                      Icu_PORT_CI_IpConfigType pPORT_CIIpConfig,
                      Icu_LPit_IpConfigType pLPitIpConfig,
                      Icu_Lptmr_IpConfigType pLptmrIpConfig,
                      Icu_IpChannelConfigType (*pIpChannelsConfig)[]
                    } Icu_IpConfigType;
```

Table 3-114. Structure Icu_IpConfigType member description

Member	Description
pFtmIpConfig	Pointer to the structure containing Ftm configuration
pPORT_CIIpConfig	Pointer to the structure containing PORT_CI configuration
pLPitIpConfig	Pointer to the structure containing LPIT configuration
pLptmrIpConfig	Pointer to the structure containing LPTMR configuration
pIpChannelsConfig	Pointer to Array containing IP type and index in the IP configuration table for each Icu channel

3.8.4.4 Structure Icu_IpChannelConfigType

Structure that Icu channel high level configuration structure.

Implements: Icu_IpChannelConfigType_struct

Declaration:

```
typedef struct
                    {
                      Icu_ChannelIpType nChannelIp,
                      uint8 u8IdxChannelConfig
                    } Icu_IpChannelConfigType;
```

Table 3-115. Structure Icu_IpChannelConfigType member description

Member	Description
nChannelIp	The IP used to implement this specific Icu channel
u8IdxChannelConfig	Index in the IP specific configuration table

3.8.4.5 Structure Icu_PORT_CI_IpConfigType

Structure that combine PORT_CI IP specific configuration structure type.

Implements: Icu_PORT_CI_IpConfigType_struct

Declaration:

```
typedef struct
                    {
                      Icu_PORT_CI_ChannelType nNumChannels,
                      Icu_PORT_CI_ChannelConfigType (*pChannelsConfig)[]
                    } Icu_PORT_CI_IpConfigType;
```

Table 3-116. Structure Icu_PORT_CI_IpConfigType member description

Member	Description
nNumChannels	Number of PORT_CI channels in the Icu configuration
pChannelsConfig	Pointer to the configured channels for PORT_CI

3.8.4.6 Structure Icu_PORT_CI_ChannelConfigType

Structure that combine configuration of each PORT_CI channels in the Icu.

Implements: Icu_PORT_CI_ChannelConfigType_struct

Declaration:

```
typedef struct
                    {
                      Icu_PORT_CI_ChannelType nHwChannel,
                      uint8                   nDefaultStartEdge
                    } Icu_PORT_CI_ChannelConfigType;
```

Table 3-117. Structure Icu_PORT_CI_ChannelConfigType member description

Member	Description
nHwChannel	Assigned PORT_CI channel id.
nDefaultStartEdge	PORT_CI Default Start Edge.

3.8.4.7 Structure Icu_Ftm_IpConfigType

Structure that combine number of Ftm channels in the Icu configuration.

Implements: Icu_Ftm_IpConfigType_struct

Declaration:

```
typedef struct
                    {
                      Icu_Ftm_ChannelType nNumChannels,
                      Icu_Ftm_ChannelConfigType (*pChannelsConfig)[]
                    } Icu_Ftm_IpConfigType;
```

Table 3-118. Structure Icu_Ftm_IpConfigType member description

Member	Description
nNumChannels	Number of Ftm channels in the Icu configuration
pChannelsConfig	Pointer to the configured channels for Ftm

3.8.4.8 Structure Icu_Ftm_ChannelConfigType

Structure that combine configuration of each Ftm channels in the Icu.

Implements: Icu_Ftm_ChannelConfigType_struct

Declaration:

```
typedef struct
                    {
                      Ftm_IcuChannelType nHwChannel,
                      Icu_Ftm_ControlType nControlValue,
                      uint8 nDefaultStartEdge,
                      uint8 nMeasurementModeType
                    } Icu_Ftm_ChannelConfigType;
```

Table 3-119. Structure Icu_Ftm_ChannelConfigType member description

Member	Description
nHwChannel	Assigned Ftm channel id.
nControlValue	Ftm channel parameters.
nDefaultStartEdge	Ftm Default Start Edge.
nMeasurementModeType	Ftm Measurement Mode .

3.8.4.9 Structure Icu_Lptmr_IpConfigType

Structure that combine Lptmr IP specific configuration structure type.

Implements: Icu_Lptmr_IpConfigType_struct

Declaration:

```
typedef struct
                     {
                       Icu_Lptmr_ChannelType nNumChannels,
                       Icu_Lptmr_GlobalConfigurationType Icu_GlobalConfiguration,
                       Icu_Lptmr_ChannelConfigType (*pChannelsConfig)[]
                     } Icu_Lptmr_IpConfigType;
```

Table 3-120. Structure Icu_Lptmr_IpConfigType member description

Member	Description
nNumChannels	Number of Lptmr channels in the Icu configuration
Icu_Lptmr_GlobalConfigurationType	The Lptmr global configuration paramters
pChannelsConfig	Pointer to the configured channels for Lptmr

3.8.4.10 Structure Icu_Lptmr_ChannelConfigType

Structure that combine configuration of each Lptmr channels in the Icu.

Implements: Icu_Lptmr_ChannelConfigType_struct

Declaration:

```
typedef struct
                     {
                       Icu_Lptmr_ChannelType nHwChannel,
                       uint8 nDefaultStartEdge,
                       uint8 nMeasurementModeType,
                     } Icu_Lptmr_ChannelConfigType;
```

Table 3-121. Structure Icu_Lptmr_ChannelConfigType member description

Member	Description
nHwChannel	Assigned Lptmr channel id.
nDefaultStartEdge	Lptmr Default Start Edge.
nMeasurementModeType	Lptmr MeasurementMode.

3.8.4.11 Structure Icu_LPit_IpConfigType

Structure that combine Lpit IP specific configuration structure type.

Implements: Icu_LPit_IpConfigType_struct

Declaration:

```
typedef struct
                    {
                      Icu_LPit_ChannelType nNumChannels,
                      Icu_LPit_ChannelConfigType (*pChannelsConfig)[]
                    } Icu_LPit_IpConfigType;
```

Table 3-122. Structure Icu_LPit_IpConfigType member description

Member	Description
nNumChannels	Number of Lpit channels in the Icu configuration
pChannelsConfig	Pointer to the configured channels for Lpit

3.8.4.12 Structure Icu_LPit_ChannelConfigType

Structure that combine configuration of each Lpit channels in the Icu.

Implements: Icu_LPit_ChannelConfigType_struct

Declaration:

```
typedef struct
                    {
                      Icu_LPit_ChannelType nHwChannel,
                      Icu_LPit_ControlType nControlValue
                    } Icu_LPit_ChannelConfigType;
```

Table 3-123. Structure Icu_LPit_ChannelConfigType member description

Member	Description
nHwChannel	Assigned Lpit channel id.
nControlValue	Lpit channel parameters.

3.8.4.13 Structure Icu_DutyCycleType

Structure that contains ICU Duty cycle parameters. It contains the values needed for calculating duty cycles i.e Period time value and active time value.

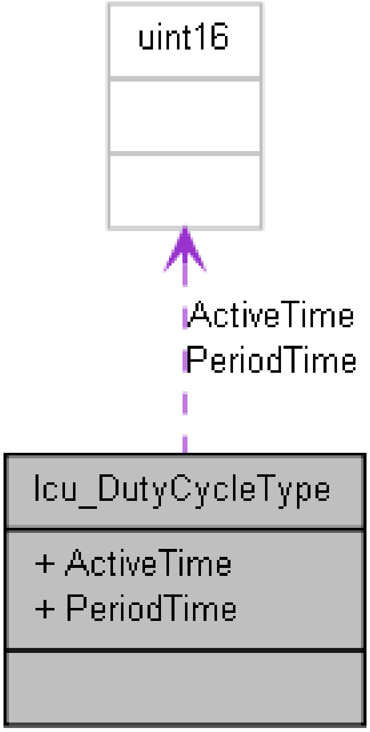


Figure 3-3. Struct Icu_DutyCycleType

Implements: Icu_DutyCycleType_struct

Declaration:

```
typedef struct
                        {
                          Icu_ValueType ActiveTime,
                          Icu_ValueType PeriodTime
                        } Icu_DutyCycleType;
```

Table 3-124. Structure Icu_DutyCycleType member description

Member	Description
ActiveTime	Low or High time value
PeriodTime	Period time value

3.8.5 Types Reference

Types supported by the driver are as per AUTOSAR ICU Driver software specification Version 4.2 Rev0002 .

3.8.5.1 Typedef Icu_ChannelType

This gives the numeric ID (hardware channel number) of an ICU channel.

Implements: Icu_ChannelType_typedef

Type: uint16

3.8.5.2 Typedef Icu_EdgeNumberType

Type for saving hardware specific edge number.

Implements: Icu_EdgeNumberType_typedef

Type:`Icu_HwSpecificEdgeNumberType`

3.8.5.3 Typedef Icu_IndexType

Type for saving the ICU Hardware specific index.

Implements: Icu_IndexType_typedef

Type:`Icu_HwSpecificIndexType`

3.8.5.4 Typedef Icu_NotifyType

The notification functions shall have no parameters and no return value.

Implements: Icu_NotifyType_typedef

Type: void(*

3.8.5.5 Typedef Icu_ParamType

The Icu_ParamType contains combined bit fields for initialization options, for different registers.

Implements: Icu_ParamType_typedef

Type: uint32

3.8.5.6 Typedef Icu_ValueType

Type for saving the timer register width value.

Implements: Icu_ValueType_typedef

Type: Icu_TimerRegisterWidthType

3.8.5.7 Typedef Icu_WakeupValueType

Type for saving the Wakeup value.

Implements: Icu_WakeupValueType_typedef

Type: uint32

3.8.5.8 Typedef Icu_MeasurementSubModeType

Type for saving the ICU measurement submode type.

Implements: Icu_MeasurementSubModeType_typedef

Type: uint16

3.8.5.9 Typedef Icu_HwSpecificEdgeNumberType

Type, to abstract the return value of the service Icu_GetEdgeNumbers().

Implements: Icu_HwSpecificEdgeNumberType_typedef

Type:uint16

3.8.5.10 Typedef Icu_HwSpecificIndexType

Type, to abstract the return value of the service Icu_GetTimestampIndex()

Implements: Icu_HwSpecificIndexType_typedef

Type:uint16

3.8.5.11 Typedef Icu_TimerRegisterWidthType

Width of the buffer for timestamp ticks and measured elapsed time ticks

Implements: Icu_TimerRegisterWidthType_typedef

Type:uint32

3.9 Symbolic Names Disclaimer

All containers having the symbolic name tag set as true in the Autosar schema will generate defines like:

#define <Container_Short_Name> <Container_ID>

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

Chapter 4
Tresos Configuration Plug-in

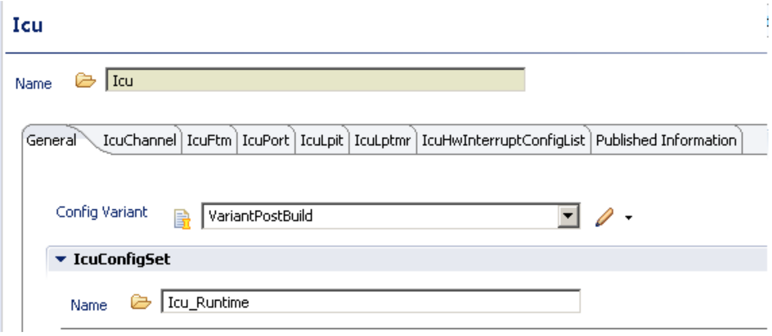
This chapter describes the Tresos configuration plug-in for the ICU Driver. The most of the parameters are described below.

4.1 Configuration elements of Icu

Included forms :

- IMPLEMENTATION_CONFIG_VARIANT
- IcuGeneral
- IcuNonAUTOSAR
- IcuOptionalApis
- CommonPublishedInformation
- IcuConfigSet

4.2 Form IMPLEMENTATION_CONFIG_VARIANT



Icu

Name: Icu

Config Variant: VariantPostBuild

IcuConfigSet

Name: Icu_Runtime

Figure 4-1. Tresos Plugin snapshot for IMPLEMENTATION_CONFIG_VARIANT form.

Table 4-1. Attribute IMPLEMENTATION_CONFIG_VARIANT detailed description

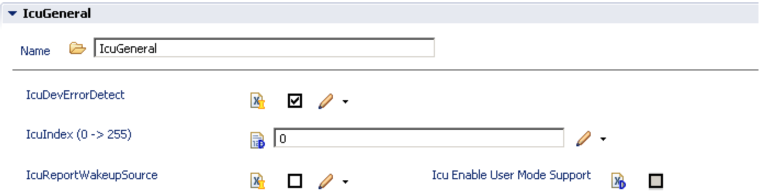
Property	Value
Label	Config Variant

Table continues on the next page...

Table 4-1. Attribute IMPLEMENTATION_CONFIG_VARIANT detailed description (continued)

Property	Value
Default	VariantPostBuild
Range	VariantPostBuild VariantPreCompile

4.3 Form IcuGeneral



IcuGeneral

Name: IcuGeneral

IcuDevErrorDetect ☑

IcuIndex (0 -> 255): 0

IcuReportWakeupSource ☐

Icu Enable User Mode Support ☐

Figure 4-2. Tresos Plugin snapshot for IcuGeneral form.

4.3.1 IcuDevErrorDetect (IcuGeneral)

Switches the Development Error Detection and Notification on or off.

- true: Enabled.
- false: Disabled.

Table 4-2. Attribute IcuDevErrorDetect (IcuGeneral) detailed description

Property	Value
Type	BOOLEAN
Origin	AUTOSAR_ECUC
SCOPE	LOCAL
Symbolic Name	false
Default	true

4.3.2 IcuIndex (IcuGeneral)

Specifies the InstanceId of this module instance.

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

Note

In current implementation this parameter is not used.

Table 4-3. Attribute IcuIndex (IcuGeneral) detailed description

Property	Value
Type	INTEGER
Origin	AUTOSAR_ECUC
SCOPE	LOCAL
Symbolic Name	false
Default	0
Enable	false
Invalid	Range >=0 <=255

4.3.3 IcuReportWakeupSource (IcuGeneral)

Switch for enabling Wakeup source reporting.

- true: Report Wakeup source.
- false: Do not report Wakeup source.

Table 4-4. Attribute IcuReportWakeupSource (IcuGeneral) detailed description

Property	Value
Type	BOOLEAN
Origin	AUTOSAR_ECUC
SCOPE	LOCAL
Symbolic Name	false
Default	true

4.3.4 IcuEnableUserModeSupport (IcuGeneral)

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

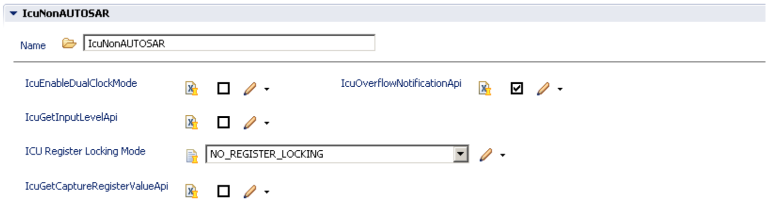
- true: User mode allowed.
- false: User mode allowed not allowed, programing will be supervisor mode

Table 4-5. Attribute IcuEnableUserModeSupport (IcuGeneral) detailed description

Property	Value
Type	BOOLEAN
Symbolic Name	false
Default	false

4.4 Form IcuNonAUTOSAR

Enabling the settings of this section will configure the driver in a mode not compliant with AUTOSAR requirements.



IcuNonAUTOSAR

Name: IcuNonAUTOSAR

IcuEnableDualClockMode ☐

IcuOverflowNotificationApi ☑

IcuGetInputLevelApi ☐

ICU Register Locking Mode: NO_REGISTER_LOCKING

IcuGetCaptureRegisterValueApi ☐

Figure 4-3. Tresos Plugin snapshot for IcuNonAUTOSAR form.

4.4.1 IcuOverflowNotificationApi (IcuNonAUTOSAR)

Add / removes Overflow Notification functionality.

Enabling IcuOverflowNotificationApi overflow events will not be treated as errors and a Notification Handler can be provided.

If this optional API is not enabled, overflow events will trigger DET Report Error.

Note

> Due to hardware implementattion, the Icu Overflow Notification is not syncronous with the event for ICU_MODE_SIGNAL_MEASUREMENT and ICU_MODE_TIMESTAMP modes. The notification will be triggered when measurement completes (for ICU_MODE_SIGNAL_MEASUREMENT) or the next timestamp event occurs (for ICU_MODE_TIMESTAMP).

Table 4-6. Attribute IcuOverflowNotificationApi (IcuNonAUTOSAR) detailed description

Property	Value
Type	BOOLEAN
Origin	Custom
Symbolic Name	false
Default	false

4.4.2 IcuGetInputLevelApi (IcuNonAUTOSAR)

Add / removes Icu_GetInputLevel API from the code.

This function returns Input pin state.

Note

This feature is not required by Autosar.

Table 4-7. Attribute IcuGetInputLevelApi (IcuNonAUTOSAR) detailed description

Property	Value
Origin	Custom
Symbolic Name	false
Default	false

4.4.3 IcuRegisterLockingMode (IcuNonAUTOSAR)

IcuRegisterLockingMode can take following values:

- UserModeSoftLocking: each function writing to hardware registers is responsible for maintaining the register soft locking mechanism; no additional actions needed in the configuration locking mechanism; the register locking management is available in user mode.
- SupervisorModeSoftLocking: the functions writing the hardware registers use register writes without locking; the configuration locking/unlocking functions are responsible for applying/releasing the register soft locking in supervisor mode only.

Table 4-8. Attribute IcuRegisterLockingMode (IcuNonAUTOSAR) detailed description

Property	Value
Label	ICU Register Locking Mode
Origin	Custom

Table continues on the next page...

Table 4-8. Attribute IcuRegisterLockingMode (IcuNonAUTOSAR) detailed description (continued)

Property	Value
Symbolic Name	false
Default	NO_REGISTER_LOCKING
Range	NO_REGISTER_LOCKING

4.4.4 IcuEnableDualClockMode (IcuNonAUTOSAR)

Add / removes Icu_SetClockMode API from the code.

This function allows configuration of the dual clock mode.

Note

This feature is not required by Autosar.

Table 4-9. Attribute IcuEnableDualClockMode (IcuNonAUTOSAR) detailed description

Property	Value
Origin	Custom
Symbolic Name	false
Default	false

4.4.5 IcuGetCaptureRegisterValueApi (IcuNonAUTOSAR)

Adds / removes service Icu_GetCaptureRegisterValue from the code. This function returns value of Capture register for the mesurement channel or timestamp mode channel which is called by the user. It's enabled when IcuTimestampApi or IcuSignalMeasurementApi is true.

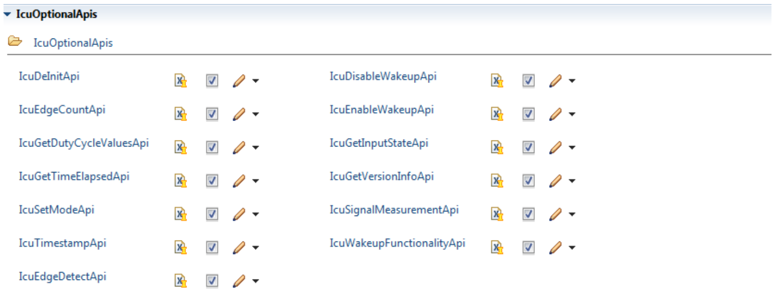
Note

This feature is not required by Autosar.

Table 4-10. Attribute IcuGetCaptureRegisterValueApi (IcuNonAUTOSAR) detailed description

Property	Value
Origin	Custom
Symbolic Name	false
Default	false

4.5 Form IcuOptionalApis



IcuOptionalApis

IcuOptionalApis

IcuDeInitApi	☑	IcuDisableWakeupApi	☑
IcuEdgeCountApi	☑	IcuEnableWakeupApi	☑
IcuGetDutyCycleValuesApi	☑	IcuGetInputStateApi	☑
IcuGetTimeElapsedApi	☑	IcuGetVersionInfoApi	☑
IcuSetModeApi	☑	IcuSignalMeasurementApi	☑
IcuTimestampApi	☑	IcuWakeupFunctionalityApi	☑
IcuEdgeDetectApi	☑		

Figure 4-4. Tresos Plugin snapshot for IcuOptionalApis form.

4.5.1 IcuDeInitApi (IcuOptionalApis)

Adds / removes the service Icu_DeInit() from the code.

- true: Icu_DeInit() can be used.
- false: Icu_DeInit() can not be used.

Table 4-11. Attribute IcuDeInitApi (IcuOptionalApis) detailed description

Property	Value
Type	BOOLEAN
Origin	AUTOSAR_ECUC
SCOPE	LOCAL
Symbolic Name	false
Default	true

4.5.2 IcuDisableWakeupApi (IcuOptionalApis)

Adds / removes the service Icu_DisableWakeup() from the code.

- true: Icu_DisableWakeup() can be used.
- false: Icu_DisableWakeup() can not be used.

Table 4-12. Attribute IcuDisableWakeupApi (IcuOptionalApis) detailed description

Property	Value
Type	BOOLEAN
Origin	AUTOSAR_ECUC
SCOPE	LOCAL
Symbolic Name	false
Default	true

4.5.3 IcuEdgeCountApi (IcuOptionalApis)

Adds / removes all services related to the edge counting functionality as listed below, from the code: Icu_ResetEdgeCount(), Icu_EnableEdgeCount(), Icu_DisableEdgeCount(), Icu_GetEdgeNumbers().

- true: The services listed above can be used.
- false: The services listed above can not be used.

Table 4-13. Attribute IcuEdgeCountApi (IcuOptionalApis) detailed description

Property	Value
Type	BOOLEAN
Origin	AUTOSAR_ECUC
SCOPE	LOCAL
Symbolic Name	false
Default	true

4.5.4 IcuEnableWakeupApi (IcuOptionalApis)

Adds / removes the service Icu_EnableWakeup() from the code.

- true: Icu_EnableWakeup() can be used.
- false: Icu_EnableWakeup() can not be used.

Table 4-14. Attribute IcuEnableWakeupApi (IcuOptionalApis) detailed description

Property	Value
Type	BOOLEAN
Origin	AUTOSAR_ECUC
SCOPE	LOCAL
Symbolic Name	false

Table continues on the next page...

Table 4-14. Attribute IcuEnableWakeupApi (IcuOptionalApis) detailed description (continued)

Property	Value
Default	true

4.5.5 IcuGetDutyCycleValuesApi (IcuOptionalApis)

Adds / removes the service Icu_GetDutyCycleValues() from the code.

- true: Icu_GetDutyCycleValues() can be used.
- false: Icu_GetDutyCycleValues() can not be used.

Note

If IcuSignalMeasurementApi == OFF this switch is shall also be set to OFF.

Table 4-15. Attribute IcuGetDutyCycleValuesApi (IcuOptionalApis) detailed description

Property	Value
Type	BOOLEAN
Origin	AUTOSAR_ECUC
SCOPE	LOCAL
Symbolic Name	false
Default	true

4.5.6 IcuGetInputStateApi (IcuOptionalApis)

Adds / removes the service Icu_GetInputState() from the code.

- true: Icu_GetInputState() can be used.
- false: Icu_GetInputState() can not be used.

Table 4-16. Attribute IcuGetInputStateApi (IcuOptionalApis) detailed description

Property	Value
Type	BOOLEAN
Origin	AUTOSAR_ECUC
SCOPE	LOCAL
Symbolic Name	false
Default	true

4.5.7 IcuGetTimeElapsedApi (IcuOptionalApis)

Adds / removes the service Icu_GetTimeElapsed() from the code.

- true: Icu_GetTimeElapsed() can be used.
- false: Icu_GetTimeElapsed() can not be used.

Note

If IcuSignalMeasurementApi == OFF this switch is shall also be set to OFF.

Table 4-17. Attribute IcuGetTimeElapsedApi (IcuOptionalApis) detailed description

Property	Value
Type	BOOLEAN
Origin	AUTOSAR_ECUC
SCOPE	LOCAL
Symbolic Name	false
Default	true

4.5.8 IcuGetVersionInfoApi (IcuOptionalApis)

Adds / removes the service Icu_GetVersionInfo() from the code.

- true: Icu_GetVersionInfo() can be used.
- false: Icu_GetVersionInfo() can not be used.

Table 4-18. Attribute IcuGetVersionInfoApi (IcuOptionalApis) detailed description

Property	Value
Type	BOOLEAN
Origin	AUTOSAR_ECUC
SCOPE	LOCAL
Symbolic Name	false
Default	true

4.5.9 IcuSetModeApi (IcuOptionalApis)

Adds / removes the service Icu_SetMode() from the code.

- true: Icu_SetMode() can be used.
- false: Icu_SetMode() can not be used.

Table 4-19. Attribute IcuSetModeApi (IcuOptionalApis) detailed description

Property	Value
Type	BOOLEAN
Origin	AUTOSAR_ECUC
SCOPE	LOCAL
Symbolic Name	false
Default	true

4.5.10 IcuSignalMeasurementApi (IcuOptionalApis)

Adds / removes the services Icu_StartSignalMeasurement() and Icu_StopSignalMeasurement() from the code.

- true: Icu_StartSignalMeasurement() and Icu_StopSignalMeasurement() can be used.
- false: Icu_StartSignalMeasurement() and Icu_StopSignalMeasurement() can not be used.

Table 4-20. Attribute IcuSignalMeasurementApi (IcuOptionalApis) detailed description

Property	Value
Type	BOOLEAN
Origin	AUTOSAR_ECUC
SCOPE	LOCAL
Symbolic Name	false
Default	true

4.5.11 IcuTimestampApi (IcuOptionalApis)

Adds / removes all services related to the timestamping functionality as listed below from the code: Icu_StartTimestamp(), Icu_StopTimestamp(), Icu_GetTimestampIndex().

- true: The services listed above can be used.
- false: The services listed above can not be used.

Table 4-21. Attribute IcuTimestampApi (IcuOptionalApis) detailed description

Property	Value
Type	BOOLEAN
Origin	AUTOSAR_ECUC
SCOPE	LOCAL
Symbolic Name	false
Default	true

4.5.12 IcuWakeupFunctionalityApi (IcuOptionalApis)

Adds / removes the service Icu_CheckWakeup() from the code.

- true: Icu_CheckWakeup() can be used.
- false: Icu_CheckWakeup() can not be used.

Table 4-22. Attribute IcuWakeupFunctionalityApi (IcuOptionalApis) detailed description

Property	Value
Type	BOOLEAN
Origin	AUTOSAR_ECUC
SCOPE	LOCAL
Symbolic Name	false
Default	true

4.5.13 IcuEdgeDetectApi (IcuOptionalApis)

Adds / removes the services Icu_EnableEdgeDetection() and Icu_DisableEdgeDetection() from the code.

- true: Icu_EnableEdgeDetection() and Icu_DisableEdgeDetection() can be used.
- false: Icu_EnableEdgeDetection() and Icu_DisableEdgeDetection() can not be used.

Table 4-23. Attribute IcuEdgeDetectApi (IcuOptionalApis) detailed description

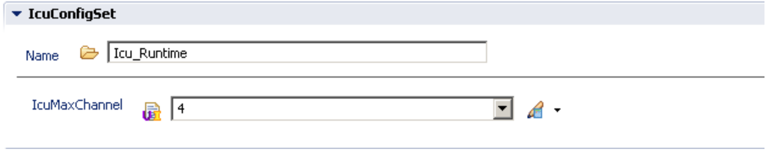
Property	Value
Type	BOOLEAN
Origin	AUTOSAR_ECUC
SCOPE	LOCAL
Symbolic Name	false
Default	true

4.6 Form IcuConfigSet

This container is the base for a multiple configuration set

Included forms :

- Form IcuChannel



Name Icu_Runtime

IcuMaxChannel 4

Figure 4-5. Tresos Plugin snapshot for IcuConfigSet form.

4.6.1 IcuMaxChannel (IcuConfigSet)

The value for the IcuMaxChannel must match with the number of IcuChannel configured For calculating the correct value use the CALC button.

Note

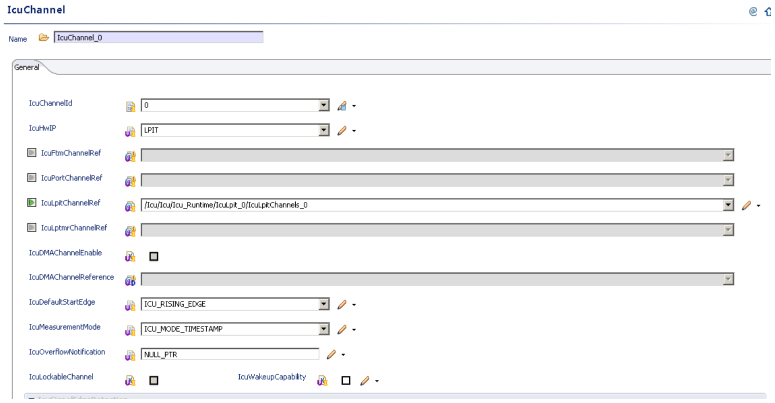
: Total number of configured channels should be same across all IcuConfigSets.

Table 4-24. Attribute IcuMaxChannel (IcuConfigSet) detailed description

Property	Value
Type	INTEGER
POSTBUILDVARIANTVALUE	true
Origin	AUTOSAR_ECUC
SCOPE	LOCAL
Symbolic Name	false
Invalid	Range >=1 <=126

4.6.2 Form IcuChannel

Is included by form : Form IcuConfigSet



Name: IcuChannel_0

IcuChannelId: 0

IcuHwIP: LPIT

□ IcuFtmChannelRef

□ IcuPortChannelRef

☑ IcuLpitChannelRef: /Icu/Icu/Icu_Runtime/IcuLpit_0/IcuLpitChannels_0

□ IcuLptmrChannelRef

IcuDMAChannelEnable: □

IcuDMAChannelReference

IcuDefaultStartEdge: ICU_RISING_EDGE

IcuMeasurementMode: ICU_MODE_TIMESTAMP

IcuOverflowNotification: NULL_PTR

IcuLockableChannel: □ IcuWakeupCapability: □

Figure 4-6. Tresos Plugin snapshot for IcuChannel form.

4.6.2.1 IcuChannelId (IcuChannel)

Channel Id of the ICU channel. This value will be assigned to the symbolic name derived of the IcuChannel container short name.

Table 4-25. Attribute IcuChannelId (IcuChannel) detailed description

Property	Value
Type	INTEGER
Origin	AUTOSAR_ECUC
SCOPE	ECU
Symbolic Name	true
Invalid	Range >=0 <126

4.6.2.2 IcuHwIP (IcuChannel)

Selects the physical IcuHwIP Channel.

Table 4-26. Attribute IcuHwIP (IcuChannel) detailed description

Property	Value
Type	ENUMERATION
POSTBUILDVARIANTVALUE	true
Origin	Custom
DEFAULT	FTM
RANGE	FTM PORT LPTMR LPIT

4.6.2.3 IcuFtmChannelRef (IcuChannel)

Select the FTM channel on which the functionality of the current ICU channel will be implemented

Reference to the FTM Channel.

Table 4-27. Attribute IcuFtmChannelRef (IcuChannel) detailed description

Property	Value
Type	REFERENCE
POSTBUILDVARIANTVALUE	true
OPTIONAL	true
Origin	Custom

4.6.2.4 IcuPortChannelRef (IcuChannel)

Select the PORT_CI channel on which the functionality of the current ICU channel will be implemented

Reference to the PORT_CI Channel.

Table 4-28. Attribute IcuPortChannelRef (IcuChannel) detailed description

Property	Value
Type	REFERENCE
POSTBUILDVARIANTVALUE	true
OPTIONAL	true
Origin	Custom

4.6.2.5 IcuLpitChannelRef (IcuChannel)

Select the LPIT channel on which the functionality of the current ICU channel will be implemented

Reference to the LPIT Channel.

Table 4-29. Attribute IcuLpitChannelRef (IcuChannel) detailed description

Property	Value
Type	REFERENCE
POSTBUILDVARIANTVALUE	true
OPTIONAL	true
Origin	Custom

4.6.2.6 IcuLptmrChannelRef (IcuChannel)

Select the LPTMR channel on which the functionality of the current ICU channel will be implemented

Reference to the LPTMR Channel.

Table 4-30. Attribute IcuLptmrChannelRef (IcuChannel) detailed description

Property	Value
Type	REFERENCE
POSTBUILDVARIANTVALUE	true
OPTIONAL	true
Origin	Custom

4.6.2.7 IcuDMAChannelEnable (IcuChannel)

IcuLockableChannel

indicates if the corresponding channel will use DMA for measurement

Table 4-31. Attribute IcuDMAChannelEnable (IcuChannel) detailed description

Property	Value
Type	BOOLEAN
POSTBUILDVARIANTVALUE	true
Origin	Custom
Symbolic Name	false
Default	false

4.6.2.8 IcuDefaultStartEdge (IcuChannel)

Configures the default-activation-edge which shall be used for this channel

if there was no activation-edge configured by the call of service Icu_SetActivationCondition().

In case the Measurement Mode is "IcuSignalMeasurement" and the properties "DutyCycle" or "Period" are set, the edge configured here is used as Default Period Start Edge.

Implementation Type: Icu_ActivationType

Table 4-32. Attribute IcuDefaultStartEdge (IcuChannel) detailed description

Property	Value
Type	ENUMERATION
SCOPE	LOCAL
POSTBUILDVARIANTVALUE	true
Origin	AUTOSAR_ECUC
Symbolic Name	false
Default	ICU_RISING_EDGE
Range	ICU_BOTH_EDGES ICU_FALLING_EDGE ICU_RISING_EDGE

4.6.2.9 IcuMeasurementMode (IcuChannel)

Configures the measurement mode of this channel.

User should enable optional parameters with respect to the selected IcuMeasurementMode.

Implementation Type: Icu_MeasurementModeType

Table 4-33. Attribute IcuMeasurementMode (IcuChannel) detailed description

Property	Value
Type	ENUMERATION
Origin	AUTOSAR_ECUC
SCOPE	LOCAL
POSTBUILDVARIANTVALUE	true
Symbolic Name	false
Default	ICU_MODE_SIGNAL_EDGE_DETECT

Table continues on the next page...

Table 4-33. Attribute IcuMeasurementMode (IcuChannel) detailed description (continued)

Property	Value
Range	ICU_MODE_EDGE_COUNTER ICU_MODE_SIGNAL_EDGE_DETECT ICU_MODE_SIGNAL_MEASUREMENT ICU_MODE_TIMESTAMP

4.6.2.10 IcuOverflowNotification (IcuChannel)

Icu Overflow Notification Handler

In order to activate this field you have to:

- enable IcuOverflowNotificationApi,
- choose one of the modes:
  - ICU_MODE_EDGE_COUNTER,
  - ICU_MODE_SIGNAL_MEASUREMENT,
  - ICU_MODE_TIMESTAMP
- to enable overflow detection on the internal counter

Note

Due to hardware implementatttion, the Icu Overflow Notification is not syncronous with the event for ICU_MODE_SIGNAL_MEASUREMENT and ICU_MODE_TIMESTAMP modes.

The notification will be triggered when measurement completes (for ICU_MODE_SIGNAL_MEASUREMENT) or the next timestamp event occurs (for ICU_MODE_TIMESTAMP).

Table 4-34. Attribute IcuOverflowNotification (IcuChannel) detailed description

Property	Value
Type	FUNCTION-NAME
POSTBUILDVARIANTVALUE	true
Origin	Custom
Symbolic Name	false
Default	NULL_PTR

4.6.2.11 IcuLockableChannel (IcuChannel)

IcuLockableChannel

indicates if the corresponding channel shall be the object of the configuration locking mechanism.

Table 4-35. Attribute IcuLockableChannel (IcuChannel) detailed description

Property	Value
Type	BOOLEAN
POSTBUILDVARIANTVALUE	true
Origin	Custom
Symbolic Name	false
Default	false

4.6.2.12 IcuWakeupCapability (IcuChannel)

Information about the wakeup-capability of this channel.

- true: Channel is wakeup capable.
- false: Channel is not wakeup capable.

Table 4-36. Attribute IcuWakeupCapability (IcuChannel) detailed description

Property	Value
Type	BOOLEAN
POSTBUILDVARIANTVALUE	true
SCOPE	LOCAL
Origin	AUTOSAR_ECUC
Symbolic Name	false
Default	false

4.6.2.13 IcuDMAChannelRef (IcuChannel)

Icu DMA Channel Reference

Reference to the DMA Channel configure for the Request

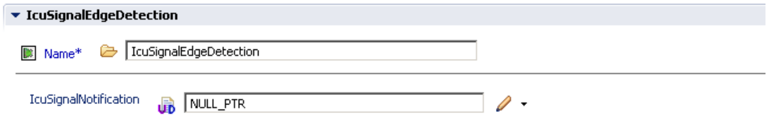
Table 4-37. Attribute IcuDMAChannelRef (IcuChannel) detailed description

Property	Value
Label	IcuDMAChannelReference
Type	CHOICE-REFERENCE
POSTBUILDVARIANTVALUE	true
Origin	Custom

4.6.2.14 Form IcuSignalEdgeDetection

This container contains the configuration (parameters) in case the measurement mode is "IcuSignalEdgeDetection"

Is included by form : Form IcuChannel



IcuSignalEdgeDetection

Name* IcuSignalEdgeDetection

IcuSignalNotification NULL_PTR

Figure 4-7. Tresos Plugin snapshot for IcuSignalEdgeDetection form.

4.6.2.14.1 IcuSignalNotification (IcuSignalEdgeDetection)

Notification function for signal notification.

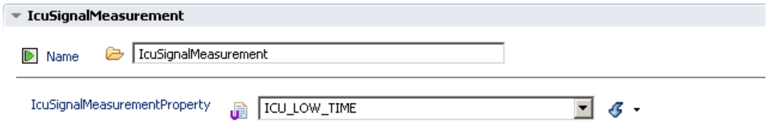
Table 4-38. Attribute IcuSignalNotification (IcuSignalEdgeDetection) detailed description

Property	Value
Type	FUNCTION-NAME
OPTIONAL	true
POSTBUILDVARIANTVALUE	true
POSTBUILDVARIANTMULTIPLICITY	true
SCOPE	LOCAL
Origin	AUTOSAR_ECUC
Symbolic Name	false
Default	NULL_PTR

4.6.2.15 Form IcuSignalMeasurement

This container contains the configuration (parameters) in case the measurement mode is "IcuSignalMeasurement"

Is included by form : Form IcuChannel



IcuSignalMeasurement

Name: IcuSignalMeasurement

IcuSignalMeasurementProperty: ICU_LOW_TIME

Figure 4-8. Tresos Plugin snapshot for IcuSignalMeasurement form.

4.6.2.15.1 IcuSignalMeasurementProperty (IcuSignalMeasurement)

Configures the property that could be measured in case the mode is "IcuSignalMeasurement".

This property can not be changed during runtime.

Followings are measurement mode

- ICU_DUTY_CYCLE
- ICU_HIGH_TIME
- ICU_LOW_TIME
- ICU_PERIOD_TIME

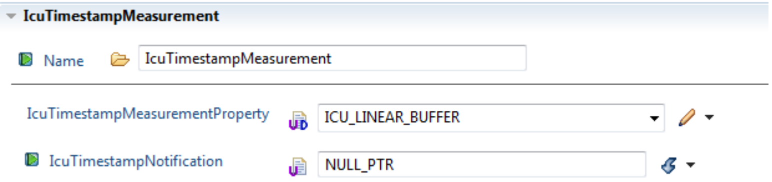
Implementation Type: Icu_SignalMeasurementPropertyType

Table 4-39. Attribute IcuSignalMeasurementProperty (IcuSignalMeasurement) detailed description

Property	Value
Type	ENUMERATION
SCOPE	LOCAL
POSTBUILDVARIANTVALUE	true
Origin	AUTOSAR_ECUC
Symbolic Name	false
Default	ICU_DUTY_CYCLE
Range	ICU_DUTY_CYCLE ICU_HIGH_TIME ICU_LOW_TIME ICU_PERIOD_TIME

4.6.2.16 Form IcuTimestampMeasurement

Is included by form : Form IcuChannel



IcuTimestampMeasurement

Name: IcuTimestampMeasurement

IcuTimestampMeasurementProperty: ICU_LINEAR_BUFFER

IcuTimestampNotification: NULL_PTR

Figure 4-9. Tresos Plugin snapshot for IcuTimestampMeasurement form.

4.6.2.16.1 IcuTimestampMeasurementProperty (IcuTimestampMeasurement)

Configures the handling of the buffer in case the mode is "Timestamp"

- ICU_CIRCULAR_BUFFER.
- ICU_LINEAR_BUFFER.

Implementation Type: Icu_TimestampBufferType

Table 4-40. Attribute IcuTimestampMeasurementProperty (IcuTimestampMeasurement) detailed description

Property	Value
Type	ENUMERATION
Origin	AUTOSAR_ECUC
SCOPE	LOCAL
POSTBUILDVARIANTVALUE	true
Symbolic Name	false
Default	ICU_LINEAR_BUFFER
Range	ICU_CIRCULAR_BUFFER ICU_LINEAR_BUFFER

4.6.2.16.2 IcuTimestampNotification (IcuTimestampMeasurement)

Notification function if the number of requested timestamps(Notification interval > 0) are acquired.

Table 4-41. Attribute IcuTimestampNotification (IcuTimestampMeasurement) detailed description

Property	Value
Type	FUNCTION-NAME
OPTIONAL	true
POSTBUILDVARIANTMULTIPLICITY	true
Origin	AUTOSAR_ECUC

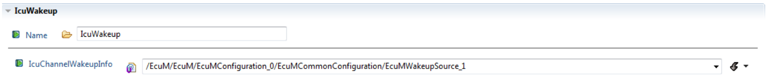
Table continues on the next page...

Table 4-41. Attribute IcuTimestampNotification (IcuTimestampMeasurement) detailed description (continued)

Property	Value
SCOPE	LOCAL
POSTBUILDVARIANTVALUE	true
Symbolic Name	false
Default	NULL_PTR

4.6.2.17 Form IcuWakeup

Is included by form : Form IcuChannel



IcuWakeup

Name: IcuWakeup

IcuChannelWakeupInfo: /EcuM/EcuM/EcuMConfiguration_0/EcuMCommonConfiguration/EcuMWakeupSource_1

Figure 4-10. Tresos Plugin snapshot for IcuWakeup form.

4.6.2.17.1 IcuChannelWakeupInfo (IcuWakeup)

If the wakeup-capability is true the wakeup source referenced is transmitted to the ECU State Manager (EcuM) .

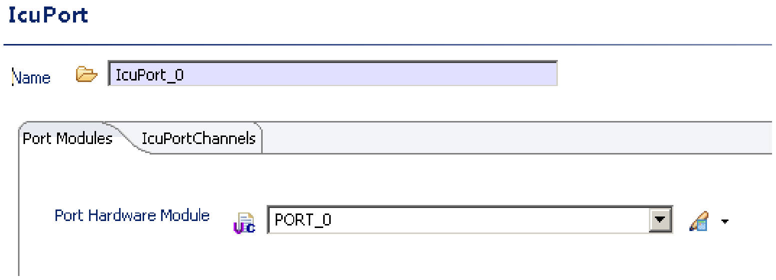
Implementation Type: reference to EcuM_WakeupSourceType

Table 4-42. Attribute IcuChannelWakeupInfo (IcuWakeup) detailed description

Property	Value
Type	SYMBOLIC-NAME-REFERENCE
OPTIONAL	true
POSTBUILDVARIANTMULTIPLICITY	true
Origin	AUTOSAR_ECUC
SCOPE	LOCAL
POSTBUILDVARIANTVALUE	true

4.6.3 Form IcuPortCl

Is included by form : Form IcuConfigSet


IcuPort

Name: IcuPort_0

Port Hardware Module: PORT_0

Figure 4-11. Tresos Plugin snapshot for IcuPortCI form.

4.6.3.1 IcuPortModule (IcuPortCI)

Select the physical PORT_CI Module.

Note

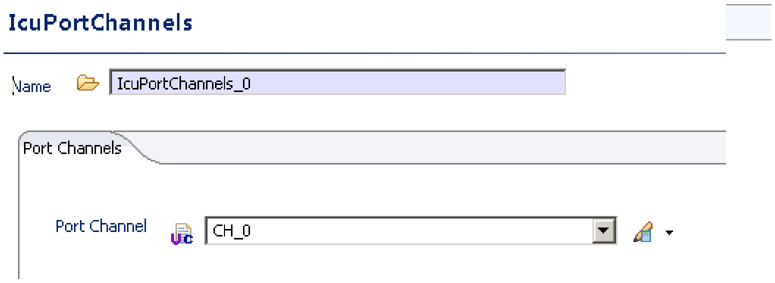
: This is an Implementation Specific Parameter.

Table 4-43. Attribute IcuPortModule (IcuPortCI) detailed description

Property	Value
LABEL	Port Hardware Module
Type	ENUMERATION
POSTBUILDVARIANTVALUE	true
Origin	Custom
Symbolic Name	false
Range	PORT_0 PORT_1 PORT_2 PORT_3 PORT_4

4.6.3.2 Form IcuPortCIChannel

Is included by form : Form IcuPortCI



IcuPortChannels

Name: IcuPortChannels_0

Port Channel: CH_0

Figure 4-12. Tresos Plugin snapshot for IcuPortClChannel form.

4.6.3.2.1 IcuPortChannel (IcuPortClChannel)

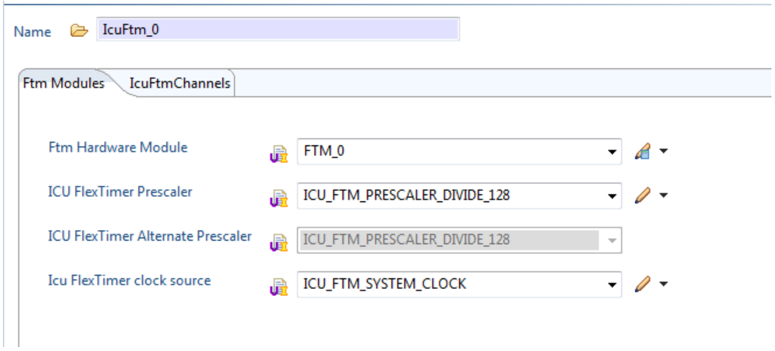
Selects one of the PORT_CI channels available on the platform.

Table 4-44. Attribute IcuPortChannel (IcuPortClChannel) detailed description

Property	Value
LABEL	Port Channel
Type	ENUMERATION
Origin	Custom
POSTBUILDVARIANTVALUE	true
Symbolic Name	false

4.6.4 Form IcuFtm

Is included by form : Form IcuConfigSet



Name: IcuFtm_0

Ftm Hardware Module: FTM_0

ICU FlexTimer Prescaler: ICU_FTM_PRESCALER_DIVIDE_128

ICU FlexTimer Alternate Prescaler: ICU_FTM_PRESCALER_DIVIDE_128

Icu FlexTimer clock source: ICU_FTM_SYSTEM_CLOCK

Figure 4-13. Tresos Plugin snapshot for IcuFtm form.

4.6.4.1 IcuFtmModule (IcuFtm)

Select the physical Ftm Module.

Table 4-45. Attribute IcuFtmModule(IcuFtm) detailed description

Property	Value
Label	Ftm Hardware Module
Type	ENUMERATION
POSTBUILDVARIANTVALUE	true
Origin	Custom
Symbolic Name	false
Range	FTM_0 FTM_1 FTM_2 FTM_3

4.6.4.2 Icu_FlexTimer_Prescaler(IcuFtm)

If an FTM channel from module is being used, this configures the prescaler value for the specific channel.

Table 4-46. Attribute Icu_FlexTimer_Prescaler(IcuFtm) detailed description

Property	Value
Type	ENUMERATION
POSTBUILDVARIANTVALUE	true
Origin	Custom
Symbolic Name	false
DEFAULT	ICU_FTM_PRESCALER_DIVIDE_128
Range	ICU_FTM_PRESCALER_DIVIDE_1 ICU_FTM_PRESCALER_DIVIDE_2 ICU_FTM_PRESCALER_DIVIDE_4 ICU_FTM_PRESCALER_DIVIDE_8 ICU_FTM_PRESCALER_DIVIDE_16 ICU_FTM_PRESCALER_DIVIDE_32 ICU_FTM_PRESCALER_DIVIDE_64 ICU_FTM_PRESCALER_DIVIDE_128

4.6.4.3 Icu_FlexTimer_Prescaler_Alternate(IcuFtm)

If an FTM channel from module is being used, this configures the alternate prescaler value for the specific channel.

This can be set when calling Icu_SetClockMode(ICU_ALTERNATE_CLOCK_MODE)

Table 4-47. Attribute Icu_FlexTimer_Prescaler_Alternate(IcuFtm) detailed description

Property	Value
Type	ENUMERATION
Origin	Custom
POSTBUILDVARIANTVALUE	true
Symbolic Name	false
DEFAULT	ICU_FTM_PRESCALER_DIVIDE_128
Range	ICU_FTM_PRESCALER_DIVIDE_1 ICU_FTM_PRESCALER_DIVIDE_2 ICU_FTM_PRESCALER_DIVIDE_4 ICU_FTM_PRESCALER_DIVIDE_8 ICU_FTM_PRESCALER_DIVIDE_16 ICU_FTM_PRESCALER_DIVIDE_32 ICU_FTM_PRESCALER_DIVIDE_64 ICU_FTM_PRESCALER_DIVIDE_128

4.6.4.4 Icu_FlexTimer_ClockSource(IcuFtm)

Number of consecutive samples that must agree prior to the input filter accepting an input transition.

Note

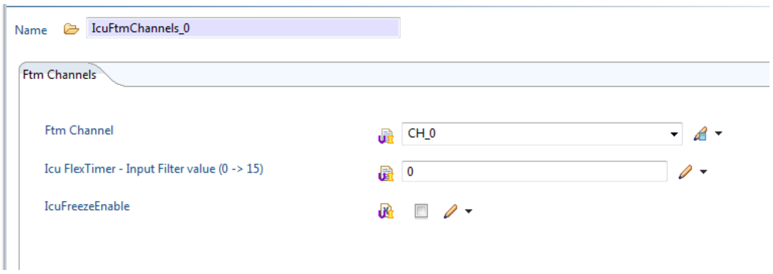
This is an Implementation Specific Parameter.

Table 4-48. Attribute Icu_FlexTimer_ClockSource(IcuFtm) detailed description

Property	Value
Type	ENUMERATION
POSTBUILDVARIANTVALUE	true
Origin	Custom
Symbolic Name	false
DEFAULT	ICU_FTM_SYSTEM_CLOCK
Range	ICU_FTM_SYSTEM_CLOCK ICU_FTM_EXTERNAL_CLOCK ICU_FTM_FIXED_FREQUENCY_CLOCK

4.6.4.5 Form IcuFtmChannel

Is included by form : Form IcuFtm



Name IcuFtmChannels_0

Ftm Channels

Ftm Channel	CH_0
Icu FlexTimer - Input Filter value (0 -> 15)	0
IcuFreezeEnable	□

Figure 4-14. Tresos Plugin snapshot for IcuFtmChannel form.

4.6.4.5.1 IcuFtmChannel (IcuFtmChannel)

Selects one of the Ftm channels available on the platform.

Table 4-49. Attribute IcuFtmChannel (IcuFtmChannel) detailed description

Property	Value
Type	ENUMERATION
POSTBUILDVARIANTVALUE	true
Origin	Custom
Symbolic Name	false
Default	CH_0
Range	CH_0 CH_1 CH_2 CH_3 CH_4 CH_5 CH_6 CH_7

4.6.4.5.2 Icu_FlexTimerFilter (IcuFtmChannel)

Input Capture Filter Control: Selects the filter value for the channel input. The filter is disabled when the value is zero.

Note

This is an Implementation Specific Parameter.

Table 4-50. Attribute Icu_FlexTimerFilter(IcuFtmChannel) detailed description

Property	Value
Label	Icu FlexTimer - Input Filter value
Type	INTEGER
POSTBUILDVARIANTVALUE	true

Table continues on the next page...

Table 4-50. Attribute Icu_FlexTimerFilter(IcuFtmChannel) detailed description (continued)

Property	Value
Origin	Custom
Symbolic Name	false
Default	0
Invalid	Range >0 <=15

4.6.4.5.3 IcuFreezeEnable (IcuFtmChannel)

Vendor specific: Select to set hardware freeze enable during debug mode.

On FlexTimer channels the freeze option must be the same for all channels belonging to the same module.

On FTM,PIT,STM base platform.If the chip is in BDM mode and A5 core configure is set then counter stop. That is hardware restriction.

Note

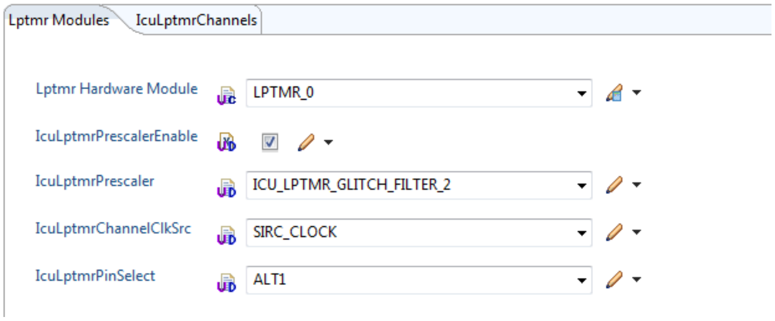
This is an Implementation Specific Parameter.

Table 4-51. Attribute IcuFreezeEnable (IcuFtmChannel) detailed description

Property	Value
Type	BOOLEAN
Origin	Custom
POSTBUILDVARIANTVALUE	true
Symbolic Name	false
Default	false

4.6.5 Form IcuLptmr

Is included by form : Form IcuConfigSet



Lptmr Hardware Module: LPTMR_0

IcuLptmrPrescalerEnable: ☑

IcuLptmrPrescaler: ICU_LPTMR_GLITCH_FILTER_2

IcuLptmrChannelClkSrc: SIRC_CLOCK

IcuLptmrPinSelect: ALT1

Figure 4-15. Tresos Plugin snapshot for IcuLptmr form.

4.6.5.1 IcuLptmrModule (IcuLptmr)

Select the physical Lptmr Module.

Note

: This is an Implementation Specific Parameter.

Table 4-52. Attribute IcuLptmrModule (IcuLptmr) detailed description

Property	Value
LABEL	Lptmr Hardware Module
Type	ENUMERATION
POSTBUILDVARIANTVALUE	true
Origin	Custom
Symbolic Name	false
Range	LPTMR_0

4.6.5.2 PrescalerEnable (IcuLptmr)

When PBYP is set, the selected prescaler clock in Time Counter mode or selected input source in Pulse Counter mode directly clocks the CNR.

Note

: This is an Implementation Specific Parameter.

Table 4-53. Attribute PrescalerEnable (IcuLptmr) detailed description

Property	Value
LABEL	IcuLptmrPrescalerEnable
Type	BOOLEAN

Table continues on the next page...

Table 4-53. Attribute PrescalerEnable (IcuLptmr) detailed description (continued)

Property	Value
POSTBUILDVARIANTVALUE	true
Origin	Custom
Symbolic Name	false
DEFAULT	true

4.6.5.3 IcuLptmrPrescaler (IcuLptmr)

Vendor specific: The ICU module specific Glitch filter value. Configures the Glitch filter in Pulse Counter mode.

- 1 - Glitch filter recognizes change on input pin after 2 rising clock edges
- 2 - Glitch filter recognizes change on input pin after 4 rising clock edges
- 3 - Glitch filter recognizes change on input pin after 8 rising clock edges
- 4 - Glitch filter recognizes change on input pin after 16 rising clock edges
- 5 - Glitch filter recognizes change on input pin after 32 rising clock edges
- 6 - Glitch filter recognizes change on input pin after 64 rising clock edges
- 7 - Glitch filter recognizes change on input pin after 128 rising clock edges
- 8 - Glitch filter recognizes change on input pin after 256 rising clock edges
- 9 - Glitch filter recognizes change on input pin after 512 rising clock edges
- 10 - Glitch filter recognizes change on input pin after 1024 rising clock edges
- 11 - Glitch filter recognizes change on input pin after 2048 rising clock edges
- 12 - Glitch filter recognizes change on input pin after 4096 rising clock edges
- 13 - Glitch filter recognizes change on input pin after 8192 rising clock edges
- 14 - Glitch filter recognizes change on input pin after 16384 rising clock edges
- 15 - Glitch filter recognizes change on input pin after 32768 rising clock edges

Note

: This is an Implementation Specific Parameter.

Table 4-54. Attribute IcuLptmrPrescaler (IcuLptmr) detailed description

Property	Value
Type	ENUMERATION
POSTBUILDVARIANTVALUE	true
Origin	Custom
Symbolic Name	false
DEFAULT	ICU_LPTMR_GLITCH_FILTER_2
Range	ICU_LPTMR_GLITCH_FILTER_2

Table 4-54. Attribute IcuLptmrPrescaler (IcuLptmr) detailed description

Property	Value
	ICU_LPTMR_GLITCH_FILTER_4 ICU_LPTMR_GLITCH_FILTER_8 ICU_LPTMR_GLITCH_FILTER_16 ICU_LPTMR_GLITCH_FILTER_32 ICU_LPTMR_GLITCH_FILTER_64 ICU_LPTMR_GLITCH_FILTER_128 ICU_LPTMR_GLITCH_FILTER_256 ICU_LPTMR_GLITCH_FILTER_512 ICU_LPTMR_GLITCH_FILTER_1024 ICU_LPTMR_GLITCH_FILTER_2048 ICU_LPTMR_GLITCH_FILTER_4096 ICU_LPTMR_GLITCH_FILTER_8192 ICU_LPTMR_GLITCH_FILTER_16384 ICU_LPTMR_GLITCH_FILTER_32768

4.6.5.4 IcuLptmrChannelClkSrc (IcuLptmr)

Vendor specific: The ICU module specific clock input for the timer unit can statically be configured and allows to select different clock sources per module. Select the clock source for the FlexTimer module for this platform:

- SIRC_CLOCK
- LPO_CLOCK
- SIM_LPO_CLOCK
- PCC_CLOCK

Note

: This is an Implementation Specific Parameter.

Table 4-55. Attribute IcuLptmrChannelClkSrc (IcuLptmr) detailed description

Property	Value
Type	ENUMERATION
Origin	Custom
POSTBUILDVARIANTVALUE	true
Symbolic Name	false
DEFAULT	SIRC_CLOCK
Range	SIRC_CLOCK LPO_CLOCK SIM_LPO_CLOCK PCC_CLOCK

4.6.5.5 IcuLptmrPinSelect (IcuLptmr)

Configures the input source to be used in Pulse Counter mode:

- TRGMUX_OUTPUT
- ALT1
- ALT2
- ALT3

Note

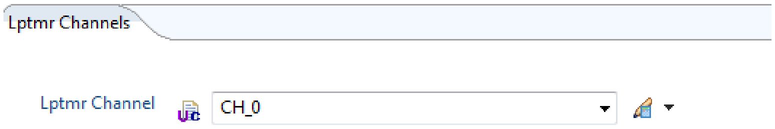
: This is an Implementation Specific Parameter.

Table 4-56. Attribute IcuLptmrPinSelect (IcuLptmr) detailed description

Property	Value
Type	ENUMERATION
Origin	Custom
POSTBUILDVARIANTVALUE	true
Symbolic Name	false
DEFAULT	ALT1
Range	TRGMUX_OUTPUT ALT1 ALT2 ALT3

4.6.5.6 Form IcuLptmrChannel

Is included by form : Form IcuLptmr



Lptmr Channel CH_0

Figure 4-16. Tresos Plugin snapshot for IcuLptmrChannel form.

4.6.5.6.1 IcuLptmrChannel (IcuLptmrChannel)

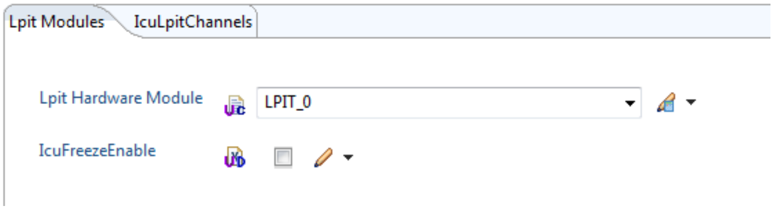
Selects one of the LPTMR channels available on the platform.

Table 4-57. Attribute IcuLptmrChannel (IcuLptmrChannel) detailed description

Property	Value
LABEL	Lptmr Channel
Type	ENUMERATION
POSTBUILDVARIANTVALUE	true
Origin	Custom
Symbolic Name	false

4.6.6 Form IcuLpit

Is included by form : Form IcuConfigSet



Lpit Hardware Module: LPIT_0

IcuFreezeEnable: □

Figure 4-17. Tresos Plugin snapshot for IcuLpit form.

4.6.6.1 IcuLpitModule (IcuLpit)

Select the physical Lpit Module.

Note

: This is an Implementation Specific Parameter.

Table 4-58. Attribute IcuLpitModule (IcuLpit) detailed description

Property	Value
LABEL	Lpit Hardware Module
Type	ENUMERATION
POSTBUILDVARIANTVALUE	true
Origin	Custom
Symbolic Name	false
Range	LPIT_0

4.6.6.2 IcuFreezeEnable (IcuLpit)

Select to set hardware freeze enable during debug mode.

Note

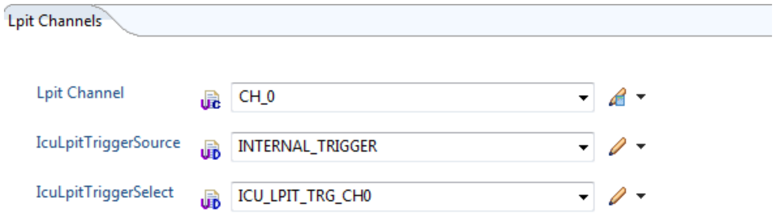
: This is an Implementation Specific Parameter.

Table 4-59. Attribute IcuFreezeEnable (IcuLpit) detailed description

Property	Value
Type	BOOLEAN
Origin	Custom
POSTBUILDVARIANTVALUE	true
Symbolic Name	false
DEFAULT	false

4.6.6.3 Form IcuLptmrChannels

Is included by form : Form IcuLpit



Lpit Channel: CH_0

IcuLpitTriggerSource: INTERNAL_TRIGGER

IcuLpitTriggerSelect: ICU_LPIT_TRG_CH0

Figure 4-18. Tresos Plugin snapshot for IcuLpitChannels form.

4.6.6.3.1 IcuLpitChannel (IcuLpitChannels)

Selects one of the LPIT channels available on the platform.

Table 4-60. Attribute IcuLpitChannel (IcuLpitChannels) detailed description

Property	Value
LABEL	Lpit Channel
Type	ENUMERATION
POSTBUILDVARIANTVALUE	true
Origin	Custom
Symbolic Name	false

4.6.6.3.2 IcuLpitTriggerSource (IcuLpitChannels)

Selects the internal and external trigger sources: INTERNAL_TRIGGER or EXTERNAL_TRIGGER.

Table 4-61. Attribute IcuLpitTriggerSource (IcuLpitChannels) detailed description

Property	Value
Type	ENUMERATION
POSTBUILDVARIANTVALUE	true
Origin	Custom
Symbolic Name	false
DEFAULT	INTERNAL_TRIGGER
Range	INTERNAL_TRIGGER EXTERNAL_TRIGGER

4.6.6.3.3 IcuLpitTriggerSelect (IcuLpitChannels)

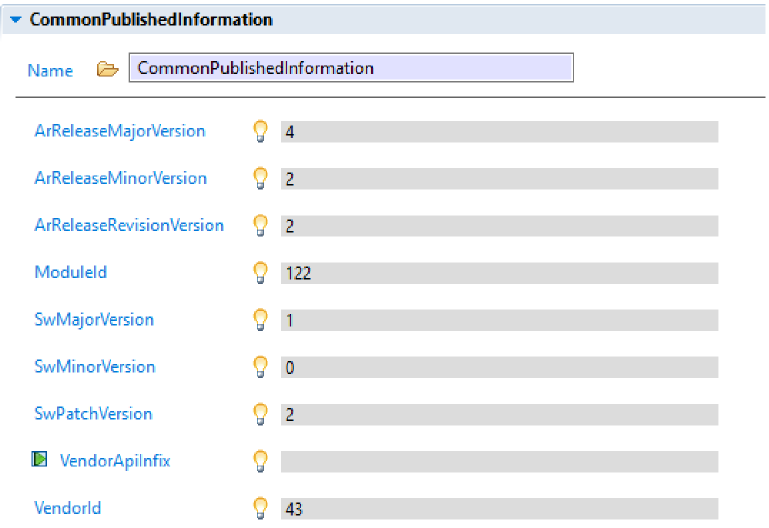
Select one trigger from the set of internal or external triggers selected by TRG_SRC.

Table 4-62. Attribute IcuLpitTriggerSelect (IcuLpitChannels) detailed description

Property	Value
Type	ENUMERATION
POSTBUILDVARIANTVALUE	true
Origin	Custom
Symbolic Name	false
DEFAULT	INTERNAL_TRIGGER
Range	ICU_LPIT_TRG_CH0 ICU_LPIT_TRG_CH1 ICU_LPIT_TRG_CH2 ICU_LPIT_TRG_CH3

4.7 Form CommonPublishedInformation

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



CommonPublishedInformation

Name: CommonPublishedInformation

ArReleaseMajorVersion	4
ArReleaseMinorVersion	2
ArReleaseRevisionVersion	2
ModuleId	122
SwMajorVersion	1
SwMinorVersion	0
SwPatchVersion	2
VendorApiInfix	
VendorId	43

Figure 4-19. Tresos Plugin snapshot for CommonPublishedInformation form.

4.7.1 ArReleaseMajorVersion (CommonPublishedInformation)

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

Table 4-63. Attribute ArReleaseMajorVersion (CommonPublishedInformation) detailed description

Property	Value
Label	AUTOSAR Major Version
Type	INTEGER_LABEL
Origin	Custom
Symbolic Name	false
Default	4
Invalid	Range >=4 <=4

4.7.2 ArReleaseMinorVersion (CommonPublishedInformation)

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

Table 4-64. Attribute ArReleaseMinorVersion (CommonPublishedInformation) detailed description

Property	Value
Label	AUTOSAR Minor Version
Type	INTEGER_LABEL
Origin	Custom
Symbolic Name	false
Default	2
Invalid	Range >=2 <=2

4.7.3 ArReleaseRevisionVersion (CommonPublishedInformation)

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

Table 4-65. Attribute ArReleaseRevisionVersion (CommonPublishedInformation) detailed description

Property	Value
Label	AUTOSAR Release Revision Version
Type	INTEGER_LABEL
Origin	Custom
Symbolic Name	false
Default	2
Invalid	Range >=2 <=2

4.7.4 ModuleId (CommonPublishedInformation)

Module ID of this module from Module List.

Table 4-66. Attribute ModuleId (CommonPublishedInformation) detailed description

Property	Value
Label	Module Id
Type	INTEGER_LABEL
Origin	Custom
Symbolic Name	false

Table continues on the next page...

Table 4-66. Attribute ModuleId (CommonPublishedInformation) detailed description (continued)

Property	Value
Default	122
Invalid	Range >=122 <=122

4.7.5 SwMajorVersion (CommonPublishedInformation)

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

Table 4-67. Attribute SwMajorVersion (CommonPublishedInformation) detailed description

Property	Value
Label	Software Major Version
Type	INTEGER_LABEL
Origin	Custom
Symbolic Name	false
Default	1
Invalid	Range >=1 <=1

4.7.6 SwMinorVersion (CommonPublishedInformation)

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

Table 4-68. Attribute SwMinorVersion (CommonPublishedInformation) detailed description

Property	Value
Label	Software Minor Version
Type	INTEGER_LABEL
Origin	Custom
Symbolic Name	false
Default	0
Invalid	Range >=0 <=0

4.7.7 SwPatchVersion (CommonPublishedInformation)

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

Table 4-69. Attribute SwPatchVersion (CommonPublishedInformation) detailed description

Property	Value
Label	Software Patch Version
Type	INTEGER_LABEL
Origin	Custom
Symbolic Name	false
Default	2
Invalid	Range >=2 <=2

4.7.8 VendorApiInfix (CommonPublishedInformation)

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><Api name from SWS>. 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.

Table 4-70. Attribute VendorApiInfix (CommonPublishedInformation) detailed description

Property	Value
Label	Vendor Api Infix
Type	STRING_LABEL
Origin	Custom
Symbolic Name	false
Default	
Enable	false

4.7.9 VendorId (CommonPublishedInformation)

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

Table 4-71. Attribute VendorId (CommonPublishedInformation) detailed description

Property	Value
Label	Vendor Id
Type	INTEGER_LABEL
Origin	Custom
Symbolic Name	false
Default	43
Invalid	Range >=43 <=43

How to Reach Us:

Home Page:
nxp.com

Web Support:
nxp.com/support

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

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

While NXP has implemented advanced security features, all products may be subject to unidentified vulnerabilities. Customers are responsible for the design and operation of their applications and products to reduce the effect of these vulnerabilities on customer's applications and products, and NXP accepts no liability for any vulnerability that is discovered. Customers should implement appropriate design and operating safeguards to minimize the risks associated with their applications and products.